Notebook

November 8, 2018

Local date & time is : 11/08/2018 11:57:42 PST

Deadline: This assignment is due Monday, November 5th 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: Home Renovations

(45 points) A realtor has hired you as a consultant to analyze data on the San Francisco housing market. A home that the realtor is selling has a large open living space, and the realtor is considering renovating the space by adding an additional bedroom to the home. This renovation would not increase the total square footage of the home, but would increase the number of bedrooms in the home from **one** to **two**. The realtor wants to know how much this renovation will *increase the value of the home*.

In other words, the causal question of interest is: how much does adding an additional bedroom to an existing home (without adding any square footage) increases the value of the home?

You will not run an experiment to answer this question -- that would be way too expensive. Instead, you will analyze obsevational data. The data you will use describe 1128 homes that were sold in San Francisco in the last 6 months which had between one and three bedrooms (the data are from Redfin). The data include 326 one bedroom homes, 498 two bedroom homes, and 304 three bedroom homes. For each home, the data include the following information:

- `hometype`: the type of property (condo, single family home, etc.)
- `location`: the neighborhood
- `price`: the price the home was sold for
- `sqft`: square footage of the home
- `beds`: number of bedrooms
- `baths`: number of bathrooms
- `yearbuilt`: the year the home was built

Run the cell below to read in the data.

a. (3 points) What is the meaning ofY_{1i}in this context?

The potential outcome in the value of the house, given that an additional room is added.

b. (3 points) Calculate average price by number of bedrooms. What is the difference between the average price of a one bedroom home and the average price of a two bedroom home?

```
In [51]: #write code here
         room_average = sf_homes.group('beds', np.mean).select('beds', 'price mean').take([0,1])
         print ('difference', room_average.column('price mean').item(1)- room_average.column('price mea
```

```
difference 594962.3296868455
```

c. (3 points) She argues that to calculate the value of adding an additional bedroom, you should control for `sqft` in your analysis. What is the reasoning behind her argument? (Be sure to cite the evidence provided in the table above.)

From the table above, there's a average area of a house that correspons to a given number of rooms in a house. In this case, it's obvious that the square area of the house is also an important covariate in its price determination. It has an effect on the treatment(number of room) and also on the outcome(price)

d. (5 points) For this question you will follow your partner's advice. Match each one bedroom home to a two bedroom home with similar square footage. You will not be able to find exact matches for each home. Instead, match each one bedroom home to the two bedroom home with the *smallest absolute difference in square footage* among all two bedroom homes. What is the difference in average price between the one bedroom homes and the matched two bedroom homes?

```
In [54]: #write your match function here
         def match_twobed_id(twobed_value):
             twobed_sq_match = abs(twobed.column('sqft') - twobed_value )
             twobed_add = twobed.with_column('two_match', twobed_sq_match)
             sq_min = min(twobed_sq_match)
             final_match = twobed_add.where('two_match', min(twobed_sq_match))
             return final_match.column('id').item(0)
```

```
In [55]: #apply your match function here
         match_indices = onebed.apply(match_twobed_id, 'sqft' )
         match_indices

Out[55]: array([ 32,  65,  57,  76,   0,  65,  52,  52,  76,  31,  13, 163,  76,
               111,  52,  52,  52,  52,  64,  76,  37, 417,  64,  52,  82,  80,
                52,  52, 230,  65, 197,  53,  17,  53,  52,  52,   0,  52,  53,
         ... Omitting 19 lines ...
                42,  50,  42,  44, 344, 385, 147,  51,   4,  25, 340, 292, 214,
                25,  29,  82, 183,  33, 215,  32, 291, 323, 322,  43,  32, 153,
                47])
```

```
In [56]: #print results here
         matched_twobed = twobed.take(match_indices)
         matched_twobed.show(5)
         print("onebed average price", np.mean(onebed.column('price')))
         print("matched twobed average price", np.mean(matched_twobed.column('price')))
         print("price difference", np.mean(onebed.column('price'))- np.mean(matched_twobed.column('pric

<IPython.core.display.HTML object>


onebed average price 917311.8128834356
matched twobed average price 987080.6104294478
price difference -69768.79754601221
```

e. (5 points) What do your findings in **part (c)** and **part (d)** imply about the sign of the correlation between beds and sqft?

There appears to be a somewhat positive correlation between beds and square feet. The number of bedrooms on average vary in direct proportion to their square feet values.

f. (5 points) Your partner is also concerned that one bedroom homes may be concentrated in neighborhoods that are more or less attractive than the neighborhoods where two bedroom homes are concentrated. In this question, you will control for neighborhood in addition to sqft.

For each one bedroom, find a two bedroom match in the same neighborhood, and with the smallest absolute difference in sqft within that neighborhood. What is the difference in average price between the one bedroom homes and the *matched* two bedroom homes?

```
In [136]: #write your match function here
          def match_hoodsqft_id(hood_value, sq_value):
              match_hood = twobed.where('neighborhood', hood_value)
              sq_match = abs(match_hood.column('sqft')- sq_value)
              min_sq = min(sq_match)
              fin_match = match_hood.with_column('sqft_match', min_sq)
              return fin_match.column('id').item(0)
```

```
In [158]: #apply your match function here
          matched_hoodsq_indices = onebed.apply(match_hoodsqft_id, 'neighborhood','sqft')
          matched_hoodsq_indices

Out[158]: array([ 83, 126,   7,  23,   0,  26,  96, 388, 145,   0,   0,  13, 245,
                  70, 245, 142, 142,  23,  23,  23,  23,  23, 104,  43,  93,   1,
                  70,  70,  23,  96,  98,  96,  96,  98,  98,  17,  44,  44,  44,
          ... Omitting 19 lines ...
                  98,  98,  98,  98,  83, 294, 294,  83, 266, 266,  24,  26, 172,
                  96,  98, 104,  17,  70,  98,  98,  98,  12,   2,  98,  98,   4,
                  98])
```

```
In [149]: #print results here
          matched_hoodsqft = twobed.take(matched_hoodsq_indices)
          matched_hoodsqft.show(5)
          print("onebed average price", np.mean(onebed.column('price')))
          print("matched hoodsqft average price", np.mean(matched_hoodsqft.column('price')))
          print("abs price difference", np.mean(onebed.column('price'))- np.mean(matched_hoodsqft.colum

<IPython.core.display.HTML object>


onebed average price 917311.8128834356
matched hoodsqft average price 1091352.8496932515
abs price difference -174041.0368098159
```

