CS4132 Data Analytics

Lab 3.2: Pandas

Submission Instructions

- Complete the following questions and upload your `.ipynb` file to Coursemology.
- Name the file in the following format: `Lab<num><YourName>.ipynb`
- Before submitting, please ensure you click on "Kernel" > "Restart and Run All" on your jupyter notebook.
- Finally, print a copy of your final solution to OneNote > Your Individual Student Notebook > Labs. Name the page `Lab <num>`.

You should harness the power of pandas operations in this lab.
You are NOT allowed to use `for` loop (including list comprehension) in all questions.

Q1

We will use the wine data as discussed in the notes for this lab.

In [135]:

```
import pandas as pd
reviews = pd.read_csv("wine.csv", index_col=0) #read in data from wine.csv
reviews.head() #display the first few data from the dataframe
```

Out[135]:

	country	description	designation	points	price	province	region_1	region_2	taster_na
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	K O'Ke
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger V
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gre
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexar Pearl
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gre

a) Select the first value from the description column of `reviews` and update it to `"The palate isn't overly expressive, offering unripened apple, citrus and dried sage alongside brisk acidity"`. Show that the data has been updated.

In [136]:

```
#your solution
reviews.loc[0, 'description'] = "The palate isn't overly expressive, offering unripened
reviews.head()
```

Out[136]:

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_tw
0	Italy	The palate isn't overly expressive, offering u...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	(
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	

b) Select the records with index labels `1`, `2`, `3`, `5`, and `8`. Update the points of all these records to 90. Show that the data has been updated.

In [137]:

```
#your solution
reviews.iloc[[1, 2, 3, 5, 8], 3] = 90
reviews.iloc[[1, 2, 3, 5, 8]]
```

Out[137]:

	country	description	designation	points	price	province	region_1	region_2	taste
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	90	15.0	Douro	NaN	NaN	Ro
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	90	14.0	Oregon	Willamette Valley	Willamette Valley	Paul
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	90	13.0	Michigan	Lake Michigan Shore	NaN	Al l
5	Spain	Blackberry and raspberry aromas show a typical...	Ars In Vitro	90	15.0	Northern Spain	Navarra	NaN	Sc
8	Germany	Savory dried thyme notes accent sunnier flavor...	Shine	90	12.0	Rheinhessen	NaN	NaN	Ann

c) For all wines where region_1 is a "Valley", increase the price by 5.

In [138]:

```
#your solution
reviews[(reviews.region_1.str.contains("Valley")) & (~reviews.region_1.isnull())]['price
reviews[(reviews.region_1.str.contains("Valley")) & (~reviews.region_1.isnull())].head()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_8184\3396367820.py:2: SettingWi
thCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  reviews[(reviews.region_1.str.contains("Valley")) & (~reviews.region_1.i
snull())]['price'] += 5
```

Out[138]:

	country	description	designation	points	price	province	region_1	region_2	taster_n
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	90	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gr
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gr
10	US	Soft, supple plum envelopes an oaky structure ...	Mountain Cuvée	87	19.0	California	Napa Valley	Napa	Vir B
12	US	Slightly reduced, this wine offers a chalky, t...	NaN	87	34.0	California	Alexander Valley	Sonoma	Vir B
33	US	Rustic and dry, this has flavors of berries, c...	Puma Springs Vineyard	86	50.0	California	Dry Creek Valley	Sonoma	

d) Split the taster name into first name and last name. Finally update taster name to be the first name only.

```
In [139]:
```

```
#your solution
reviews['taster_name'] = reviews['taster_name'].str.split(' ').str[0]
reviews.head()
```

```
Out[139]:
```

	country	description	designation	points	price	province	region_1	region_2	taster_na
0	Italy	The palate isn't overly expressive, offering u...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	K
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	90	15.0	Douro	NaN	NaN	Rc
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	90	14.0	Oregon	Willamette Valley	Willamette Valley	F
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	90	13.0	Michigan	Lake Michigan Shore	NaN	Alexar
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	F

e) Find all wines tasted by taster whose first name is 5 characters long.

In [140]:

```
#your solution
reviews[(reviews['taster_name'].str.len() == 5)].head()
```

Out[140]:

	country	description	designation	points	price	province	region_1	region_2	taster_nam
0	Italy	The palate isn't overly expressive, offering u...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Keri
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	90	15.0	Douro	NaN	NaN	Roge
6	Italy	Here's a bright, informal red that opens with ...	Belsito	87	16.0	Sicily & Sardinia	Vittoria	NaN	Keri
7	France	This dry and restrained wine offers spice in p...	NaN	87	24.0	Alsace	Alsace	NaN	Roge
9	France	This has great depth of flavor with its fresh ...	Les Natures	87	27.0	Alsace	Alsace	NaN	Roge

Q2

Complete the following questions using the sample DataFrame given:

In [141]:

```
data = pd.DataFrame({"col1": range(3),"col2": range(3,6)})
data
```

Out[141]:

	col1	col2
0	0	3
1	1	4
2	2	5

a) Insert a third column, named col3, with entries 6, 9, 10

In [142]:

```
#your solution
data.insert(2, "col3", [6, 9, 10])
data
```

Out[142]:

	col1	col2	col3
0	0	3	6
1	1	4	9
2	2	5	10

b) Insert a forth row with entries 5.

In [143]:

```
#your solution
data.loc[3] = 5
data
```

Out[143]:

	col1	col2	col3
0	0	3	6
1	1	4	9
2	2	5	10
3	5	5	5

c) Swap the data in col2 and col1.

