Notebook

November 21, 2018

Local date & time is : 11/21/2018 05:07:58 PST

```
In [67]: from datascience import *
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import gsExport
         %matplotlib inline
         plt.style.use('fivethirtyeight')
```

Deadline: This assignment is due Monday, November 19th at noon (12pm). Late work will not be accepted.

You will submit your solutions using both OKpy and Gradescope. You will find detailed submission instructions at the bottom of this notebook and on bCourses (here). **Please do not remove or add cells and please ignore the '#newpage' cells** (these are here to facilitate Gradescope submission).

You should start early so that you have time to get help if you're stuck. Post questions on Piazza. Check the syllabus for the office hours schedule. Remember that Connector Assistant office hours are for *coding questions only*.

1 newpage

1.1 Question 1: Orange Juice Sales

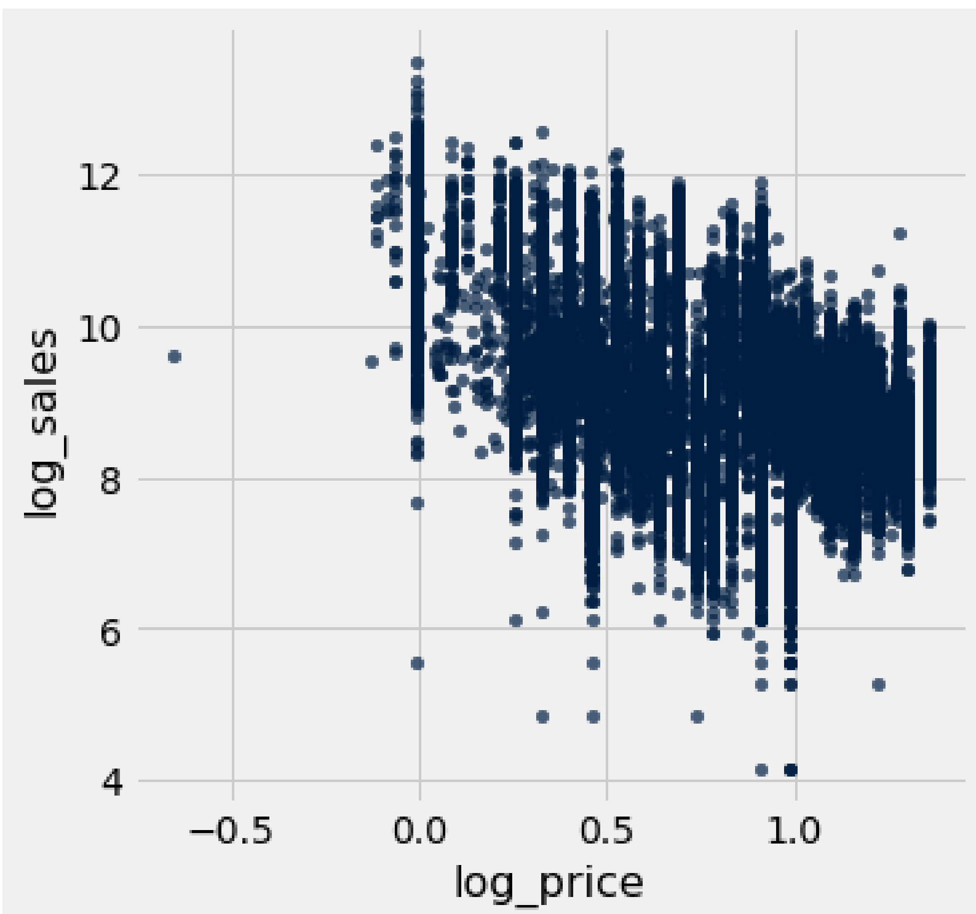
In [69]:
```
#first we'll define log_sales and log_price and add them as columns
log_sales = np.log(oj_data.column('sales'))
log_price = np.log(oj_data.column('price'))

oj_data = oj_data.with_columns(['log_sales', log_sales, 'log_price', log_price])
oj_data
```

Out[69]:
```
sales | price | brand     | feat | tropicana | log_sales | log_price
8256  | 3.87  | tropicana | 0    | 1         | 9.0187    | 1.35325
6144  | 3.87  | tropicana | 0    | 1         | 8.72323   | 1.35325
... Omitting 5 lines ...
8512  | 3.29  | tropicana | 0    | 1         | 9.04923   | 1.19089
5504  | 3.29  | tropicana | 0    | 1         | 8.61323   | 1.19089
... (19288 rows omitted)
```

a. **(3 points)** Create a scatterplot with `log_sales` on the vertical axis and `log_price` on the horizontal axis. Describe the relationship you see in a sentence.

In [70]:
```
#write code here
oj_data.scatter('log_price', 'log_sales')
```



The scatter plot has a negative non-linear relationship.

b. (5 points) Estimate the price elasticity of demand (the regression given above). Report your coefficients. Interpret your estimate for the price elasticity (β) in a full sentence. (Be sure to mention the *magnitude* of the estimate and not just the *sign*.)

```
In [71]: #write your code here: define MSE function
         def sales_short_mse (treatment_slope, intercept):
             t = oj_data.column('log_price')
             y = oj_data.column('log_sales')
             fitted = intercept + treatment_slope*t
             return np.mean((y - fitted)**2)
```

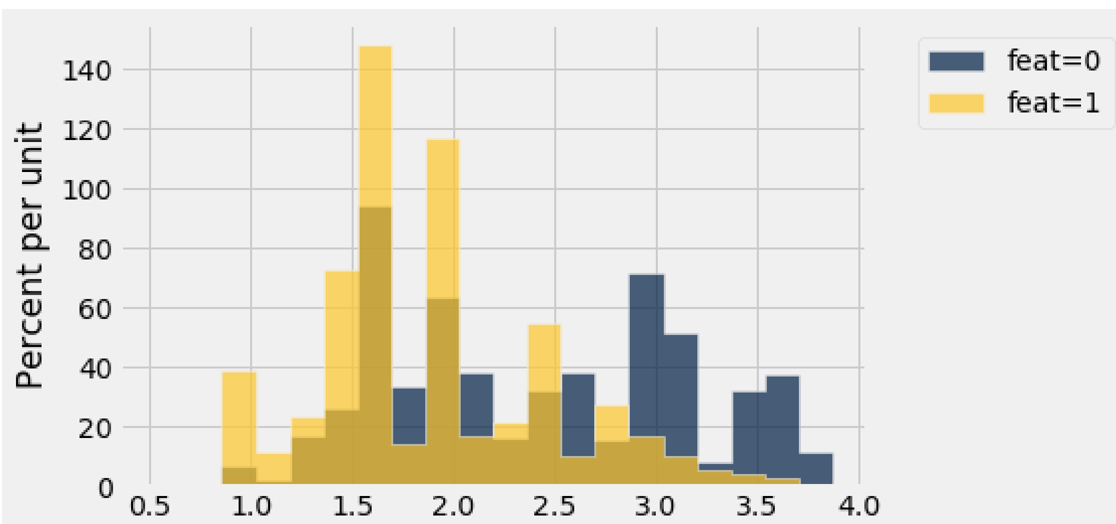
```
In [72]: #write any additional code here
         coefficients_short = minimize(sales_short_mse)
         coefficients_short
```

```
Out[72]: array([-1.34710604, 10.19495404])
```

-1.347 is the percentage change or increase(which could be a negative increase) in sales associated with one percentage increase in price of juice

c. (3 points) Plot a histogram of `price`, grouped by `feat`. How do prices for featured and non-featured orange juices compare?

```
In [73]: #write your code here
         oj_data.hist('price', group = 'feat', bins = 20)
```



Featured or promoted orange juice tended to be less expensive than the non-featured orange juice. In other words, a significant proportion(approx 60%) of the featured juices were between the first and second quintile of price range.

d. (3 points) Calculate average sales by `feat`. How do sales for featured and non-featured orange juices compare?

```
In [74]: #write your code here
         print ("average non-featured sales", np.mean(oj_data.where('feat', 0).column('sales')))
         print ("average featured sales", np.mean(oj_data.where('feat', 1).column('sales')))
```

```
average non-featured sales 10837.584330222164
average featured sales 39243.56513222331
```

Featured orange juices did an average of approximately 28000 in sales than the non-featured juices. Featured had more market demand.

e. (3 points) How do you anticipate controlling for `feat` in the regression will affect your estimated price elasticity? Why?

I anticipate that controlling for feat in the regression will make our estimated price elasticity less positive. In other words it'll decrease the price difference in sales of both non-featured and featured. In terms of the omitted variable bias, I will argue that feat is negatively associated with our treatment-price(When goods are promoted, they do so for the 'good deals' aka lower prices for same or more quantity, being offered e.g McDonalds). On the other hand, feat is positively associated with Sales. My reasoning is that after promotions, sales are likely to increase. So then multiplying the negative and positive gives a negative sign for the omitted variable bias.

f. (5 points) Using regression, estimate the price elasticity of demand while *controlling for* `feat`. Report your coefficients. Interpret your estimate for the price elasticity in a sentence. (Be sure to mention the *magnitude* of the estimate and not just the *sign*.)

```
In [75]: #write your code here: define MSE function
         def sales_long_mse (treatment_slope, slope1, intercept):
             t = oj_data.column('log_price')
             x1 = oj_data.column('feat')
             y = oj_data.column('log_sales')
             fitted = intercept + treatment_slope*t + slope1*x1
             return np.mean((y - fitted) ** 2)
```

```
In [76]: #write any additional code here
         coefficients_long = minimize(sales_long_mse)
         coefficients_long

Out[76]: array([-1.00115365,  0.95231601,  9.72330351])
```

A one percentage increase in price is associated with a -1.00 increase in sales.

g. (4 points) Using regression, estimate the price elasticity of demand while *controlling for* `tropicana` (but not `feat`). Report your coefficients.

```
In [77]: #write your code here: define MSE function
         def tropicana_sales_long_mse (treatment_slope, slope1, intercept):
             t = oj_data.column('log_price')
             x1 = oj_data.column('tropicana')
             y = oj_data.column('log_sales')
             fitted = intercept + treatment_slope*t + slope1*x1
             return np.mean((y - fitted)**2)
```

```
In [78]: #write any additional code here
         coefficients_long_tropicana = minimize(tropicana_sales_long_mse)
         coefficients_long_tropicana

Out[78]: array([-3.07308561,  1.49663935, 10.79424941])
```

h. (5 points) What do your regression results in **part (b)** and **part (g)** tell you about the difference in average `log_price` between Tropicana and Dominic's brand orange juice? Why?

Tropicana, because of its higher prices is positively related to our treatment-Price. However, because of price elasticity, it is negatively related to our outcome-Sales. Hence Tropicana's omitted variable bias has a negative sign. With Dominic's it's positive on our treatment and outcome.So Dominic's OVB is positive.

i. (4 points) Prices tend to be higher in supermarkets located in more dense areas of Chicago. These supermarkets also tend to attract more customers per day. Explain why the fact that we are not controlling for the density of the supermarket's location may bias our estimate for the price elasticity of demand. What sign do you anticipate for that bias, and why?

Not controlling for the density of the supermarket's location may bias our estimate for the price elasticity because location density is very likely to be positively related with both price and sales. Thus giving us a positive sign for omitted variable bias. By implication, we'd get the estimate that shows orange juices to be very price inelastic.