g. (5 points) For both sets of matches you construct (in **part (d)** and **part (f)**), calculate and print the average absolute difference in `sqft` across all matched pairs of homes. For example, the first home in `onebed` is 847 square feet. Suppose the two bedroom match you found in **part (f)** for that home is 1054 square feet. Then the absolute difference for that pair is$|847 - 1054| = 207$(square feet).

```
In [155]: #write code here for match from part (d)
          abs(onebed.column('sqft')-matched_twobed.column('sqft'))

Out[155]: array([   3,    9,    1,   11,    1,    2,    2,  221,   10,    1,    1,
                    0,    7,    3,  133,   13,   45,  211,    3,    1,    5,    2,
                    3,   24,    2,    4,    1,  156,    0,    3,    2,    6,    8,
          ... Omitting 23 lines ...
                    2,    2,    1,    1,    0,    1,    1,    0,    2,   10,    1,
                    5,    0,    0,  214,    5,    0,    2,    2,    0,    5,    1,
                    1,    0,    0,    2,    0,    1,    1])

In [156]: #write code here for match from part (f)
          abs(onebed.column('sqft') - matched_hoodsqft.column('sqft'))

Out[156]: array([ 383,  553,  249,  149,    1,  516,  448,  768,  593,   65,    7,
                  276,  269,   28,  479,  537,  569,  433,   12,  137,   83,  296,
                  356,  427,   24,   72,  245,  400,  307,  400,  457,  426,  354,
          ... Omitting 23 lines ...
                  597,  597,  711,  663,  710,  637,  284,  475,  747,  440,  450,
                  530,  340,  257, 2079,  180,  332,  363,  432,   55,  512,  666,
                  271,   41,  865,  421,  667,   33,   16])
```

h. (3 points) You should find that the average absolute difference is larger for the match from **part (f)**, meaning that matched homes are less comparable on `sqft` when you match on both `neighborhood` and `sqft` than when you match on just `sqft`. Why is this expected?

You would expect this because of relationship between neighborhood and square ft. Houses with similar square feet will differ largely in price in an affluent neighborhood from the same house in low income neighborhood. While one would think that matching on neighborhood and square feet will reduce the systematic differences, it further creates an imbalance and makes the differences larger.

i. (3 points) Compute difference in age between one bedroom and matched two bedroom apartments (use your first match from **part (d)**).

```
In [160]: #write code here
          abs(onebed.column('age') - matched_hoodsqft.column('age'))
```

```
Out[160]: array([  8,  68,   0,   7, 114,   3,   3,  79,  25, 100, 109,  97,  24,
                  42,  21,  24,  17,  90,   7,   4,   1,   3, 108,  84,  43,  34,
                  35,  35,   4,   3,  88,   8,   8,   0,  13,  15,  26,   0,   0,
          ... Omitting 19 lines ...
                   4,  12,   5,  13,  19,   0,   0,  95,  23,  58,  88,  12,  16,
                   4,   9,  49,  67,  19,  11,  11,  97,  14,  15,  67,  14,  76,
                  87])
```

```
In [161]: print("onebed average age", np.mean(onebed.column('age')))
          print("matched twobed average age", np.mean(matched_twobed.column('age')))
          print("age difference", np.mean(onebed.column('age'))- np.mean(matched_twobed.column('age')))
```

```
onebed average age 38.50920245398773
matched twobed average age 45.03680981595092
age difference -6.5276073619631845
```

j. (5 points) Based on results in **part (i)**, how do you think controlling for home age (in addition to square footage) would affect the difference in price you measure? Why?

I think it'll reduce the difference in price estimate.

k. (5 points) Recall that we are ultimately interested in measuring the average causal effect of adding an additional bedroom to an existing home, holding total square footage fixed. Suppose we were to control for `sqft`, `neighborhood`, and `age` and estimate this average causal effect using the difference in selling price between one bedrooms and matched two bedrooms. What is the **selection on observables** assumption that would be required for this approach to generate an unbiased estimate for the average causal effect of interest? Does this assumption seem plausible here? Why or why not?

The selection on observables assumption would be that there are no other ommitted variable bias to estimating our price difference other than the ones we have controlled for. There could be other unobervable omitted variable bias. we are not controlling for.

2 newpage

2.1 Submission

```
In [ ]: from client.api.notebook import Notebook
        ok = Notebook('pset7.ok')
        _ = ok.auth(inline=True)
        _ = ok.submit()
```

```
======================================================================
Assignment: Pset 7
OK, version v1.12.5
... Omitting 0 lines ...

Successfully logged in as jwonder001@berkeley.edu
```

```
In [ ]: gsExport.generateSubmission('pset7.ipynb')
```