BW_141b_hw3

October 24, 2018

1 STA 141B: Homework 3

Fall 2018

1.1 Information

After the colons (in the same line) please write just your first name, last name, and the 9 digit student ID number below.

First Name: Bailey
Last Name: Wang
Student ID: 914955801

1.2 Instructions

1.2.1 New item: Please print your answer notebook to pdf (make sure that it is not too many pages, > 10, due to long output) and submit as the homework solution with your zip file.

We use a script that extracts your answers by looking for cells in between the cells containing the exercise statements. So you

- MUST add cells in between the exercise statements and add answers within them and
- MUST NOT modify the existing cells, particularly not the problem statement

To make markdown, please switch the cell type to markdown (from code) - you can hit 'm' when you are in command mode - and use the markdown language. For a brief tutorial see: https://daringfireball.net/projects/markdown/syntax

1.2.2 Introduction

The US Department of Agriculture publishes price estimates for fruits and vegetables online. The most recent estimates are based on a 2013 survey of US retail stores.

The estimates are provided as a collection of MS Excel files, with one file per fruit or vegetable. The `hw3_data.zip` file contains the fruit and vegetable files in the directories `fruit` and `vegetables`, respectively.

Exercise 1. Use pandas to extract the "Fresh" row(s) from the fruit Excel files. Combine the data into a single data frame. Your data frame should look something like this:

type	food	form	price_per_lb	yield	lb_per_cup	price_per_cup
fruit	watermelon	Fresh1	0.333412	0.52	0.330693	0.212033
fruit	cantaloupe	Fresh1	0.535874	0.51	0.374786	0.3938
vegetables	onions	Fresh1	1.03811	0.9	0.35274	0.406868
...						

It's okay if the rows and columns of your data frame are in a different order. These modules are especially relevant:

- `str` methods
- `os`
- `os.path`
- pandas: `read_excel()`, `concat()`, `.fillna()`, `.str`, plotting methods

Ask questions and search the documentation/web to find the functions you need.

In [1]:
```
#Collabarated with Tiffany Chen

import os
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

def fruit_function(file_name, subfile_name):

    """Takes in a file path and the subfile which is the food type into the function.
     The function cleans the data by concating the food types into a new dataframe.
     The function also subsets the data into "fresh" and "fresh1" food types.

    Args:
        file_name: str
        subfile_name: str

    Returns:
        fruit1: Pandas dataframe
    """

    fruit_path = os.path.join(file_name, subfile_name)
    ls_fruit_file = os.listdir(fruit_path)

    fruit1 = pd.DataFrame()

    for file in ls_fruit_file:
        fruit2 = pd.read_excel(fruit_path + "/" + file, header = 1)
        fruit2["food"] = file.split(".")[0]
        fruit2["type"] = subfile_name
        fruit1 = pd.concat([fruit2, fruit1],sort = True)
```

```
    fruit1 = fruit1[fruit1["Form"].str.contains("Fresh").fillna(False)]

    fruit1 = fruit1.drop(columns = ["Unnamed: 2",
                                    "Unnamed: 5",
                                    "Unnamed: 7",
                                    "Unnamed: 8"])
    fruit1 = fruit1.rename(columns = {"Average price": "price_per_cup",
                                      "Average retail price ": "price_per_lb",
                                      "Preparation": "yield",
                                      "Size of a ": "lb_per_cup",
                                      "Form": "form"})
    fruit1 = fruit1[["type", "food", "form",
                     "price_per_lb",
                     "yield",
                     "lb_per_cup",
                     "price_per_cup"]]
    fruit1= fruit1.reset_index(drop=True)
    return(fruit1)
```

```
In [2]: fruit3 = fruit_function("hw3_data/", "fruit") #call for fruits
        fruit3.head()
```

Out[2]:

	type	food	form	price_per_lb	yield	lb_per_cup	price_per_cup
0	fruit	watermelon	Fresh1	0.333412	0.52	0.330693	0.212033
1	fruit	tangerines	Fresh1	1.37796	0.74	0.407855	0.759471
2	fruit	strawberries	Fresh1	2.35881	0.94	0.31967	0.802171
3	fruit	raspberries	Fresh1	6.97581	0.96	0.31967	2.32287
4	fruit	pomegranate	Fresh1	2.17359	0.56	0.341717	1.32634

There are 24 rows in fruit data.

Exercise 2. Reuse your code from exercise 1.1 to extract the "Fresh" row(s) from the vegetable Excel files.

Does your code produce the correct prices for tomatoes? If not, why not? Do any other files have the same problem as the tomatoes file?

You don't need to extract the prices for these problem files. However, make sure the prices are extracted for files like asparagus that don't have this problem.

```
In [3]: #Collabarated with Jared Yu

        vegetable1 =fruit_function("hw3_data/", "vegetables") #call for vegetables

        vegetable1.loc[17,'food']='green_cabbage' #rename food type
        vegetable1.loc[18,'food']='red_cabbage'
        vegetable1.loc[24,'food']='unpeeled_cucumber'
        vegetable1.loc[25,'food']='peeled_cucumber'

        vegetable1.head()
```

Out[3]:

	type	food	form	price_per_lb	yield	lb_per_cup	price_per_cup
0	vegetables	turnip_greens	Fresh1	2.47175	0.75	0.31967	1.05353
1	vegetables	tomatoes	Fresh	NaN	NaN	NaN	NaN
2	vegetables	sweet_potatoes	Fresh1	0.918897	0.811301	0.440925	0.4994
3	vegetables	summer_squash	Fresh1	1.63948	0.7695	0.396832	0.84548
4	vegetables	spinach	Fresh1	NaN	NaN	NaN	NaN

There are 33 rows in the vegetable data.

The vegetables with N/A values are tomatoes, spinach, mushrooms, lettuce_romaine, celery, cauliflower, broccoli, and carrots. In total there are 8 rows with errors.

Exercise 3. Remove rows without a price from the vegetable data frame and then combine the fruit and vegetable data frames. Make sure all columns of numbers are numeric (not strings).

In [4]:
```
#Collabarated with Jared Yu

vegetable2 = vegetable1.dropna() #drops na values in the dataframe
food1 = [fruit3, vegetable2]
food2 = ["price_per_cup", "price_per_lb", "yield", "lb_per_cup"]
food3 = pd.concat(food1)
for col in food2:
    food3[col] = food3[col].astype(float) #in the for loop,
food3.head()                              #converts the listed objects into floats
```

Out[4]:

	type	food	form	price_per_lb	yield	lb_per_cup	price_per_cup
0	fruit	watermelon	Fresh1	0.333412	0.52	0.330693	0.212033
1	fruit	tangerines	Fresh1	1.377962	0.74	0.407855	0.759471
2	fruit	strawberries	Fresh1	2.358808	0.94	0.319670	0.802171
3	fruit	raspberries	Fresh1	6.975811	0.96	0.319670	2.322874
4	fruit	pomegranate	Fresh1	2.173590	0.56	0.341717	1.326342

There are 25 vegetables after dropping the N/A values.

There are 49 rows for the table with fruits and vegetables.

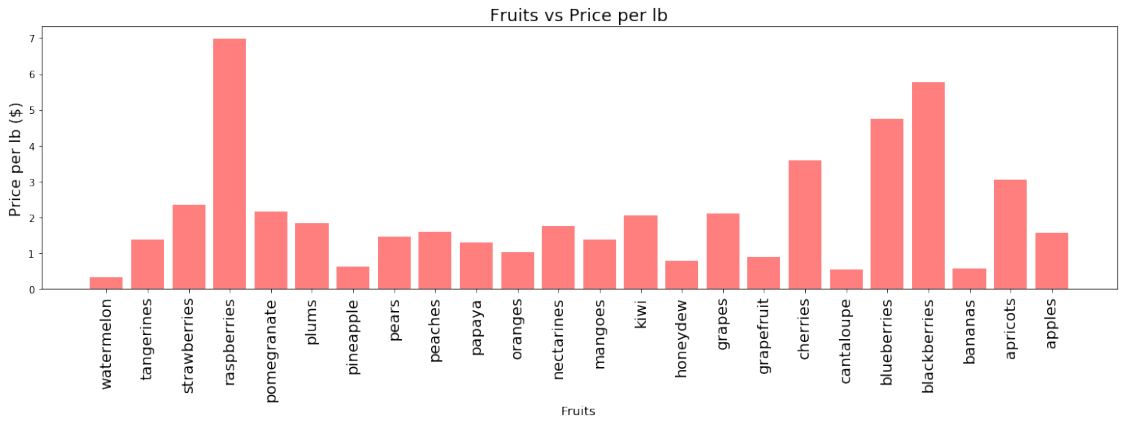
Exercise 4. Discuss the questions below (a paragraph each is sufficient). Use plots to support your ideas.

- What kinds of fruits are the most expensive (per pound)? What kinds are the least expensive?
- How do the price distributions compare for fruit and vegetables?
- Which foods are the best value for the price?
- What's something surprising about this data set?
- Which foods do you expect to provide the best combination of price, yield, and nutrition? A future assignment may combine this data set with another so you can check your hypothesis.

```
In [6]: fruit4 = food3[food3["type"] == "fruit"] #creates subgroup for only fruits

        x_pos = np.arange(len(fruit4)) #creates tick locations for all the foods
        plt.bar(fruit4["food"], fruit4["price_per_lb"],
                    align='center',alpha=.5, color='red') #creates the graph
        plt.xticks(x_pos, fruit4["food"], rotation=90, fontsize=16) #writes the labels
        plt.xlabel('Fruits', fontsize=13) #creates xlabel
        plt.ylabel('Price per lb ($)', fontsize=16) #creates ylabel
        plt.title('Fruits vs Price per lb',fontsize=18) #creates title
        plt.rcParams['figure.figsize'] = [20, 5] #changes graph size

        plt.show() #Prints graph
```

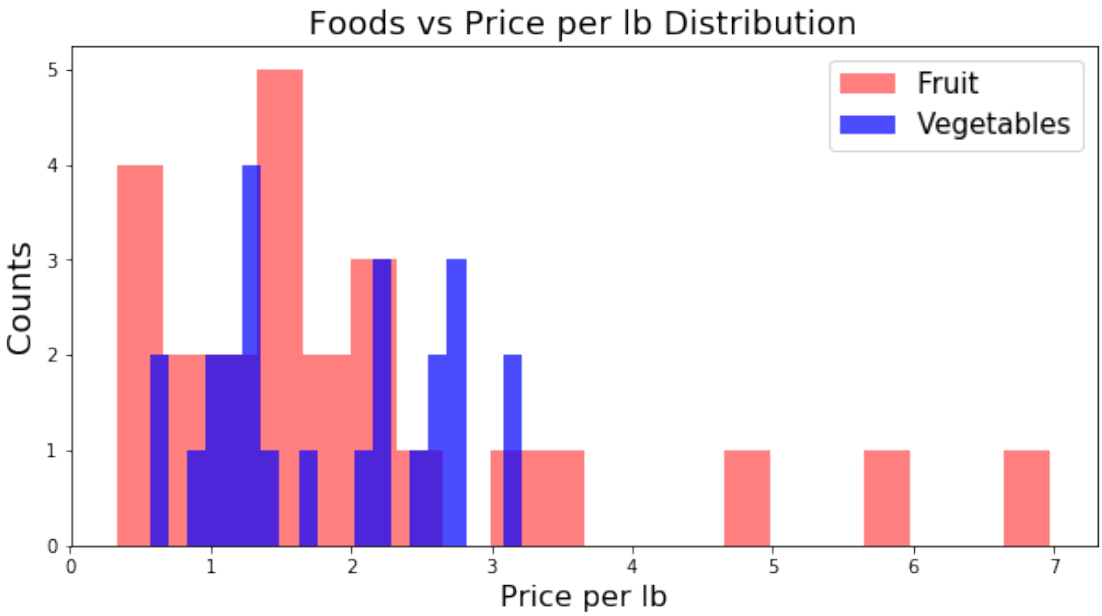


Most Exepensive Fruits:
Raspberries 6.97 per pound, Blackberries 5.77 per pound, Blueberries 4.73 per pound
Least Exepensive Fruits:
Watermelon .33 per pound, Cantaloupe .53 per pound, Bananas .56 per pound

Raspberries are the most expensive fruits. Interestly, the reason raspberries are so expensive is due to the fact that it has high proudction cost and low yield. Also it can only grow in certain areas. Watermelon are the least expensive fruits. These fruits contain a lot of mass therefore, it will take less take to harvest watermelons compared to berries.

```
In [8]: plt.hist(food3[food3["type"]=="fruit"]["price_per_lb"], #Make histogram
                    bins=20, color='red',alpha=.5, label="Fruit")  #for fruit
        plt.hist(food3[food3["type"]=="vegetables"]["price_per_lb"], #Make histogram
                    bins=20, color='blue',alpha=.7, label="Vegetables") #for vegetables
        plt.xlabel('Price per lb', fontsize=16) #creates xlabel
        plt.ylabel('Counts', fontsize=18) #creates ylabel
        plt.title('Foods vs Price per lb Distribution', fontsize=18) #creates title
        plt.legend(fontsize=15) #Create legend
        plt.rcParams['figure.figsize'] = [10, 5] #changes graph size

        plt.show() #prints graph
```



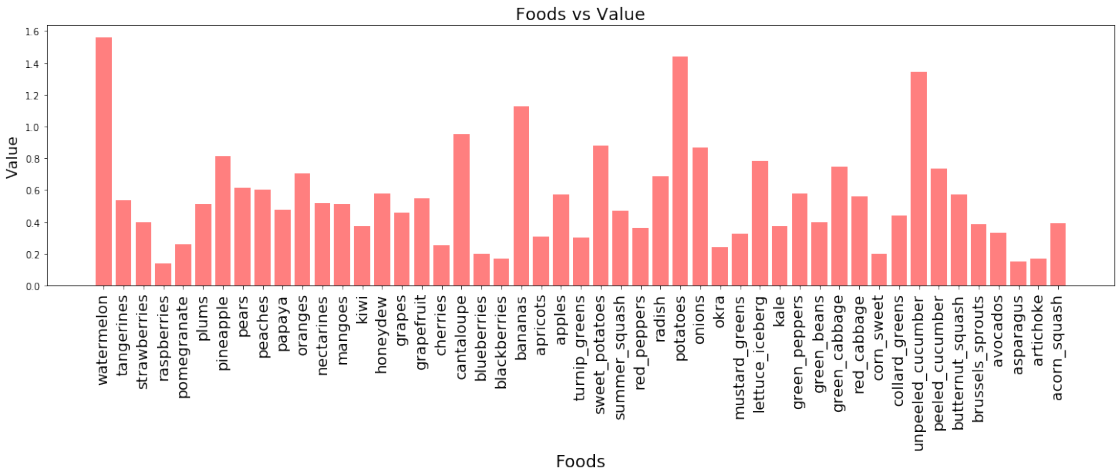
Most Exepensive Fruits:
Raspberries 6.97 per pound, Blackberries 5.77 per pound Blueberries 4.73 per pound
Least Expensive Fruits:
Watermelon .33 per pound, Cantaloupe .53 per pound, Bananas .56 per pound
Most Expensive Vegetables:
Okra 3.21 per pound, asparagus 3.21 per pound, Brussel Sprouts 2.76 per pound
Least Expensive Vegetables:
Potatoes .56 per pound, Sweet potatoes .91 per pound, Onions 1.03 per pound

The vegatables with higher costs are not as expensive as the fruits with higher costs. Okra has a high cost due to the production during warmer seasons. Therefore, during winter seasons, the US imports okra from outside thus the cost increase. Potatoes are the cheapest, since they require very little work to produce. The mass of potatoes are also varies between medium and large potatoes.

The fruit distribution (red) shows a skewed-right distribution, which might have a few outliers. The vegetable distribution (blue) shows a bimodel, and does not have any extreme outliers.

```
In [10]: food3["value"]=food3["yield"]/food3["price_per_lb"] #Create new variable for value
                                                            #using yield and price_per_lb
         x_pos = np.arange(len(food3)) #creates tick locations for all the foods
         plt.bar(x_pos, food3["value"],align='center',alpha=.5, color='red') #creates the graph
         plt.xticks(x_pos, food3["food"], rotation=90, fontsize=16) #writes the labels
         plt.xlabel('Foods', fontsize=18) #creates xlabel
         plt.ylabel('Value', fontsize=16) #creates ylabel
         plt.title('Foods vs Value', fontsize=18) #creates title
         plt.rcParams['figure.figsize'] = [20, 5] #changes graph size

         plt.show() #Prints graph
```



bananas have cheap cost at .56 and medium yield .64
Watermelon have cheap cost at .33 and medium yield .52
Potato have cheap cost at .56 and a high yield .81
Sweet potato have cheap cost at .91 and a high yield .81
The formula for value = yield / price_per_lb

The fruits and vegetables varies between the cost and yield. The fruits with low cost have rather medium yields. Rather the vegatables have low cost and high yield. The best combination between cost and yield would be low cost and high yield. Therefore, watermelon and okra are considered the best value for price against yield.

I find it interesting that many of the vegetables have medium to low yields but the price isn't low. Similarly, a lot of berries are extremely expensive compared to other fruits. The cheapest fruits have low yield, while the cheapest vegetables have the highest yield.