Lecture 16: Causal Inference II

aka STAT109A, AC209A, CSCIE-109A

CS109A Introduction to Data Science

Pavlos Protopapas, Natesh Pillai, and Chris Gumb

Outline

AB Testing (Randomized, Control Trials)

Adjusting for Confounders

Propensity Scores

- Estimation

- Adjusting, Weighting, and Matching

Covariate Balancing

Experiments and A/B Testing

- In the world of Data Science, performing experiments to determine causation, like a randomized, controlled trial (RCT), is called **A/B testing**.

- Control group: group of users exposed to the “normal” or “baseline” version of the system

- Treatment group: group of users exposed to the experimental version of the system

Experiments and A/B Testing

A/B testing is often used in the tech industry

- to determine which form of website design (the treatment) leads to more ad clicks, purchases, etc… (the response).

- or to determine the effect of a new app rollout (treatment) on revenue or usage (the response).

What is an A/B Test?

Options

A. The Presidential Fitness Test of how many sit-ups you can do in 60 seconds.

B. Any comparison of CS 109A and CS 109B.

C. Any comparison of 2 groups or treatments.

[✓ D. An example of a Randomized, Controlled Trial to determine a causal relationship between treatment and a response/outcome.]

Today's thought example: the golden AI goose

CS 109 is thinking about implementing an AI teaching assistant to aid in Python coding for the course.

- This will be an interactive widget, and appear as a golden goose inside the Jupyter notebook:

How should we go about implementing an experiment (A/B test) in order to determine its effectiveness? How should we measure effectiveness?

Completely Randomized Design

There are many ways to design an experiment, depending on the number of treatment types, number of treatment groups, how the treatment effect may vary across subgroups, etc…

The simplest type of experiment is called a Completely Randomized Design (CRD). If two treatments, call them treatment *A* and treatment *B*, are to be compared across *n* subjects, then *n*/2 subject are randomly assigned to each group.

- If *n* = 100, this is equivalent to putting all 100 names in a hat, and pulling 50 names out and assigning them to treatment *A*.

Assigning subject to treatments

In order to **balance confounders**, the subjects must be properly randomly assigned to the treatment groups, and sufficient enough sample sizes need to be used.

For a CRD with 2 treatment arms, how can this randomization be performed via a computer?

You can just sample $n/2$ numbers from the values 1, 2, …, n without replacement and assign those individuals (in a list) to treatment group *A*, and the rest to treatments group *B*. This is equivalent to sorting the list of numbers, with the first half going to treatment *A* and the rest going to treatment *B*.

This is just like a 50-50 test-train split!

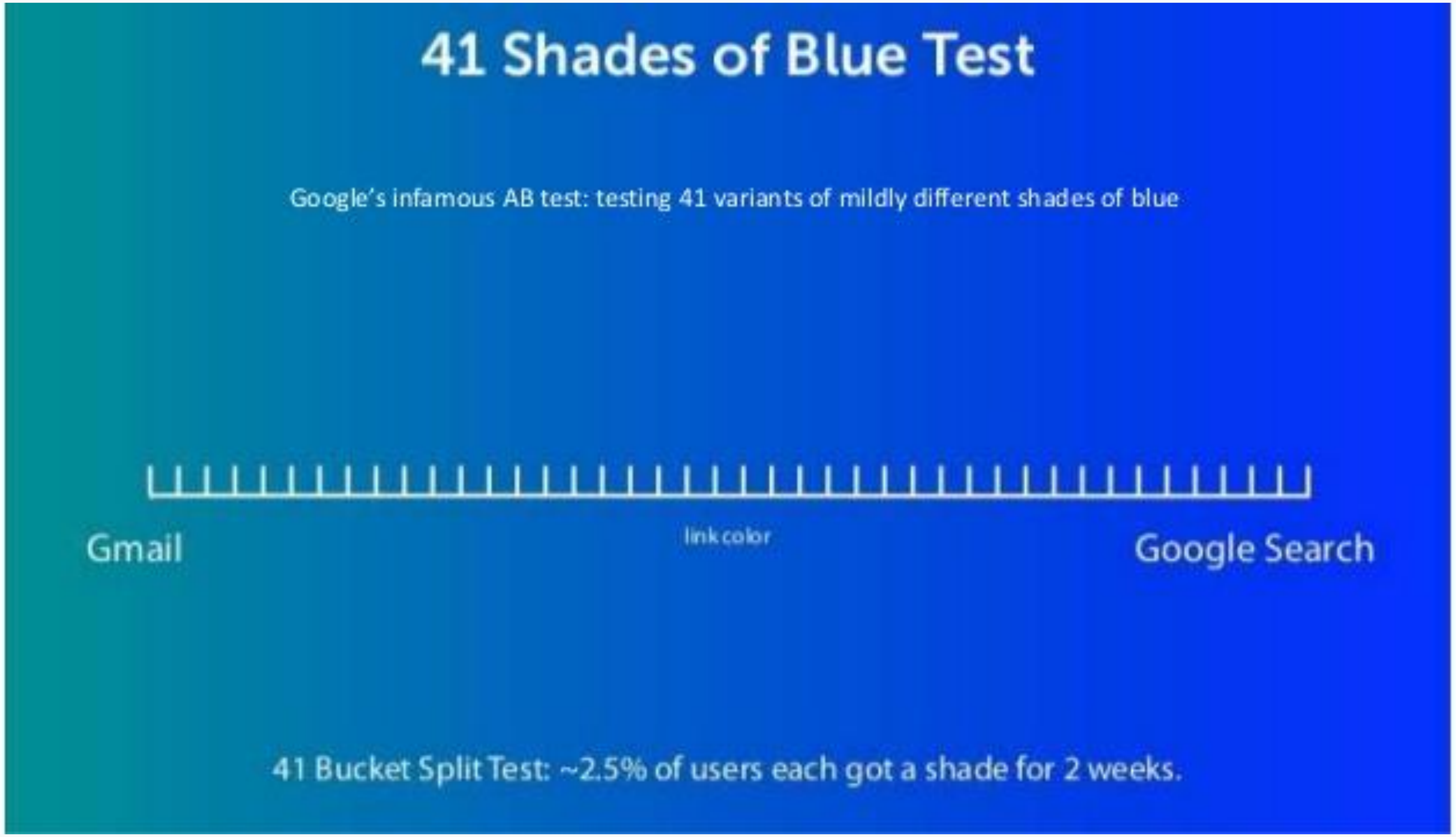
Beyond just A vs. B

How can an AB test be expanded to include more than two options? What if there are more than just one type of treatment?

1. The **multivariate experimental design** generalizes this approach. If there are two treatment types (font color, and website layout), then both treatments' effects can (and should) be tested simultaneously. Why?

2. In a **full factorial experimental** design, each and every combination of treatments are considered different treatment groups. Experiments online are cheap. Full factorial designs are often possible and feasible.

Example #1: An infamous A/B Test, 41 Shades of Blue



41 Shades of Blue Test

Google's infamous AB test: testing 41 variants of mildly different shades of blue

41 Bucket Split Test: ~2.5% of users each got a shade for 2 weeks.

How should the study proceed? How should the data be analyzed?

Example #2: The 2008 Obama Campaign

In 2008, the Obama campaign raised much of its money via online donations through its website.

They wanted to optimize the launch page that visitors saw when they came to the campaign website. They were attempting to maximize the number of visitors that would sign up for their emailing list.

There were 2 treatments they attempted to vary:

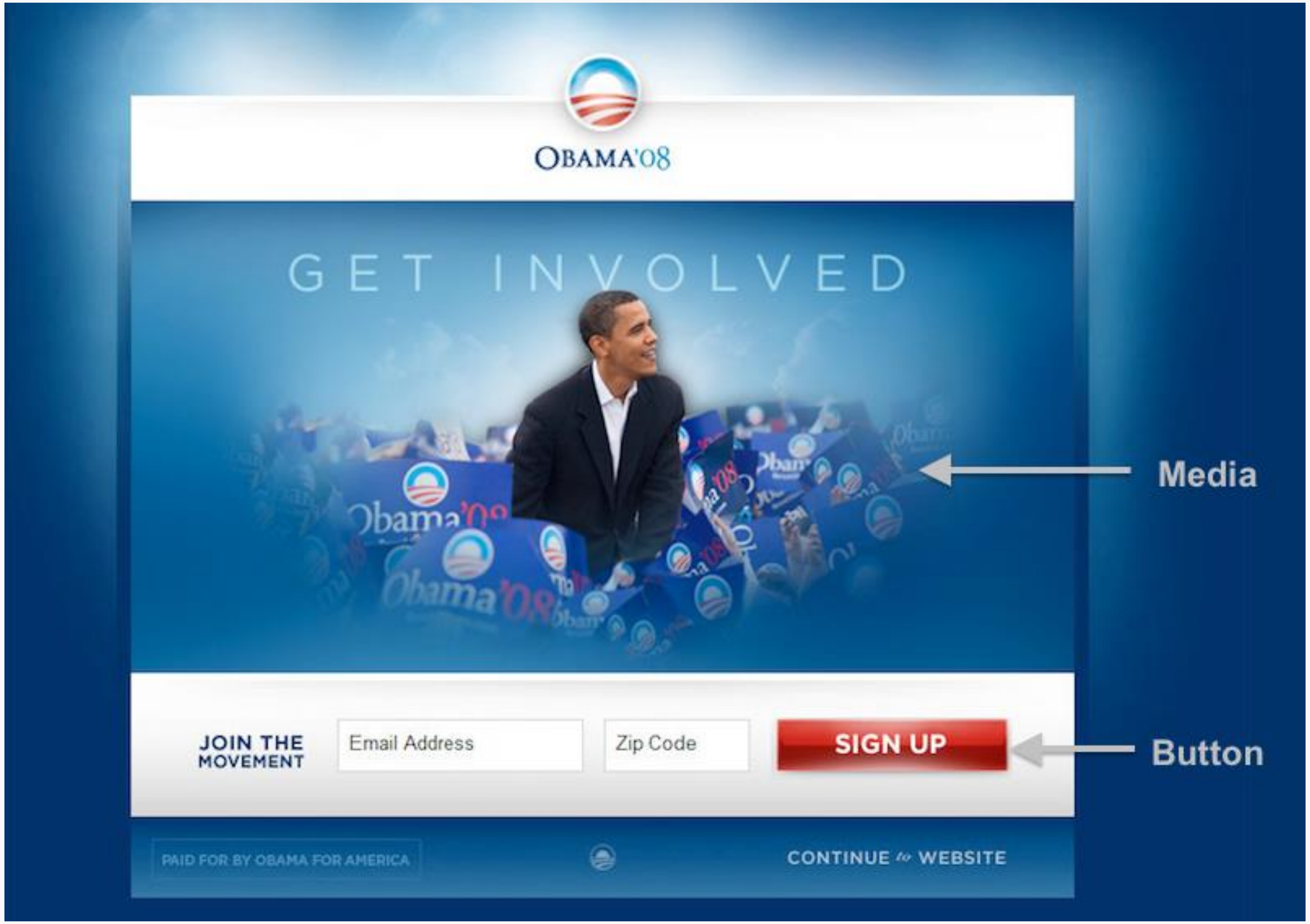
- the image or video the user saw.
- the words on the click-through button.

Media choices (6 of them):

- 3 images and 3 videos were possibly shown

Click- through button

- one of 4 choices:



How to design the experiment?

How should this experiment unfold?

1. What was the response variable?
2. What were the treatments? What were the treatment groups?
3. How many observations (sample size) needed to be selected in order to determine which treatment group is most effective?
4. What analysis should be performed?

Obama 2008: the specifics

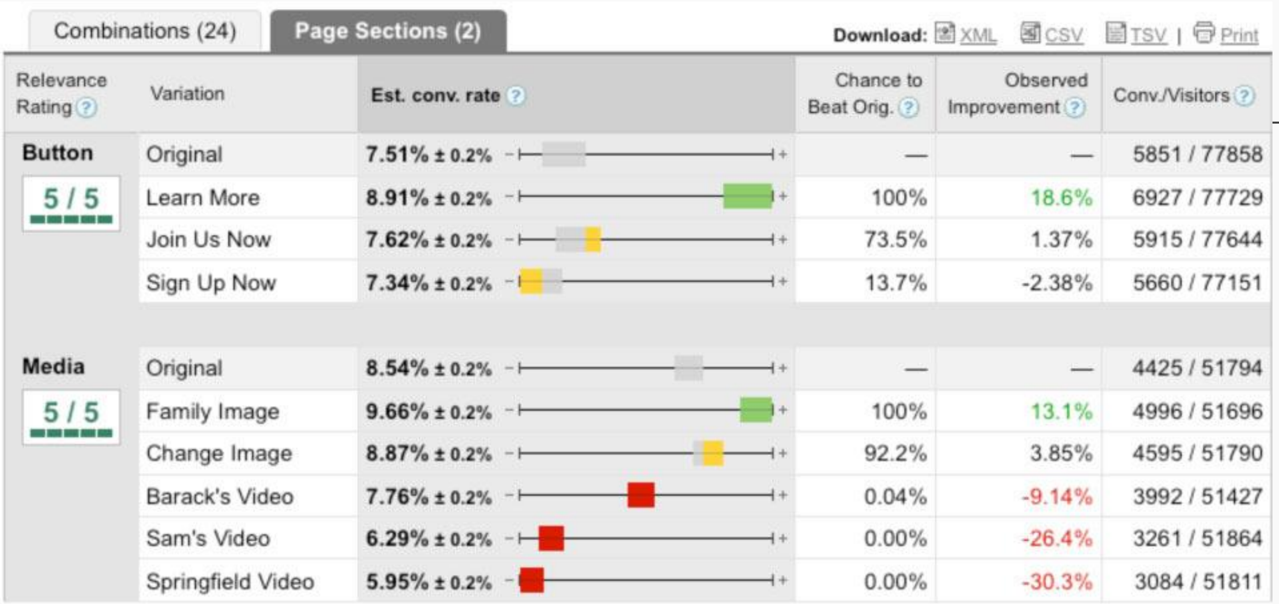
1. What was the response variable?
   – sign-up rate
2. What were the treatments? What were the treatment groups?
   – 2 treatments (media type and button). 24 treatment groups
3. How many observations (sample size) needed to be selected in order to determine which treatment group is most effective?
   – The campaign decided to run the experiment on 310,382 visitors!!!
4. What analysis should be performed?
   – Classically, a χ^2 test for independence could be performed. Or better yet, a randomization test ☺

The data were overwhelming…

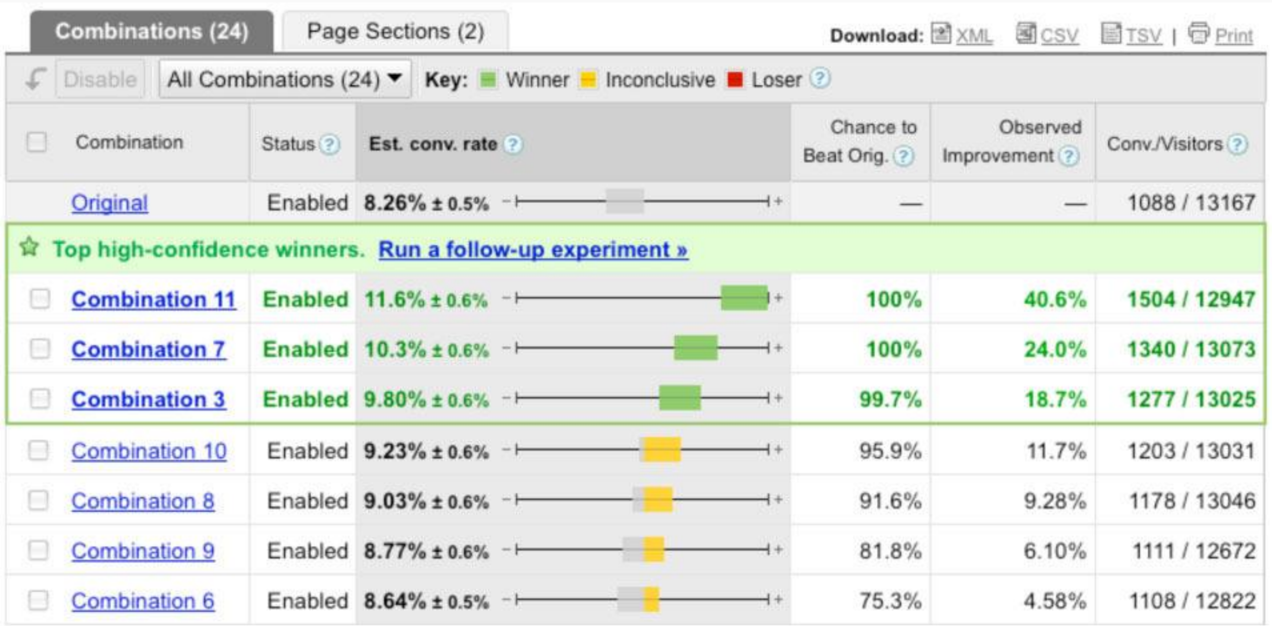
The Results

The results are shown to the right (note: they are from a 3rd party site that runs AB tests for website design: Optimizely).

https://blog.optimizely.com/2010/11/29/how-obama-raised-60-million-by-running-a-simple-experiment/



Relevance Rating ?	Variation	Est. conv. rate ?	Chance to Beat Orig. ?	Observed Improvement ?	Conv./Visitors ?
Button	Original	**7.51%** ± 0.2%	—	—	5851 / 77858
5 / 5	Learn More	**8.91%** ± 0.2% 🟩	100%	18.6%	6927 / 77729
	Join Us Now	**7.62%** ± 0.2% 🟨	73.5%	1.37%	5915 / 77644
	Sign Up Now	**7.34%** ± 0.2% 🟨	13.7%	-2.38%	5660 / 77151
Media	Original	**8.54%** ± 0.2%	—	—	4425 / 51794
5 / 5	Family Image	**9.66%** ± 0.2% 🟩	100%	13.1%	4996 / 51696
	Change Image	**8.87%** ± 0.2% 🟨	92.2%	3.85%	4595 / 51790
	Barack's Video	**7.76%** ± 0.2% 🟥	0.04%	-9.14%	3992 / 51427
	Sam's Video	**6.29%** ± 0.2% 🟥	0.00%	-26.4%	3261 / 51864
	Springfield Video	**5.95%** ± 0.2% 🟥	0.00%	-30.3%	3084 / 51811



Key: 🟩 Winner 🟨 Inconclusive 🟥 Loser ?

☐ Combination	Status ?	Est. conv. rate ?	Chance to Beat Orig. ?	Observed Improvement ?	Conv./Visitors ?
Original	Enabled	**8.26%** ± 0.5%	—	—	1088 / 13167
☆ **Top high-confidence winners.** **Run a follow-up experiment »**					
☐ **Combination 11**	**Enabled**	**11.6%** ± 0.6% 🟩	**100%**	**40.6%**	**1504 / 12947**
☐ **Combination 7**	**Enabled**	**10.3%** ± 0.6% 🟩	**100%**	**24.0%**	**1340 / 13073**
☐ **Combination 3**	**Enabled**	**9.80%** ± 0.6% 🟩	**99.7%**	**18.7%**	**1277 / 13025**
☐ Combination 10	Enabled	**9.23%** ± 0.6% 🟨	95.9%	11.7%	1203 / 13031
☐ Combination 8	Enabled	**9.03%** ± 0.6% 🟨	91.6%	9.28%	1178 / 13046
☐ Combination 9	Enabled	**8.77%** ± 0.6% 🟨	81.8%	6.10%	1111 / 12672
☐ Combination 6	Enabled	**8.64%** ± 0.5% 🟨	75.3%	4.58%	1108 / 12822

