5_Pandas_Dictionaries_1_solution

September 19, 2019

1 Dictionaries

Create a dictionary called prices. Put the following key-value pairs of fruits and their prices in your dictionary:

banana - 0.79

apple - 1.99

orange - 1.50

pear - 0.97

[1]:
```
prices = {'banana':0.79, 'apple':1.99, 'orange':1.50, 'pear':0.97}
```

If you were to buy one of each fruit, how much would it cost? How about five of each fruit? Create a function to calculate the cost of n of each fruit

[2]:
```
def buy_fruit(prices, n):
    return sum(prices.values()) * n
```

Woah, avocados are on sale for $0.69 each! Add this key-value pair to prices.

Now, using your function from above, print the price of purchasing 12 of each fruit.

[3]:
```
prices['avocado'] = 0.69

buy_fruit(prices,12)
```

[3]:
```
71.28
```

2 Pandas

The Titanic dataset provides information about the survival of passengers onboard the Titanic, which sank after hitting an iceburg during its maiden voyage in April 1912. We will perform some basic analyses of these data using Pandas!

[5]:
```
# First, lets be sure pandas and numpy are both available for use:
import numpy as np
```

```
import pandas as pd

# Now, let's read in the data from the file titanic.csv.
titanic_data = pd.read_csv("titanic.csv")

# The .head() method allows us to see the first n rows of titanic_data. Let's␣
 ↪see 4 rows
# Is this object a series or a dataframe?
titanic_data.head(4)
```

[5]:

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	0	3	Mr. Owen Harris Braund	male	22.0	1	0	7.2500
1	1	1	Mrs. John Bradley (Florence Briggs Thayer) Cum…	female	38.0	1	0	71.2833
2	1	3	Miss. Laina Heikkinen	female	26.0	0	0	7.9250
3	1	1	Mrs. Jacques Heath (Lily May Peel) Futrelle	female	35.0	1	0	53.1000

```
# Check the dimensions of our data (i.e. how many rows and columns does it have?
 ↪)
titanic_data.shape
```

[6]:
```
(887, 8)
```

```
# What data do we have on each passenger anyway? lets check the column names to␣
 ↪find out
titanic_data.columns
```

[7]:
```
Index([u'Survived', u'Pclass', u'Name', u'Sex', u'Age',
       u'Siblings/Spouses Aboard', u'Parents/Children Aboard', u'Fare'],
      dtype='object')
```

2.0.1 Subsetting

```
# I want to see the first 5 values in the 'Survived' column using the column␣
 ↪name
# There are two ways to do this - can you figure out both? Double check they␣
 ↪give the same result
titanic_data['Survived'][0:5]
```

[9]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

[10]:
```
titanic_data.Survived[0:5]
```

[10]:
```
0    0
1    1
2    1
3    1
4    0
Name: Survived, dtype: int64
```

[11]:
```
# Now, I want to see the ages that correspond to the survival values in these␣
 ↪first 5 rows
# How can we select 5 rows and 2 columns? (use column names for accessing these␣
 ↪data)
titanic_data[['Survived','Age']][0:5]
```

[11]:
```
   Survived   Age
0         0  22.0
1         1  38.0
2         1  26.0
3         1  35.0
4         0  35.0
```

[12]:
```
# Time for some analysis. We want to know the total number and percentage of␣
 ↪surviors and non-survivors
# HINT: there is a method called .value_counts() that will help you here - use␣
 ↪the documentation!
titanic_data['Survived'].value_counts()
```

[12]:
```
0    545
1    342
Name: Survived, dtype: int64
```

[15]:
```
titanic_data['Survived'].value_counts(normalize=True)
```

[15]:
```
0    0.614431
1    0.385569
Name: Survived, dtype: float64
```

2.0.2 Cross-tabulation

[16]:
```
# Now we want to look into the distribution of males and females in both the survior and non-survivor groups
# HINT: how can the .crosstab() function help you here?
pd.crosstab( titanic_data.Sex, titanic_data.Survived )
```

[16]:
```
Survived    0    1
Sex
female     81  233
male      464  109
```

[19]:
```
# After you've found the totals, let's view these as percentages
# HINT: play with the normalize parameter to see how the results change.
# Which tells you the percentage of women that survived?

# Percent of women/men (rows add up to 1)
pd.crosstab( titanic_data.Sex, titanic_data.Survived, normalize = "index" )

# Percent of surviors/non-survivors (columns add up to 1)
pd.crosstab( titanic_data.Sex, titanic_data.Survived, normalize = "columns" )

# Percent of all people on board (all values add up to 1)
pd.crosstab( titanic_data.Sex, titanic_data.Survived, normalize = "all" )
```

[19]:
```
Survived         0         1
Sex
female    0.091319  0.262683
male      0.523112  0.122886
```

2.0.3 Filtering

[20]:
```
# We want to find out how many children under the age of 5 survived.
# Subset titanic_data to include only these data and save it to a new object, titanic_below_5
titanic_below_5 = titanic_data[ titanic_data.Age <= 5 ]
```

[21]:
```
# How many total children under 5 were onboard the titanic?
len(titanic_below_5)
```

[21]:
```
49
```

[22]:
```
# Now find the total numbers as well as percentages of survivors/non-survivors in this subset
titanic_below_5["Survived"].value_counts( )
```

[22]:
```
1    33
0    16
Name: Survived, dtype: int64
```

[23]:
```
titanic_below_5["Survived"].value_counts(normalize=True)
```

[23]:
```
1    0.673469
0    0.326531
Name: Survived, dtype: float64
```

[24]:
```
# BONUS: Find all the rows in titanic_data where the name contains "Allen"
titanic_data[titanic_data.Name.str.contains( "Allen") ]
```

[24]:

	Survived	Pclass	Name	Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
4	0	3	Mr. William Henry Allen	male	35.0	0	0	8.0500
726	1	1	Miss. Elisabeth Walton Allen	female	29.0	0	0	211.3375

[ ]: