

Rebecca Scheidegger, Emily Padvorac, Dana Chermesh

FOOD DESERTS & BIRTH OUTCOMES

Motivation



Objective:

- Identifying the important factors impacting **low birth weight** and **preterm births** on the zip code level

Implications:

- Helping determine what **policy levers** to pull or what **demographic groups** to focus intervention programs on
- Department of Health

Low birth weight; Preterm Birth

Short-term impacts;

Long-term impacts:

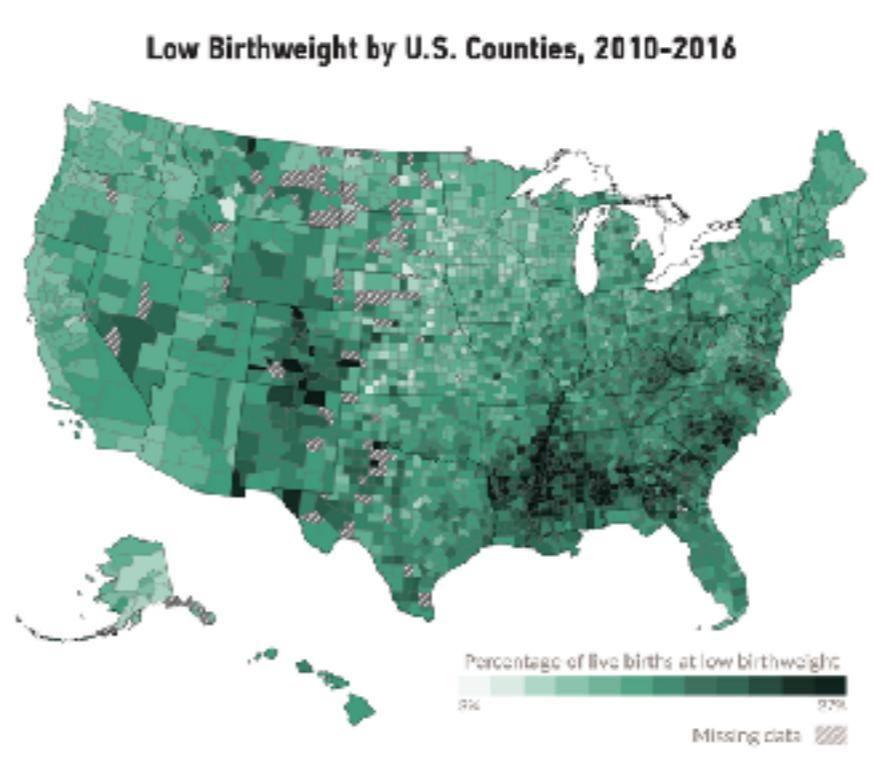
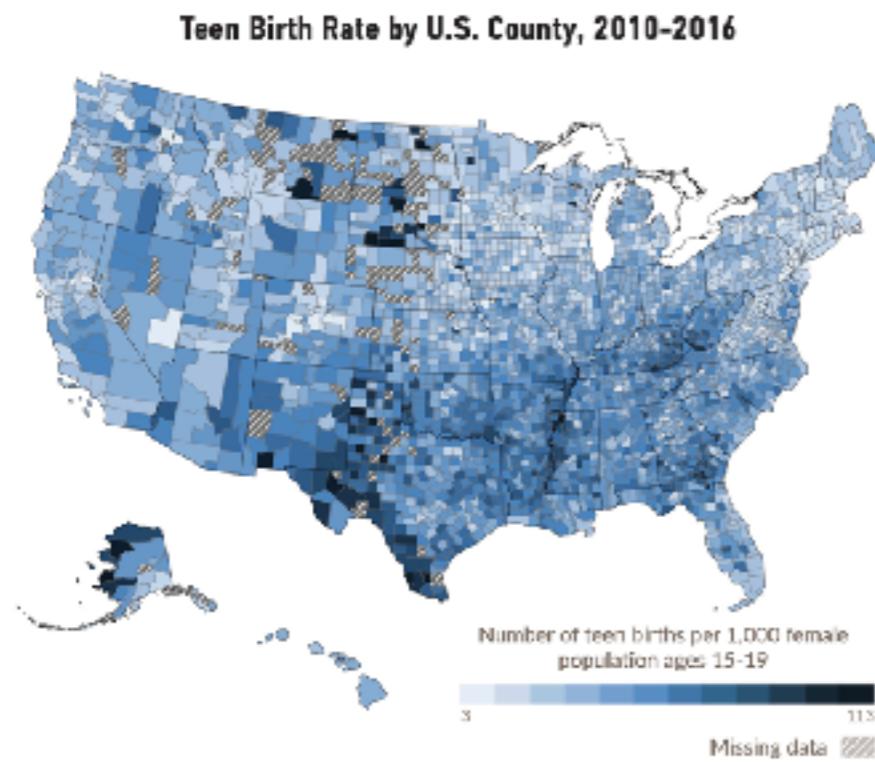
- neurological problems
- autism
- ADHD
- anxiety
- asthma
- hearing loss
- vision loss
- intestinal problems
- and more...