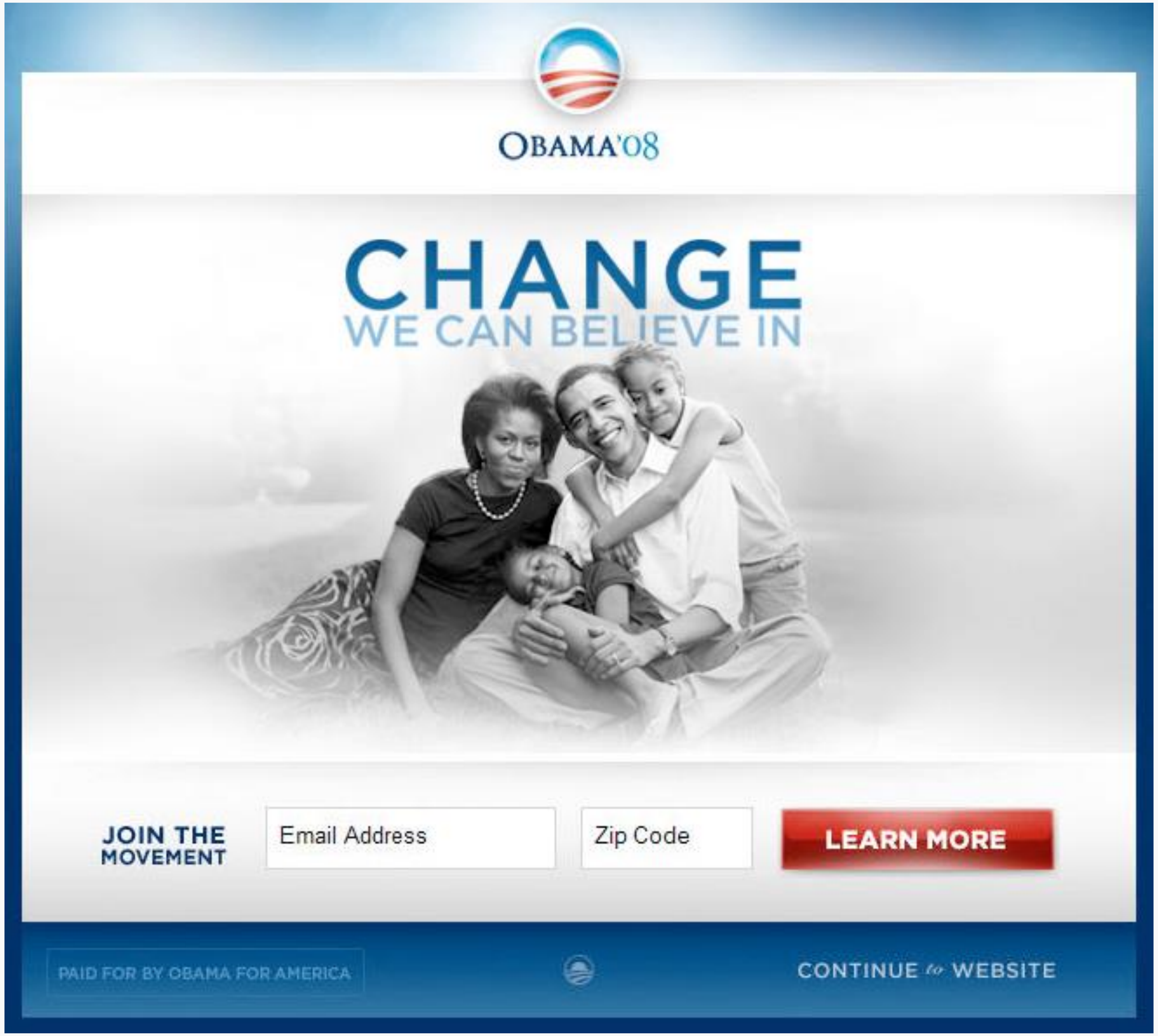
The results (cont.)

The winning variation had a sign-up rate of 11.6%. The original page had a sign-up rate of 8.26%. That's an improvement of 40.6% in sign-up rate…[leading to an] additional 2,880,000 email addresses translated into 288,000 more volunteers…and an additional $60 million in donations.

See any issue in this conclusion?

But more importantly, they did learn one lesson: those intimately involved in designing websites (or medical treatments) are too biased to properly make conclusions as to what works best. They like the videos, and they performed the worst.

And the winner was:



Example #3: The app update roll-out problem

A company is interested in updating their app/program, so they start a ‘pilot program’ to test the waters to see how this update will affect some important measure (like revenue or usage). How should they do this?

They select a sample of users and ask them to voluntarily update the app on their phones in order to estimate the affect of this update.

Any issues with this design?

Volunteers will always be the most excited, dedicated users: a biased sample from all of their users.

We can potentially check for this bias via a χ^2 test for goodness-of-fit.

Challenges in A/B Testing

1. Understanding how short-term metrics (measured during A/B tests) lead to long-term improvements in user experience and/or revenue.

   Short-term vs. Long-term Metrics:

   – An increase in ad clicks suggests an increase in revenue.

   – Showing lots of ads (often) hurts the user experience and decreases retention (i.e., long-term ad-click revenue)

2. Selecting the correct metric.
3. Making sure that your claims are exactly tested.
4. Fixing bugs in the experimental infrastructure.
5. Using sound statistical methods.

Potential Problems in A/B Testing

1. We need enough to reach statistical significance. Insufficient sample sizes can lead to inconclusive or misleading results.
2. Users exposed to both variants over time might change their behavior, distorting test outcomes.
3. Testing different versions can create inconsistent user experiences, potentially frustrating or confusing users.
4. Designing, running, and analyzing A/B tests can consume significant resources in terms of time, tools, and manpower.

Outline

AB Testing (Randomized, Control Trials)

Adjusting for Confounders

Propensity Scores

- Estimation
- Adjusting, Weighting, and Matching

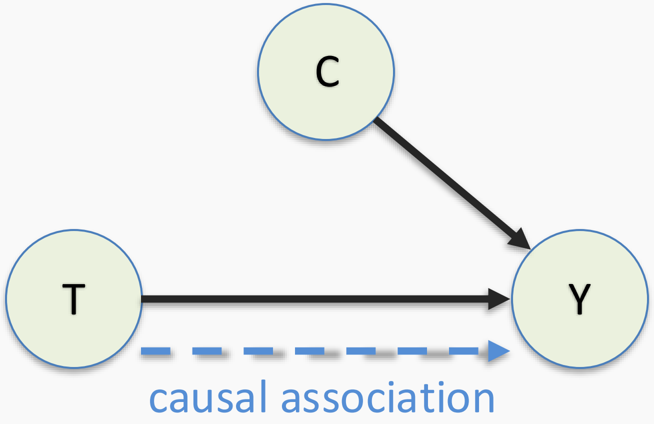
Covariate Balancing

Observational Studies

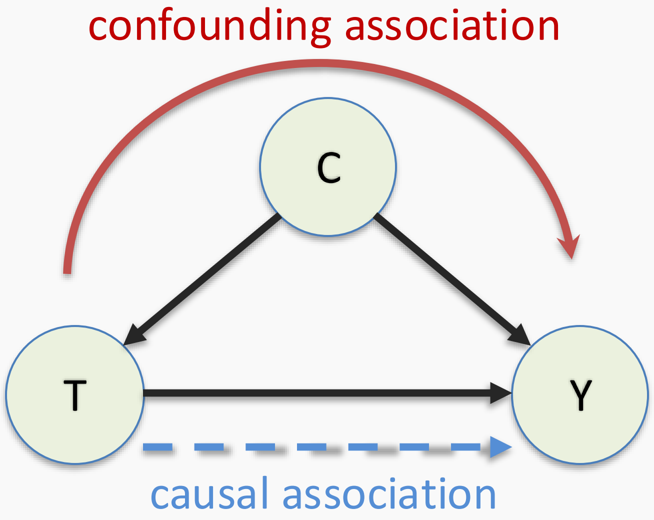
Can’t always perform RCT

- Can’t randomize
- Ethical problem
- Not feasible
- Not possible
- RCTs are expensive and difficult to set

Ideal



Observational studies



Recall the PyCA-109a example

Treatment T: A (Raderzole) and B (Gumboxin)

Condition Severity C: mild (0) or severe (1)

Outcome Y: alive (0) or dead (1)

	Condition Severity		
Treatment	Mild	Severe	Total
A (Raderzole)	17% 150/900	40% 40/100	19% 190/1000
B (Gumboxin)	**14%** 50/350	**33%** 50/150	20% 100/500

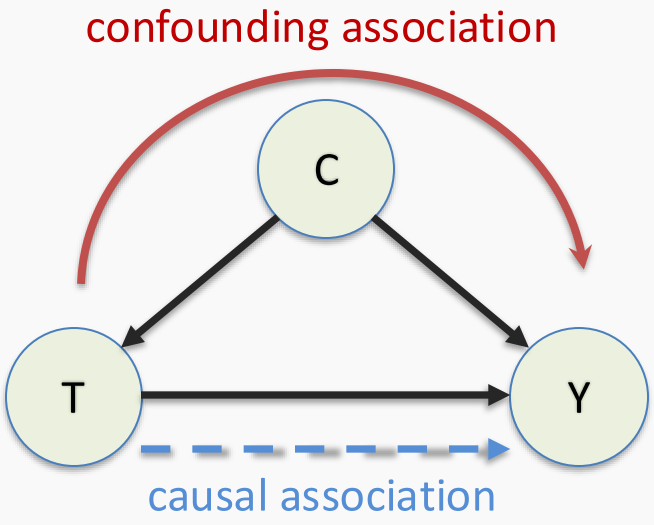
$\mathbb{E}[Y|T, C=0]$ $\mathbb{E}[Y|T, C=1]$ $\mathbb{E}[Y|T]$

Measuring a Causal Effect in Observational Studies

Adjust for cofounders C (condition severity).
Consider the quantity

$$\mathbb{E}[Y(t)|C = c]$$

Observational studies

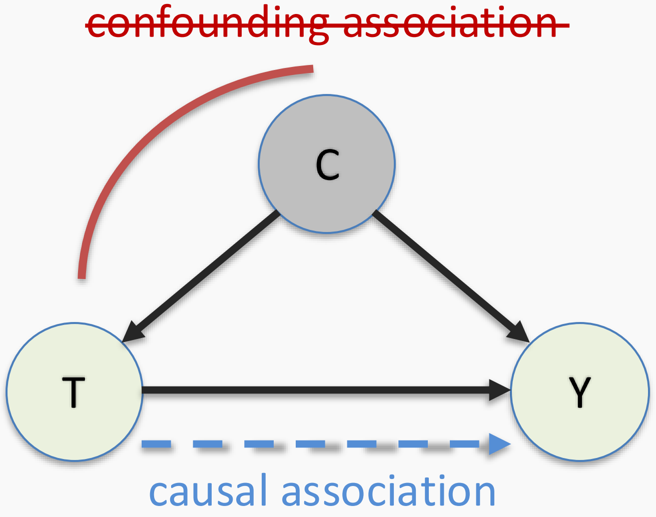


Measuring a Causal Effect in Observational Studies

Adjust for cofounders C (condition severity).

$$\mathbb{E}[Y(t)|C = c] = \mathbb{E}[Y|t, c]$$

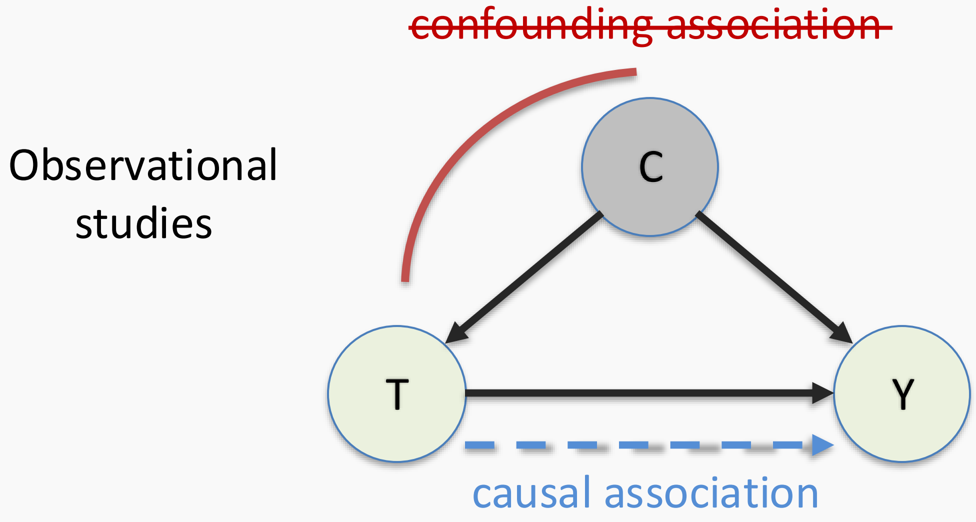
Observational studies



Measuring a Causal Effect in Observational Studies

Adjust for cofounders C (condition severity).

$$\mathbb{E}[Y(t)|C=c] = \mathbb{E}[Y|t,c]$$

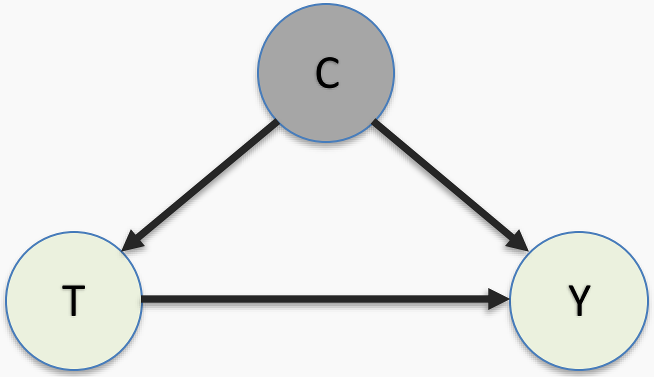


$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t,C]$$

Marginalize over confounder C

Example of Adjustment on PyCA-109a

$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t, C]$$

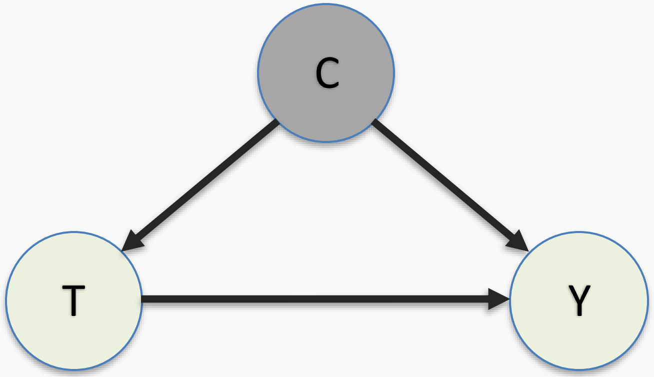


	Condition		
Treatment	Mild	Severe	Total
A	17% 150/900	40% 40/100	19% 190/1000
B	**14%** 50/350	**33%** 50/150	20% 100/500

$\mathbb{E}[Y|T, C = 0]$ $\mathbb{E}[Y|T, C = 1]$ $\mathbb{E}[Y|T]$

Example of Adjustment on PyCA-109a

$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t, C] = \sum_c \mathbb{E}_C[Y|t, c]\, P(c)$$

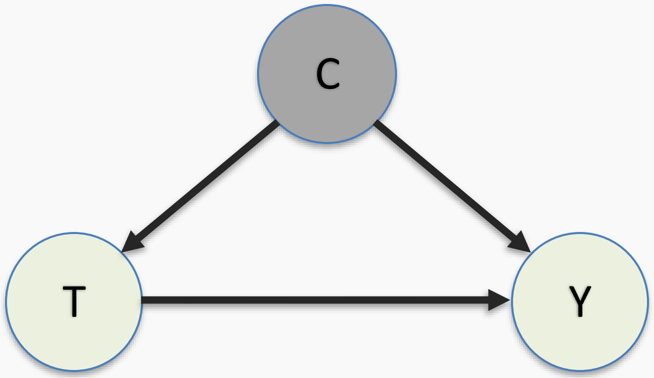


	Condition		
Treatment	Mild	Severe	Total
A	17% 150/900	40% 40/100	19% 190/1000
B	**14%** 50/350	**33%** 50/150	20% 100/500

$\mathbb{E}[Y|T, C = 0]$ $\mathbb{E}[Y|T, C = 1]$ $\mathbb{E}[Y|T]$

Example of Adjustment on PyCA-109a

$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t, C] = \sum_c \mathbb{E}_C[Y|t, c]\, P(c)$$



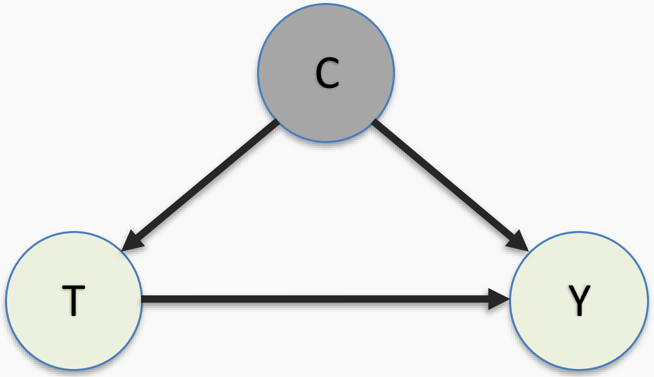
$$\boxed{\mathbb{E}[Y|T = A, C = 0] = 0.17}$$

	Condition			
Treatment	Mild	Severe	Total	Causal
A	17% 150/900	40% 40/100	19% 190/1000	
B	**14%** 50/350	**33%** 50/150	20% 100/500	

$\mathbb{E}[Y|T, C = 0]$ $\mathbb{E}[Y|T, C = 1]$ $\mathbb{E}[Y|T]$ $\mathbb{E}[Y|do(T)]$

Example of Adjustment on PyCA-109a

$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t,C] = \sum_c \mathbb{E}_C[Y|t,c]\,P(c)$$



	Condition			
Treatment	Mild	Severe	Total	Causal
A	17% 150/900	40% 40/100	19% 190/1000	22.5%
B	**14%** 50/350	**33%** 50/150	20% 100/500	17.2%

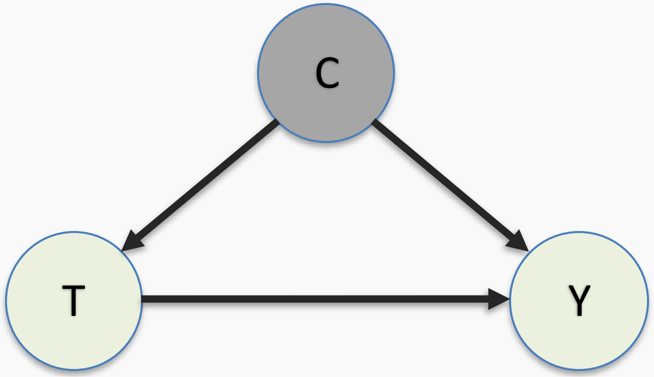
$\mathbb{E}[Y|T,C=0]$ $\mathbb{E}[Y|T,C=1]$ $\mathbb{E}[Y|T]$ $\mathbb{E}[Y|do(T)]$

$$\frac{1250}{1500}(0.17) + \frac{250}{1500}(0.4) = 0.208$$

$$\frac{1250}{1500}(0.14) + \frac{250}{1500}(0.33) = 0.172$$

Example of Adjustment on PyCA-109a

$$\mathbb{E}[Y(t)] = \mathbb{E}_C[Y|t,C] = \sum_c P(c)\mathbb{E}_C[Y|t,c]$$



	Condition Severity			
Treatment	Mild	Severe	Total	Causal, adjusted for C
A	17% 150/900	40% 40/100	19% 190/1000	20.83%
B	**14%** 50/350	**33%** 50/150	20% 100/500	17.17%

$\mathbb{E}[Y|T,C=0]$ $\mathbb{E}[Y|T,C=1]$ $\mathbb{E}[Y|T]$ $\mathbb{E}_C[Y|T]$

$P(c)$

$P(c)$ ***rebalances*** the confounder across treatments.

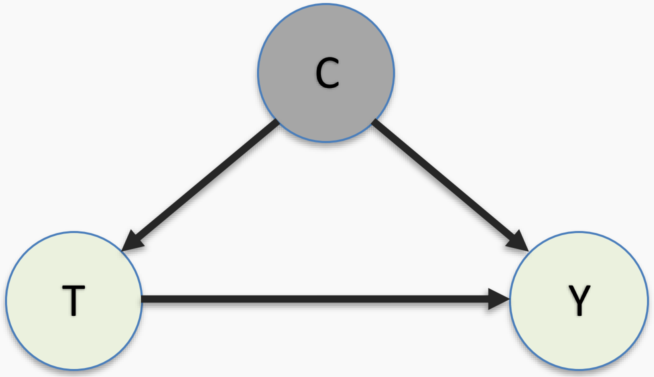
$$\frac{1250}{1500}(0.17) + \frac{250}{1500}(0.4) = 0.2083$$

$$\frac{1250}{1500}(0.14) + \frac{250}{1500}(0.33) = 0.1717$$

Example of Adjustment on PyCA-109a

Average treatment effect (ATE):

$$\mathbb{E}[Y_i(1) - Y_i(0)] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$$



	Condition Severity			
Treatment	Mild	Severe	Total	Causal, adjusted for C
A	17% 150/900	40% 40/100	19% 190/1000	20.83%
B	**14%** 50/350	**33%** 50/150	20% 100/500	17.17%
	$\mathbb{E}[Y \mid T, C = 0]$	$\mathbb{E}[Y \mid T, C = 1]$	$\mathbb{E}[Y \mid T]$	$\mathbb{E}_C[Y \mid T]$

$P(c)$

$P(c)$ ***rebalances*** the confounder across treatments.

$$\frac{1250}{1500}(0.17) + \frac{250}{1500}(0.4) = 0.2083$$

$$\frac{1250}{1500}(0.14) + \frac{250}{1500}(0.33) = 0.1717$$

AND THIS IS JUST THE BEGINNING!!! (and overly simplified)

- Synthetic control
- Regression discontinuity
- Instrumental variables
- Propensity scores
- Etc.

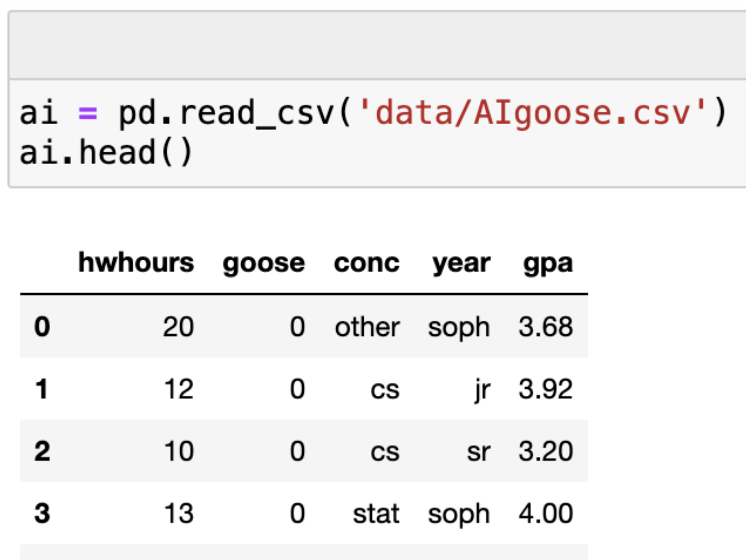
Today's thought example: the golden AI goose

How should we go about implementing an observational study in order to determine the golden AI goose's effectiveness?

What other variables (predictors/confounders) should we measure?

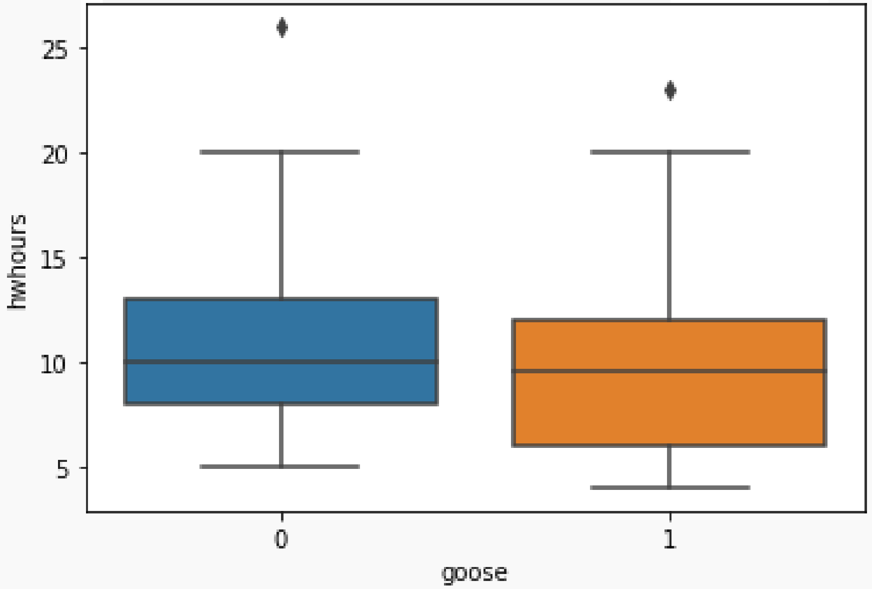
- Class year
- Concentration
- GPA
- Etc.

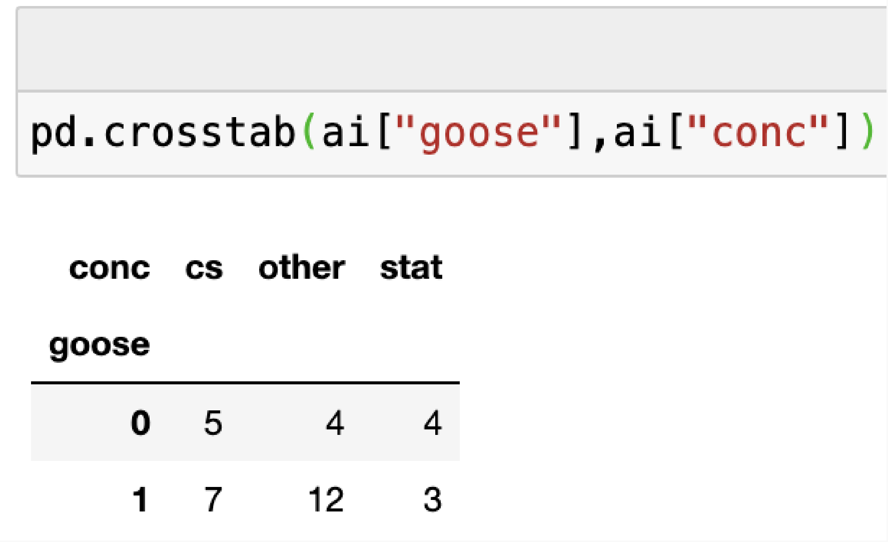
Example: the golden AI goose (synthetic data)



```
ai = pd.read_csv('data/AIgoose.csv')
ai.head()
```

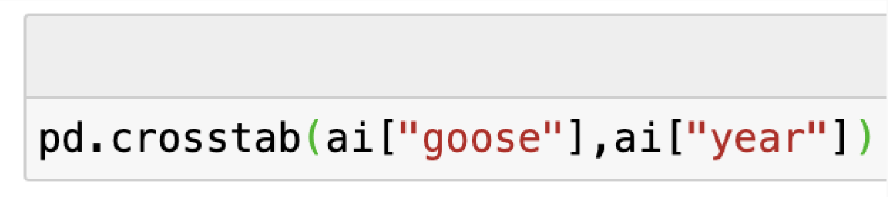
	hwhours	goose	conc	year	gpa
0	20	0	other	soph	3.68
1	12	0	cs	jr	3.92
2	10	0	cs	sr	3.20
3	13	0	stat	soph	4.00





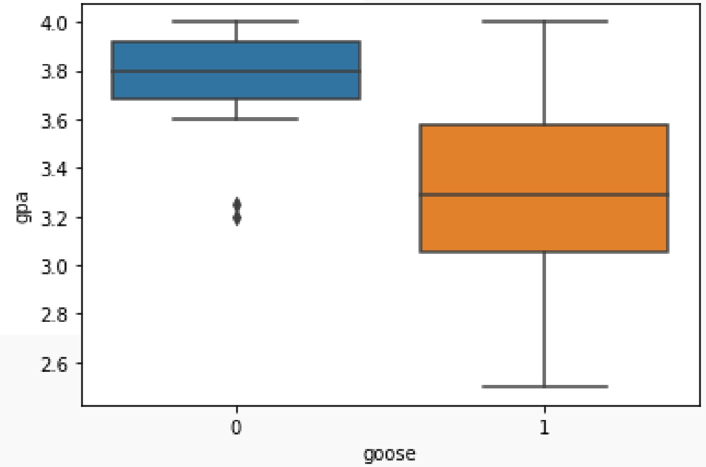
```
pd.crosstab(ai["goose"],ai["conc"])
```

conc	cs	other	stat
goose			
0	5	4	4
1	7	12	3



```
pd.crosstab(ai["goose"],ai["year"])
```

year	jr	soph	sr
goose			
0	7	3	3
1	10	5	7



Outline

AB Testing (Randomized, Control Trials)

Adjusting for Confounders

Propensity Scores

- Estimation
- Adjusting, Weighting, and Matching

Covariate Balancing

Propensity Scores

To adjust for the confounding with treatment assignment that may exist in an observational study, the **propensity score** can be estimated.

Propensity score: the **probability** that a subject/observation (ex: user, person, classroom, etc.) is assigned to a particular treatment, T_i, given their set of observed predictors, X_i.

$$e(X_i) = P(T_i = 1|X_i)$$

Why Does Propensity Score Matter

- Dimension Reduction:

  When the dimension of covariate X is very large, estimating causal effects can be complex. Propensity score simplifies this by reducing dimension.

- Correct Selection Bias:

  By using propensity score, we can upweight the units that are underrepresented in our data.

- Balance Covariates:

  Conditional on propensity score $e(X)$, the distribution of covariate X is the same for treated and untreated.

Propensity Scores: estimation

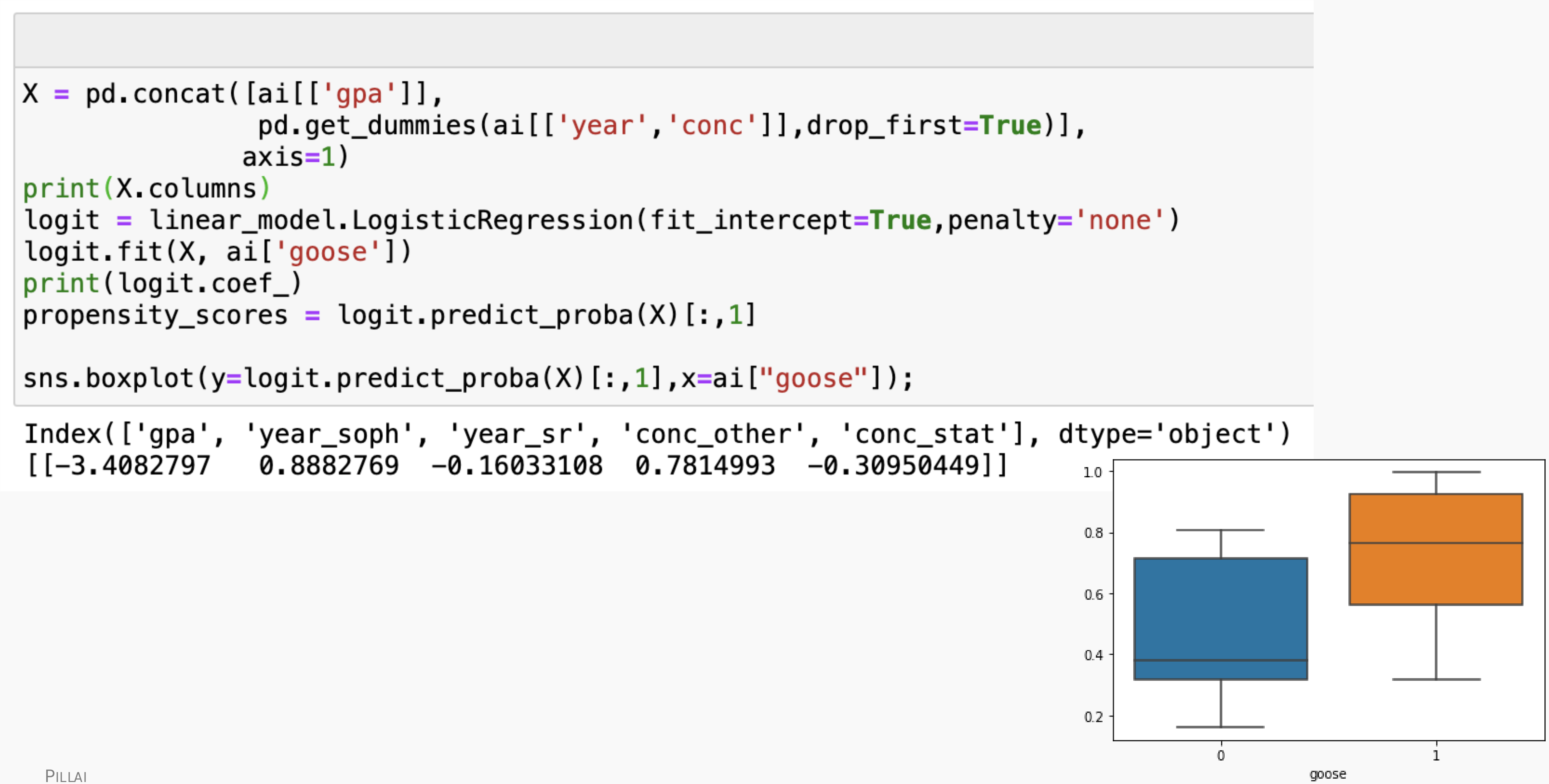
The propensity score is the probability of receiving a treatment given X (the predictors). How can we estimate/model this probability?

We can use **any** reasonable classification model to estimate the probability of **treatment** using the rest of the set of covariates.

Standard approach: use logistic regression.

Key confusion: this is a model for treatment, T_i, not the response, Y_i.

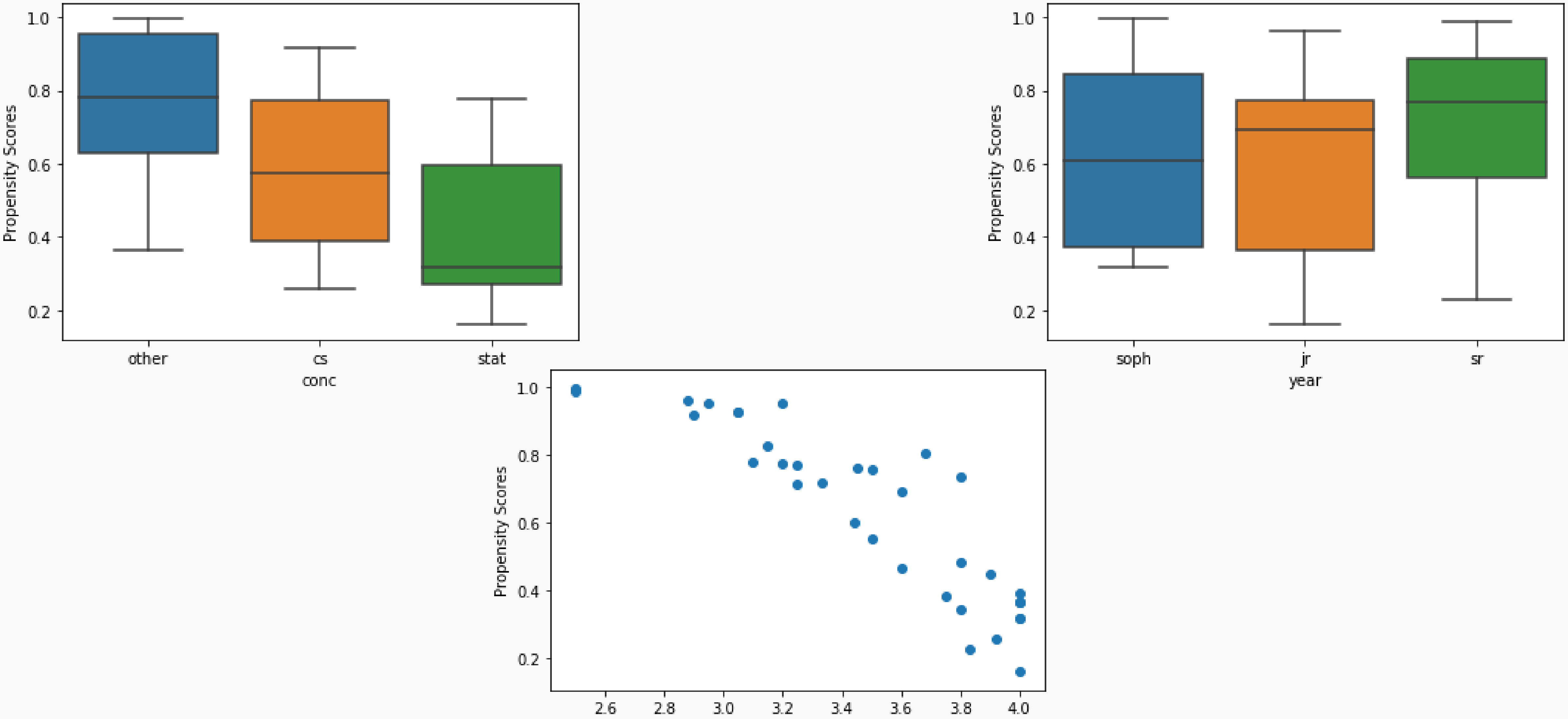
Propensity Scores: estimation



```
X = pd.concat([ai[['gpa']],
               pd.get_dummies(ai[['year','conc']],drop_first=True)],
              axis=1)
print(X.columns)
logit = linear_model.LogisticRegression(fit_intercept=True,penalty='none')
logit.fit(X, ai['goose'])
print(logit.coef_)
propensity_scores = logit.predict_proba(X)[:,1]

sns.boxplot(y=logit.predict_proba(X)[:,1],x=ai["goose"]);
```

```
Index(['gpa', 'year_soph', 'year_sr', 'conc_other', 'conc_stat'], dtype='object')
[[-3.4082797   0.8882769  -0.16033108  0.7814993  -0.30950449]]
```

Propensity Scores: evidence of confounding?



Using Propensity Scores for balancing confounders

Great, we now have propensity scores! How should we use them?

There are generally 3 approaches to incorporating propensity scores:

1. Covariate Adjustment:
2. Weighting:
3. Matching:

Using Propensity Scores: covariate adjustment

The simplest way to use propensity scores is to incorporate them into the main response model as **an additional (and only) predictor**:

This can result in a poorly adjusted model as the **estimand** is poorly defined.

Using Propensity Scores: inverse probability weighting

The second simplest way to use propensity scores is **to incorporate them as sampling weights** into the main response model:

This can result in a poorly estimated causal effect as the sampling weights can be huge (if $\hat{e}(X_i)$ is close to 0 or 1).

Using Propensity Scores: matching

The best way to use propensity scores is to match on them: for ever treated observation, find the most similar control observation. This creates a synthetic counterfactual for every observed treated observation.

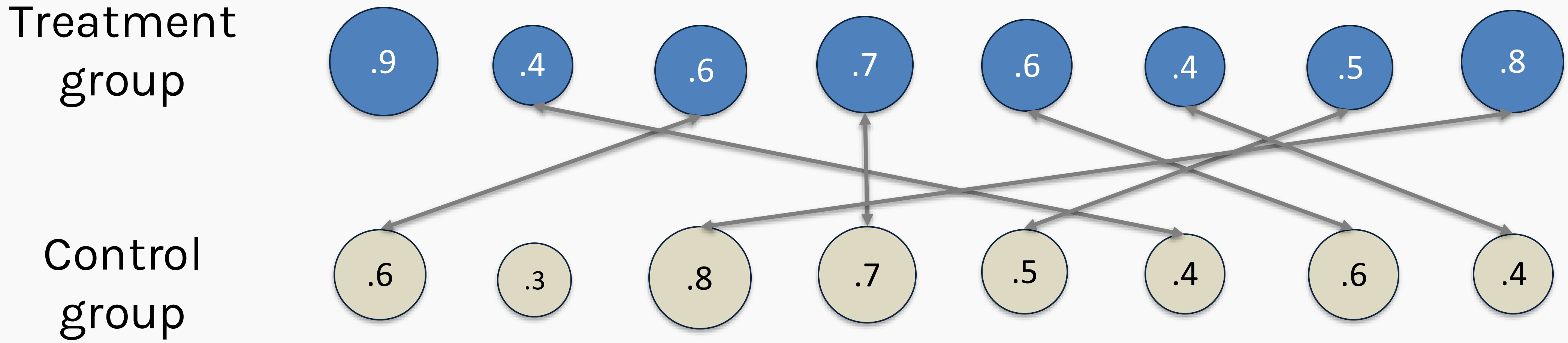
A few key considerations:

- Should we match with replacement?
- Should we strictly perform 1-to-1 matching?
- Should we throw away the extremes?

There are many ways to perform matching, and there is no one best approach or one measure to determine which matching algorithm is the best.

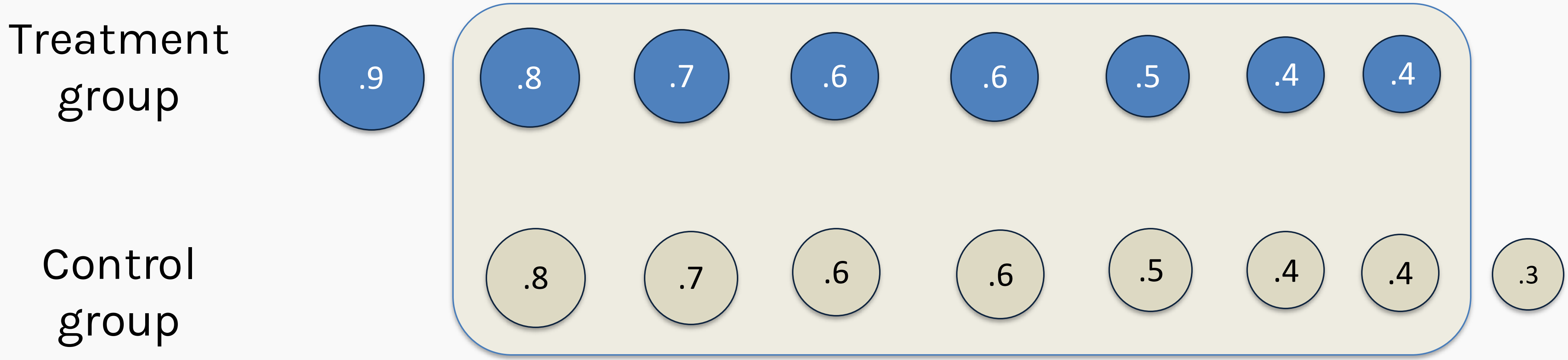
Propensity Score Matching: toy example

units with varying propensity scores



Propensity Score Matching: toy example

propensity scores macthing

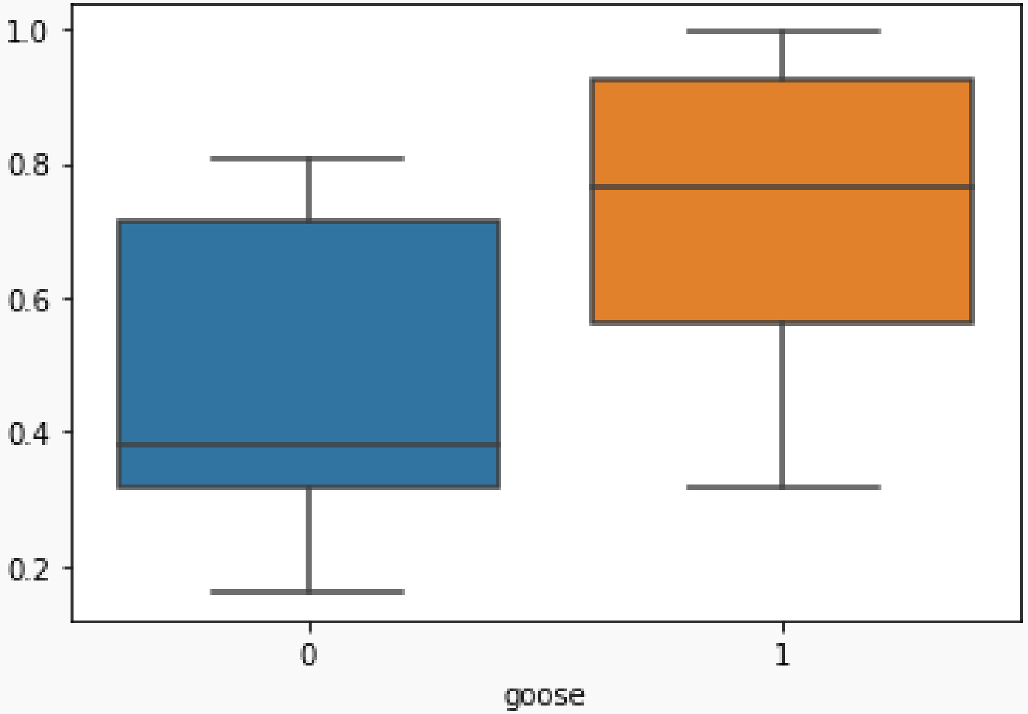


Using Propensity Scores: throwing away extremes

A thought question:

Should we even consider individuals that have probability of being treated close to 1 (or close to 0) in an observational study?