2 newpage

2.1 Question 2: Regression and the Oregon Health Study

In Problem Set 5 we estimated the causal effect of winning the lottery on `cost_any_owe` by comparing the mean of `cost_any_owe` for lottery winners to the same mean for lottery losers. Recall that we can make that same comparison using regression by estimating the following model:

$$\text{cost\_any\_owe}_i = \alpha + \beta \times \text{win\_lottery}_i + e_i$$

a. (4 points) Estimate the regression above. Report the coefficients.

```
In [80]: #write your code here: define MSE function
         def cost_short_mse(treatment_slope, intercept):
             t = ohs_data.column('win_lottery')
             y = ohs_data.column('cost_any_owe')
             fitted = intercept + treatment_slope*t
             return np.mean((y - fitted) ** 2)
```

```
In [81]: #write any additional code here
         coefficients_short = minimize(cost_short_mse)
         coefficients_short
```

```
Out[81]: array([-0.06927365,  0.61554309])
```

b. (3 points) Confirm that you get the same treatment effect estimate by comparing means of `cost_any_owe` for lottery winners and lottery losers. Make sure to print your calculation of the difference in means.

```
In [82]: #write code here
         print("average winners cost_any_owe", np.mean(ohs_data.where('win_lottery', 1).column('cost_an
         print("average non-winners cost_any_owe", np.mean(ohs_data.where('win_lottery',0).column('cost_
         print("average difference", np.mean(ohs_data.where('win_lottery', 1).column('cost_any_owe')) -
               np.mean(ohs_data.where('win_lottery', 0).column('cost_any_owe')))
```

```
average winners cost_any_owe 0.5462694437121013
average non-winners cost_any_owe 0.6155430925947122
average difference -0.0692736488826109
```

c. (4 points) Describe the interpretation for yourβestimate in a sentence. (Be sure to mention the *magnitude* of the estimate and not just the *sign*.) Is this a causal effect? Why or why not?

A one percentage point increase in winning the lottery for lottery winners is associated with a - 0.06 change in medical expenses debt. It is a 'naive' causal estimate because we have not accounted for all relevant biases. Hence, it is not a definitive causal effect at this level of our investigation.

d. (4 points) Now we'll try estimating a similar regression but using our other covariates, `female`, `age`, `english`, `zip_msa`, as controls.

Estimate the following regression model:

$$\text{cost\_any\_owe}_i = \alpha + \beta \times \text{win\_lottery}_i + \gamma_1 \times \text{female}_i + \gamma_2 \times \text{age}_i + \gamma_3 \times \text{english}_i + \gamma_4 \times \text{zip\_msa}_i + \epsilon_i$$

Be sure to report the coefficients.

```
In [83]: #write your code here: define MSE function
         def cost_longer_mse(treatment_slope, slope1, slope2, slope3, slope4, intercept):
             t = ohs_data.column('win_lottery')
```

```
    x1 = ohs_data.column('female')
    x2 = ohs_data.column('age')
    x3 = ohs_data.column('english')
    x4 = ohs_data.column('zip_msa')
    y = ohs_data.column('cost_any_owe')
    fitted = intercept + treatment_slope*t + slope1*x1 + slope2*x2 + slope3*x3 + slope4*x4
    return np.mean((y - fitted)**2)
```

```
In [84]: #write any additional code here
         coefficients_long = minimize(cost_longer_mse)
         coefficients_long
```

```
Out[84]: array([-0.06873784,  0.06148294, -0.00091316,  0.18550344, -0.0189371 ,
                 0.45360977])
```

e. (5 points) Your estimate forβshould not meaningfully change from **part (a)** to **part (d)**. What does this tell you about the relationship between `win_lottery` and the other covariates? Do you find this result surprising? Why or why not?

My beta estimate does not vary meaningfully and I don't find that surprising. This tells me the other covariates presents no statistically signifant selection and omitted variable bias to win_lottery and cost_any_owe. It means, win_lottery is comparable across other covariates, and is randomly associated with owed medical expenses.

3 newpage

3.1 Submission

```
In [ ]: from client.api.notebook import Notebook
        ok = Notebook('pset8.ok')
        _ = ok.auth(inline=True)
        _ = ok.submit()
```

```
=====================================================================
Assignment: Pset 8
OK, version v1.12.5
... Omitting 0 lines ...

Successfully logged in as jwonder001@berkeley.edu


<IPython.core.display.Javascript object>
```

```
In [ ]: gsExport.generateSubmission('pset8.ipynb')
```

```
In [ ]:
```