Low birth weight; Preterm Birth

Recent studies >>

- County / State level



- Health Variables;

- > Problematic to obtain (data confidentiality) &
- > Difficult to address (educating the public)

From: “*The Geography of Health in America*” //
Alastair Boone, CityLab; published April 23th, 2018
https://www.citylab.com/equity/2018/04/the-geography-of-health-in-america/557921/?utm_source=instagram

Food Deserts, Health, Inequality

Recent studies

Granularity of County >>
Zip code

From: "*It's not the food desert it's the inequality*" //
Richard Florida, CityLab; published January 18, 2018
<https://www.citylab.com/equity/2018/01/its-not-the-food-deserts-its-the-inequality/550793/>

Average Health Index of Store Purchases by County

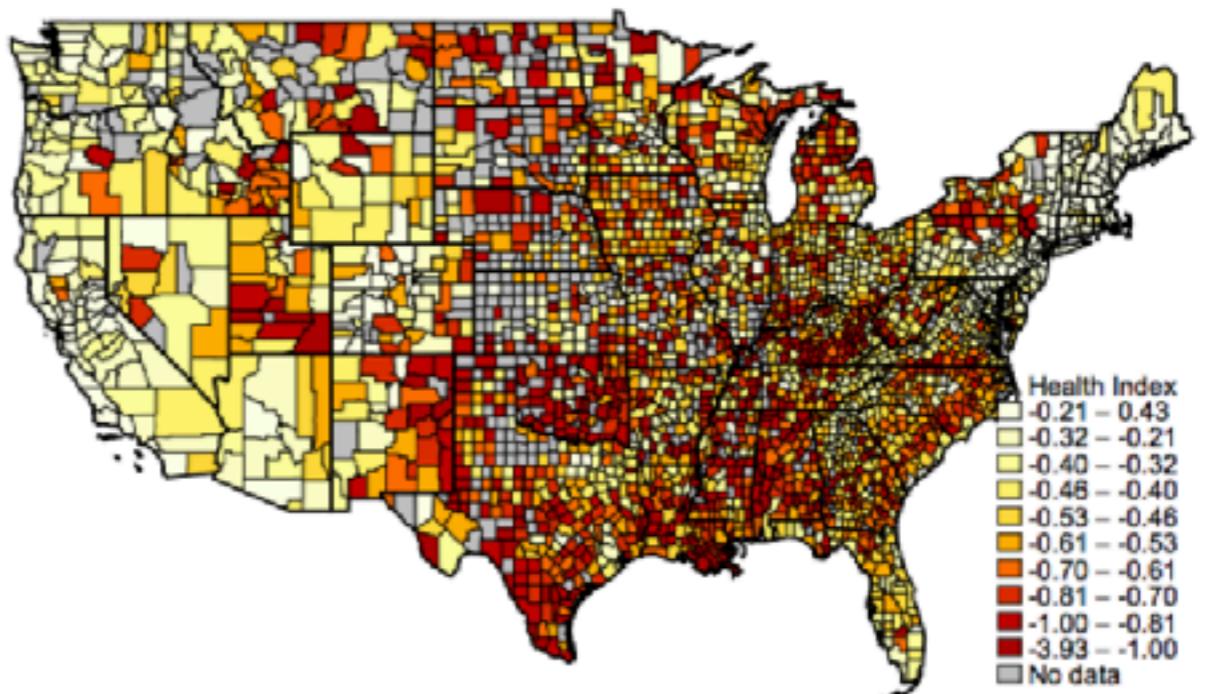
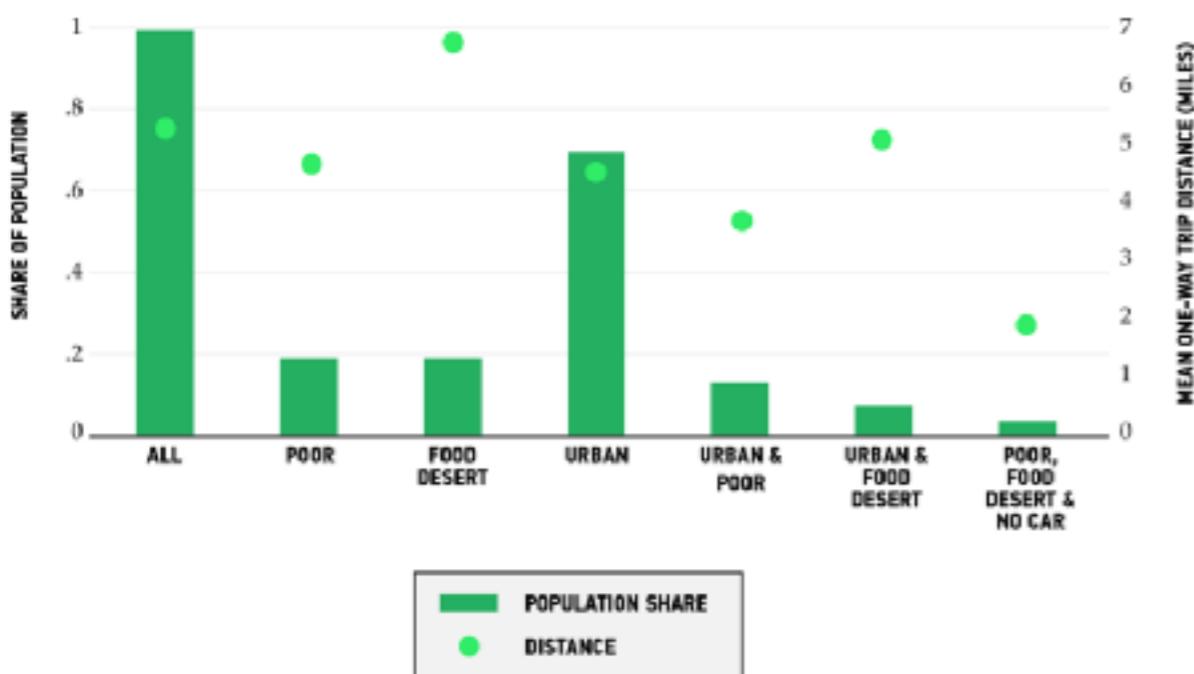


Figure 7: Shopping Trip Distances by Household Income



Researchers found that access to higher-quality groceries doesn't change eating habits as much as might be expected. // Alex Brandon/AP

It's Not the Food Deserts: It's the Inequality

RICHARD FLORIDA JAN 18, 2018

A new study suggests that America's great nutritional divide goes deeper than the problem of food access within cities.

Data

ACS data

Variables of interest:

- %Foreign
- %Uninsured
- %Vehicle access
- %Foodstamps
- %Poverty
- Urban / Rural

Health data

Independent Variables:

- %Teen Birth

Dependent Variables:

- %Low Birth Weight
- %Premature Birth

**data limitations! NYS only;
data is not available
country-wide**

Data

ACS data

Variables of interest:

- %Foreign
- %Uninsured
- %Vehicle access
- %Foodstamps
- %Poverty
- Urban / Rural

Health data

Independent Variables:

- %Teen Birth

Dependent Variables:

- %Low Birth Weight
- %Premature Birth

**** Not included:**

- Individual health data (BMI)
- Mother's age groups
- Multiple birth factor

Methodology - Supervised Learning

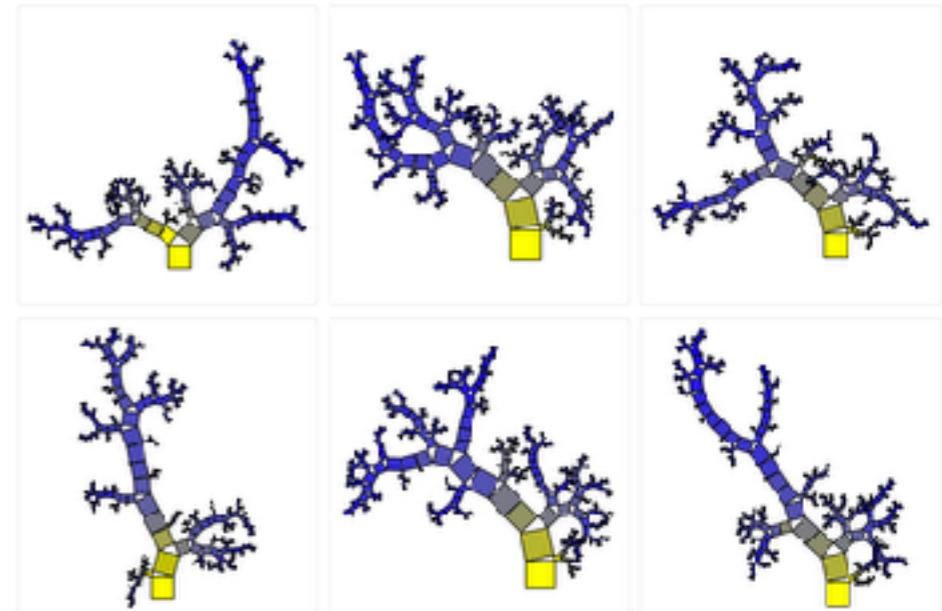
Labeled data (ACS data + birth outcomes)

Two approaches >> in order to allow results to be compared and contrasted to better understand the structure and accuracy of model choice

1

Random Forest;

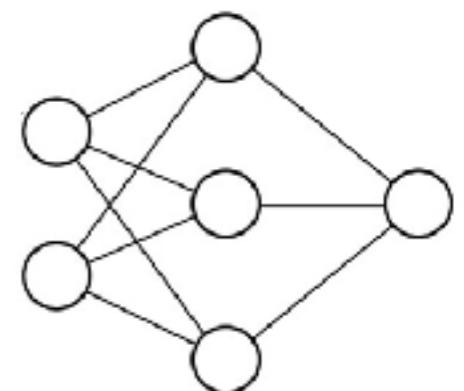
- Feature importance
- Accuracy



2

Bayesian Networks

- Learn structure of relationships between Variables



Methodology - Supervised Learning

Labeled data (ACS data + birth outcomes)

Two approaches >> in order to allow results to be compared and contrasted to better understand the structure and accuracy of model choice



1 Random Forest;

- Make **dummy variable** for the dependent variable based on std
- Test sample 30%
- Max depth tuning for best OS accuracy score
- Compute feature importance

Methodology - Supervised Learning

Labeled data (ACS data + birth outcomes)

Two approaches >> in order to allow results to be compared and contrasted to better understand the structure and accuracy of model choice

2

Bayesian Networks

- Dummy variables for all variables in the network
- Best model and HillClimbSearch

```
In [9]: for i in dfunpre:  
    dfunpre[i] = pd.qcut(dfunpre[i].rank(method='first'), 4, labels=False)  
print(dfunpre.head())
```

```
%Uninsured %Vehicle %FoodStamp %Poverty %Foreign urban \  
zipcode  
10001      1      3      3      3      3      1  
10002      2      3      3      3      3      1  
10003      1      3      3      1      3      1  
10004      1      3      3      0      3      1  
10005      1      3      3      2      3      1
```

```
%PrematureBirth  
zipcode  
10001      2  
10002      1  
10003      1  
10004      1  
10005      2
```

Results

① Random Forest

Low Birth Weight

max depth = 8

In sample accuracy: 89.95%

Out of sample accuracy: 63.9%

Premature Birth

	%Foreign	%Uninsured	%vehicle	%foodStamp	%poverty	urban	%TeenBirthRate
0	0.114901	0.19955	0.11475	0.249985	0.181518	0.013323	0.125973

Results

① Random Forest

Low Birth Weight

max depth = 8

In sample accuracy: 89.95%

Out of sample accuracy: 63.9%

Premature Birth

max depth = 2

In sample accuracy: 60.0%

Out of sample accuracy: 60.73%

	%Foreign	%Uninsured	%vehicle	%foodStamp	%poverty	urban	%TeenBirthRate
0	0.104882	0.184619	0.138065	0.250522	0.134889	0.015932	0.171092

Results

①

Random Forest

Low Birth Weight

Feature Importance:

1. % Food stamps; ~25%
2. % Uninsured; 18–19%
3. % Poverty
4. % Teen Birth Rate
5. % Foreign + Vehicle Access
6. Urban / rural

Premature Birth

Feature Importance:

1. % Food stamps; ~25%
2. % Uninsured; 18–19%
3. % Teen Birth Rate
4. % Vehicle Access + % Poverty
5. % Foreign
6. Urban / rural

for RF; accuracy ~ 61–63%:

Both negative birth outcomes showed relatively similar results

Results

② Bayes Nets

Low Birth Weight

```
hc = HillClimbSearch(train, scoring_method=BicScore(train))
best_model = hc.estimate()
print(best_model.edges())
[('urban', '%foodStamp'), ('%Foreign', 'urban'), ('%TeenBirthRate', '%poverty'), ('%poverty', '%Uninsured'), ('%poverty', '%vehicle'), ('%vehicle', '%LowBirthWeight'), ('%vehicle', '%Foreign')]
```

```
best_model.fit(train, estimator=BayesianEstimator, prior_type='K2')
for cpd in best_model.get_cpds():
    print("CPD of {variable}: ".format(variable=cpd.variable))
    print(cpd)
```

```
CPD of urban:
+-----+
| %Uninsured(2) | 0.1384083044982699 | 0.25806451612903225 | 0.3192982456140351 | 0.32989690721649484 |
+-----+
| %Uninsured(3) | 0.08650519031141868 | 0.14695340501792115 | 0.312280701754386 | 0.40893470790378006 |
+-----+
CPD of %vehicle:
+-----+
| %poverty | %poverty(0) | %poverty(1) | %poverty(2) | %poverty(3) |
+-----+
| %vehicle(0) | 0.44982698961937717 | 0.25089605734767023 | 0.20701754385964913 | 0.09621993127147767 |
+-----+
| %vehicle(1) | 0.29411764705882354 | 0.3727598566308244 | 0.23508771929824562 | 0.10309278350515463 |
+-----+
| %vehicle(2) | 0.2041522491349481 | 0.24372759856630824 | 0.34035087719298246 | 0.26804123711340205 |
+-----+
| %vehicle(3) | 0.05190311418685121 | 0.13261648745519714 | 0.21754385964912282 | 0.5326460481099656 |
+-----+
```

```
predicted_test = best_model.predict(test.loc[:, "%Uninsured": "urban"])
print "Out of sample:", (test.loc[:, "%LowBirthWeight"] == predicted_test['%LowBirthWeight']).mean()
```

Out of sample: 0.3071808510638298

Premature Birth

Results

② Bayes Nets

Low Birth Weight

```
hc = HillClimbSearch(train, scoring_method=BicScore(train))
best_model = hc.estimate()
print(best_model.edges())

[('urban', '%foodStamp'), ('%Foreign', 'urban'), ('%TeenBirthRate', '%poverty'), ('%poverty', '%Uninsured'), ('%poverty', '%vehicle'), ('%vehicle', '%PrematureBirth'), ('%vehicle', '%Foreign')]
```

```
best_model.fit(train, estimator=BayesianEstimator, prior_type='K2')
for cpd in best_model.get_cpds():
    print("CPD of {variable}: ".format(variable=cpd.variable))
    print(cpd)
```

```
| %TeenBirthRate(1) | 0.24735 |
+-----+-----+
| %TeenBirthRate(2) | 0.241166 |
+-----+-----+
| %TeenBirthRate(3) | 0.25 |
+-----+-----+
```

```
CPD of %foodStamp:
```

urban	urban(0)	urban(1)	urban(2)	urban(3)
%foodStamp(0)	0.22758620689655173	0.10469314079422383	0.48986486486486486	0.17793594306049823
%foodStamp(1)	0.30689655172413793	0.23826714801444043	0.2635135135135135	0.1494661921708185
%foodStamp(2)	0.2655172413793103	0.37184115523465705	0.10472972972972973	0.3487544483985765
%foodStamp(3)	0.2	0.2851985559566787	0.14189189189189189	0.3238434163701068

```
predicted_test = best_model.predict(test.loc[:, "%Uninsured": "urban"])
print "Out of sample:", (test.loc[:, "%PrematureBirth"] == predicted_test["%PrematureBirth"]).mean()
```

Out of sample: 0.2925531914893617

Premature Birth

Results

② Bayes Nets

Low Birth Weight

```
[('urban', '%foodStamp'), ('%Foreign', 'urban'), ('%TeenBirthRate', '%poverty'), ('%poverty', '%Uninsured'), ('%poverty', '%vehicle'), ('%vehicle', '%PrematureBirth'), ('%vehicle', '%Foreign')]
```

Premature Birth

for BN; accuracy ~ 30%:

Identical structure was learned for both negative birth outcomes

Results

Comparison RF vs BN



Higher Accuracy
(61-63%)



Accuracy (30%)

+ We want more data!

Thank You :)