Consider this plot of propensity scores:

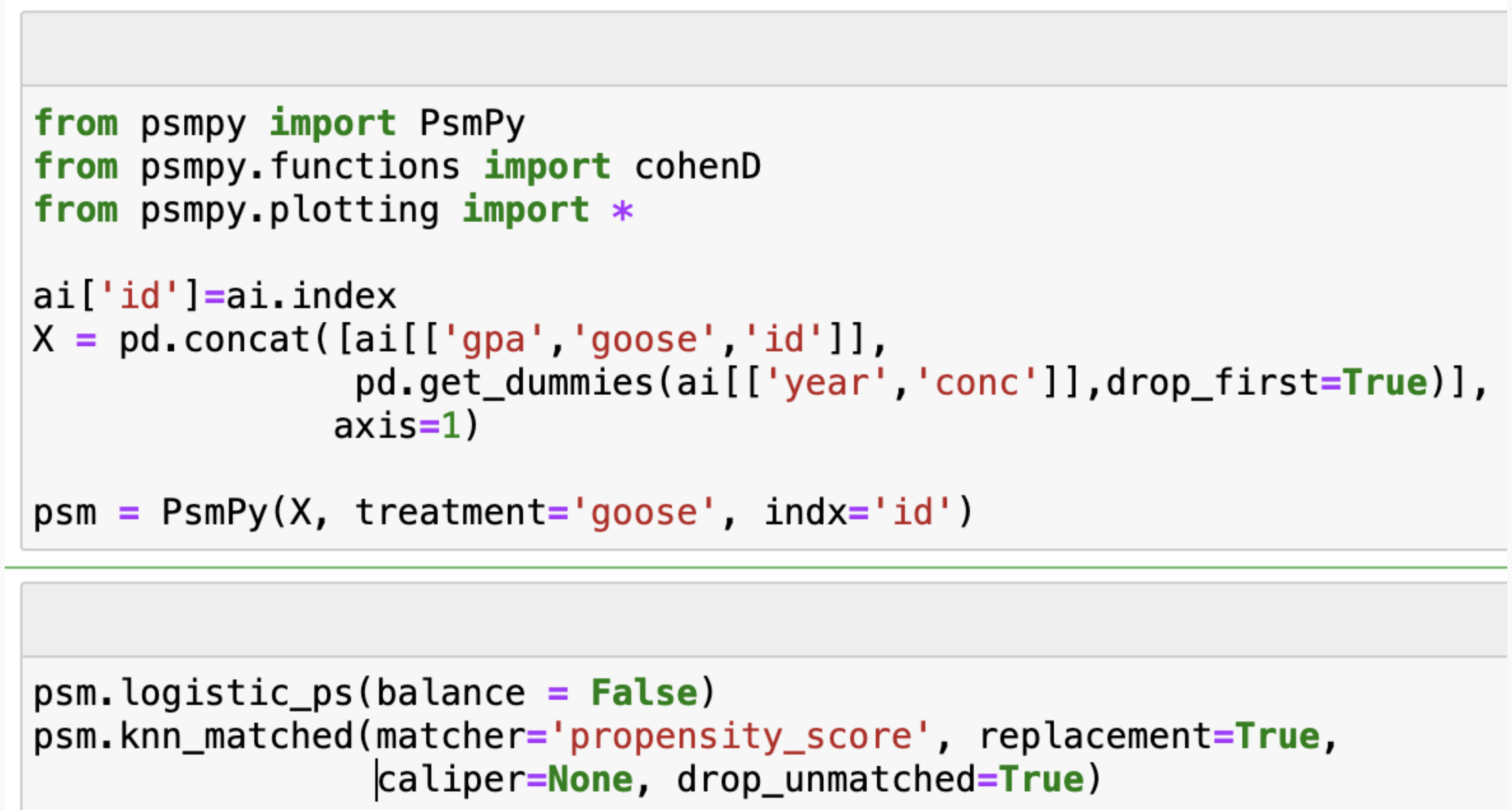


What are the matches for those students that did not use the golden goose (`goose = 0`) that have $\hat{e}(X_i) < 0.3$? What about the students for which `goose = 1` and $\hat{e}(X_i) > 0.8$?

Propensity Score Matching in Python

There is a Python package to aid in propensity score matching:

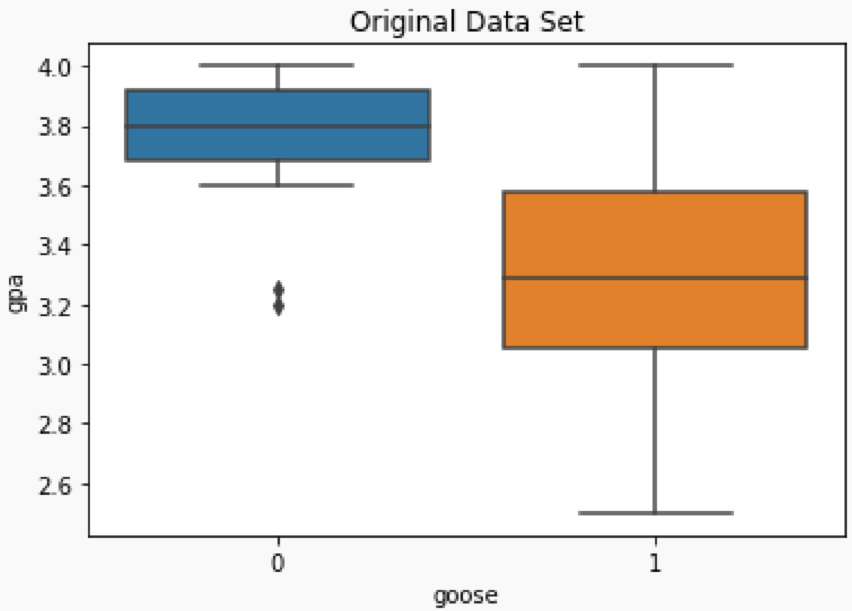
PsmPy: https://pypi.org/project/psmpy/

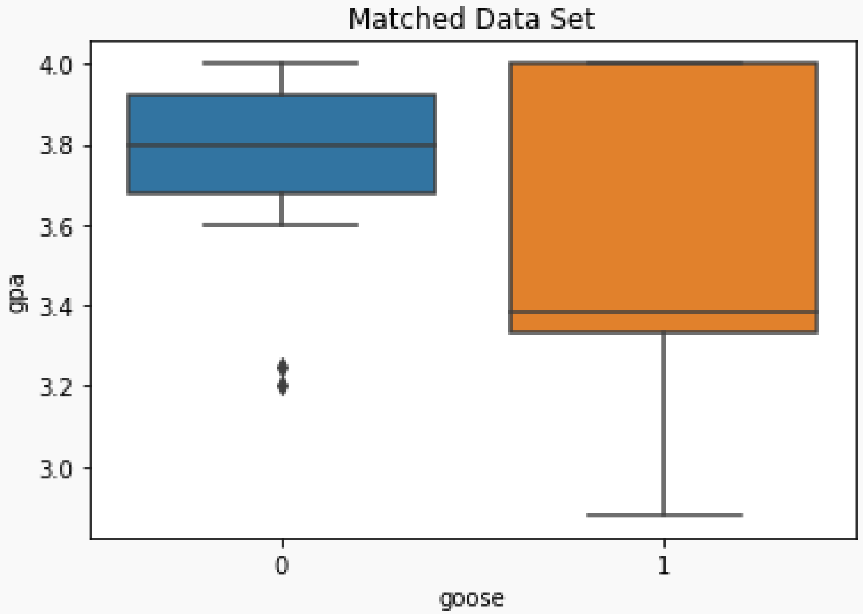


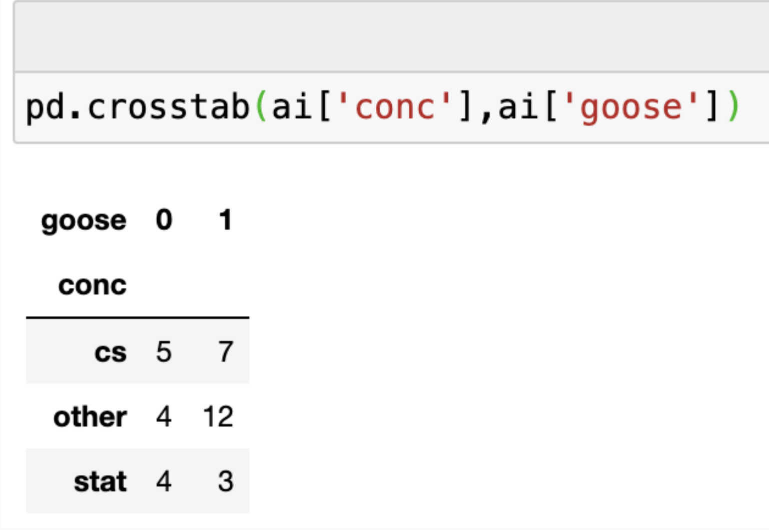
```
from psmpy import PsmPy
from psmpy.functions import cohenD
from psmpy.plotting import *

ai['id']=ai.index
X = pd.concat([ai[['gpa','goose','id']],
               pd.get_dummies(ai[['year','conc']],drop_first=True)],
              axis=1)

psm = PsmPy(X, treatment='goose', indx='id')
```

```
psm.logistic_ps(balance = False)
psm.knn_matched(matcher='propensity_score', replacement=True,
                caliper=None, drop_unmatched=True)
```

Propensity Score Matching: does it work?

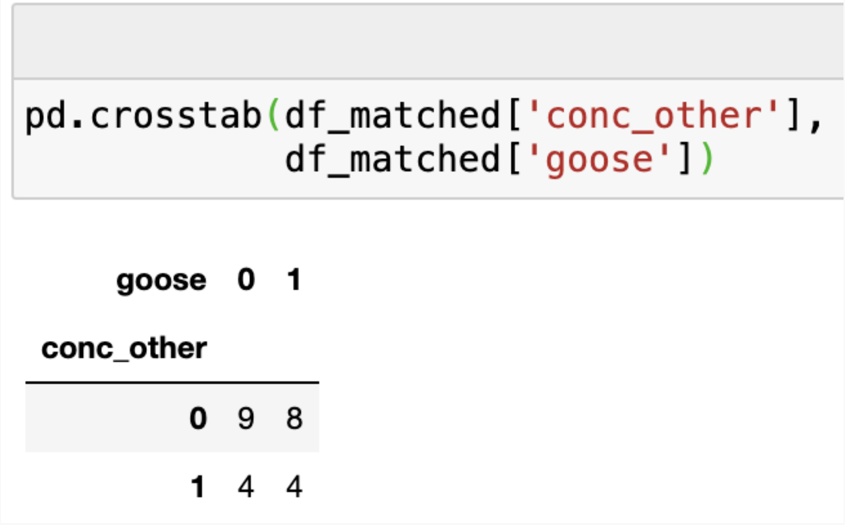






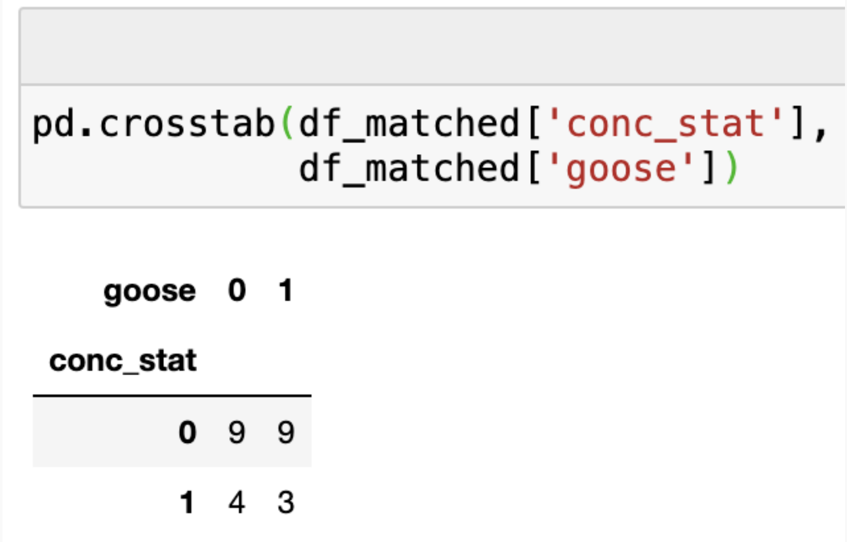
```
pd.crosstab(ai['conc'],ai['goose'])
```

```
goose  0   1
 conc
   cs  5   7
other  4  12
 stat  4   3
```



```
pd.crosstab(df_matched['conc_other'],
            df_matched['goose'])
```

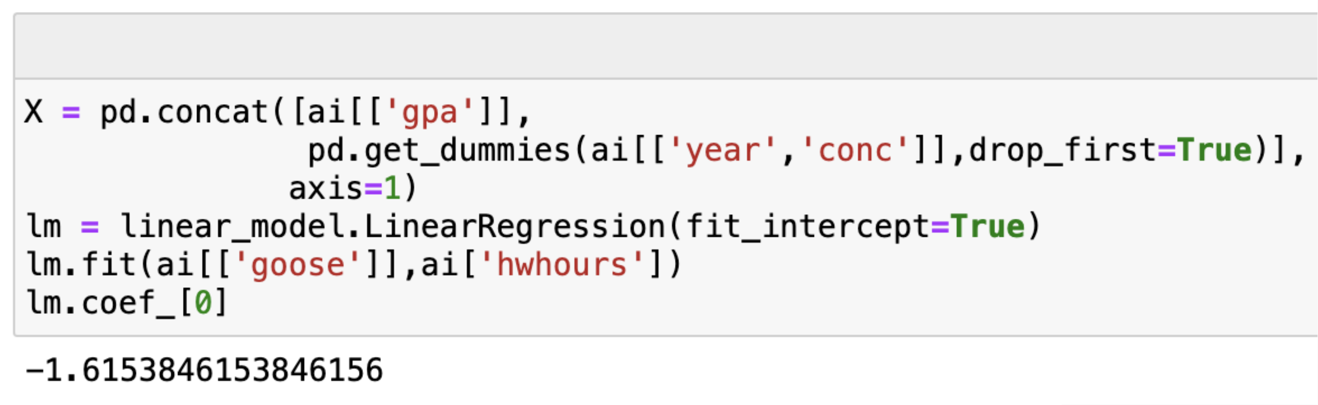
```
     goose  0  1
conc_other
         0  9  8
         1  4  4
```



```
pd.crosstab(df_matched['conc_stat'],
            df_matched['goose'])
```

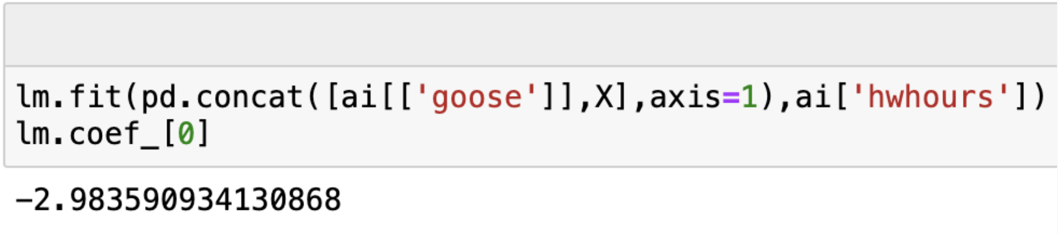
```
    goose  0  1
conc_stat
        0  9  9
        1  4  3
```

Propensity Score Matching: the estimated causal effect?



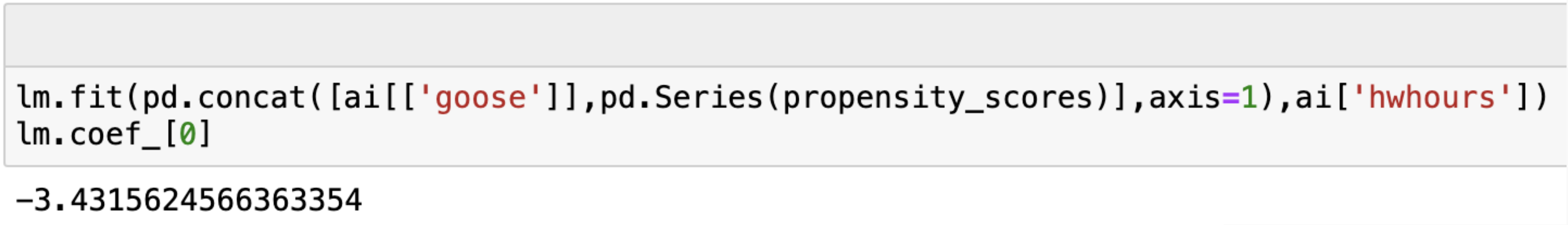
```
X = pd.concat([ai[['gpa']],
               pd.get_dummies(ai[['year','conc']],drop_first=True)],
              axis=1)
lm = linear_model.LinearRegression(fit_intercept=True)
lm.fit(ai[['goose']],ai['hwhours'])
lm.coef_[0]
```

```
-1.6153846153846156
```



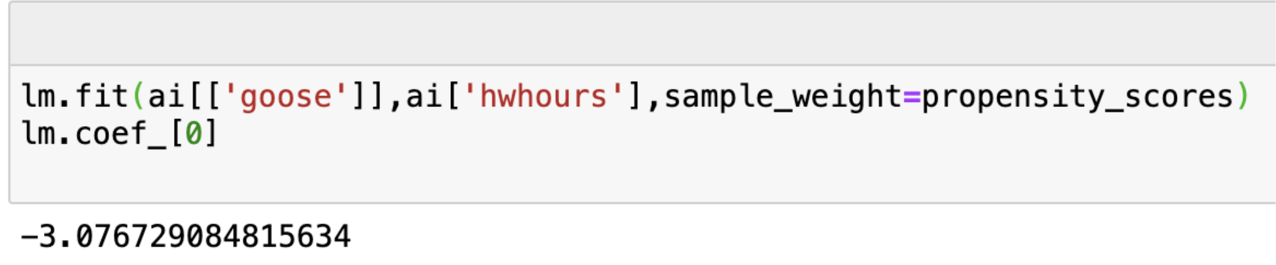
```
lm.fit(pd.concat([ai[['goose']],X],axis=1),ai['hwhours'])
lm.coef_[0]
```

```
-2.983590934130868
```



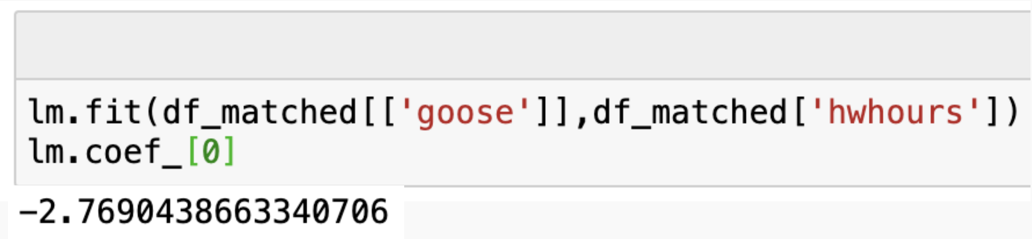
```
lm.fit(pd.concat([ai[['goose']],pd.Series(propensity_scores)],axis=1),ai['hwhours'])
lm.coef_[0]
```

```
-3.4315624566363354
```



```
lm.fit(ai[['goose']],ai['hwhours'],sample_weight=propensity_scores)
lm.coef_[0]
```

```
-3.076729084815634
```



```
lm.fit(df_matched[['goose']],df_matched['hwhours'])
lm.coef_[0]
```

```
-2.7690438663340706
```

Outline

AB Testing (Randomized, Control Trials)

Adjusting for confounders

Propensity Scores

- Estimation

- Adjusting, Weighting, and Matching

Covariate Balancing

Direct Covariate Balancing

What was the purpose of using propensity scores?

- To adjust for covariate imbalance among predictors.

Propensity scores are great, but not the only approach to achieve this goal.

We can directly match the covariates in order to balance them! This done through a linear programming algorithm.

Word of warning

Remember the main issue with observational studies

- Covariates, both measured and unmeasured, are likely to be imbalanced across treatment groups.

Using adjustments, models, propensity scores, and covariate balancing does a great job of fixing this imbalance for the measured confounders, but **does NOT guarantee** that all unmeasured confounders are balanced.

Note: it is likely to improve the unmeasured confounders balance. Why?

- Unmeasured confounders are likely correlated to those that are measured.