In [144]:

```
#your solution
data['col1'], data['col2'] = data['col2'], data['col1']
data
```

Out[144]:

	col1	col2	col3
0	3	0	6
1	4	1	9
2	5	2	10
3	5	5	5

d) Drop the first two rows. This update should be reflected in `data` .

In [145]:

```
#your solution
data.drop([0, 1], axis = 0, inplace = True)
data
```

Out[145]:

	col1	col2	col3
2	5	2	10
3	5	5	5

e) Divide the elements in first row by the elements in the second row. Insert the result from the division as a new row in the dataframe as the first row.

In [146]:

```
#your solution
temp = (data.iloc[0] / data.iloc[1]).to_frame().T
pd.concat((temp, data), axis = 0)
```

Out[146]:

	col1	col2	col3
0	1.0	0.4	2.0
2	5.0	2.0	10.0
3	5.0	5.0	5.0

Q3

There is no inbuilt insert function in pandas to insert a row of data.

Write your own `insertRow()` function to allow a user to insert a row of data (in the form of a list) into a pandas dataframe.

You are NOT allowed to use the 2 methods described in the notes, and not allowed to use any for loops in this question.

In [182]:

```
def insertRow(df, index, data):
    temp1 = df[:index]
    temp2 = df[index:]
    return pd.concat((temp1, data, temp2), axis = 0)
```

In [185]:

```
#Test your code here
data = insertRow(data, 0, temp)
insertRow(data, 1, temp)
```

Out[185]:

	col1	col2	col3
0	1.0	0.4	2.0
0	1.0	0.4	2.0
2	5.0	2.0	10.0
3	5.0	5.0	5.0

Q4

Use the data below to answer the questions.

In [149]:

```
import pandas as pd
reviews = pd.read_csv("wine.csv", index_col=0) #read in data from wine.csv
reviews.head(5) #display the first few data from the dataframe
```

Out[149]:

	country	description	designation	points	price	province	region_1	region_2	taster_na
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	K O’Ke
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger V
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gre
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexan Pear
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gre

a) What is the median of the `points` column in the `reviews` DataFrame?

In [188]:

```
#your solution
reviews.points.median()
```

Out[188]:

```
86.0
```

b) What countries are represented in the dataset? (Your answer should not include any duplicates.)

In [189]:

```
#your solution
reviews.country.unique()
```

Out[189]:

```
array(['Italy', 'Portugal', 'US', 'Spain', 'France', 'Germany',
       'Argentina', 'Chile', 'Australia', 'Austria'], dtype=object)
```

c) How often does each country appear in the dataset? Create a Series `reviews_per_country` mapping countries to the count of reviews of wines from that country.

In [193]:

```
#your solution
reviews_per_country = reviews.country.value_counts()
reviews_per_country
```

Out[193]:

```
US           43
Italy        24
France       14
Chile         5
Germany       4
Spain         3
Portugal      2
Argentina     2
Australia     2
Austria       1
Name: country, dtype: int64
```

d) Which wine is the "best bargain"? Create a variable `bargain_wine` with the title of the wine with the highest points-to-price ratio in the dataset.

In [233]:

```
#your solution

temp = (reviews.points / reviews.price)
frame = pd.concat((reviews, temp.rename('ppp')), axis = 1)
bargain_wine = frame[frame.ppp == frame.ppp.max()].iloc[0]['title']
bargain_wine
```

Out[233]:

```
'Henry Fessy 2012 Nouveau  (Beaujolais)'
```

e) Is a wine more likely to be "tropical" or "fruity"? Create a Series `descriptor_counts` counting how many times each of these two words appears in the description column in the dataset.

In [242]:

```
#your solution
trop = len(reviews[reviews['description'].str.contains('tropical')])
fruit = len(reviews[reviews['description'].str.contains('fruity')])
descriptor_counts = pd.Series({'tropical': trop, 'fruity': fruit})
descriptor_counts
```

Out[242]:

```
tropical    4
fruity      8
dtype: int64
```

f) We'd like to host these wine reviews on our website, but a rating system ranging from 80 to 100 points is too hard to understand - we'd like to translate them into simple star ratings. A score of 95 or higher counts as 3 stars, a score of at least 85 but less than 95 is 2 stars. Any other score is 1 star.

Also, the Canadian Vintners Association bought a lot of ads on the site, so any wines from Canada should automatically get 3 stars, regardless of points.

Create a series star_ratings with the number of stars corresponding to each review in the dataset.

In [267]:

```
#your solution
import numpy as np
temp = np.where((reviews.points >= 85) & (reviews.points < 95), 2, np.where((reviews.cou
pd.Series(temp)
```

Out[267]:

```
0      2
1      2
2      2
3      2
4      2
      ..
95     2
96     2
97     2
98     2
99     2
Length: 100, dtype: int32
```

g) For each taster, count how many reviews each person wrote.

In [269]:

```
#your solution
reviews.taster_name.value_counts()
```

Out[269]:

```
Roger Voss           16
Virginie Boone       16
Kerin O’Keefe        13
Michael Schachner    10
Paul Gregutt          6
Sean P. Sullivan      6
Anna Lee C. Iijima    5
Alexander Peartree    3
Matt Kettmann         3
Joe Czerwinski        2
Jim Gordon            1
Anne Krebiehl MW      1
Name: taster_name, dtype: int64
```

Hence, find which revewier wrote the most review in this dataset.

In [278]:

```
#your solution
temp = reviews.taster_name.value_counts()
temp[temp == temp.max()]

reviews.taster_name.mode()
```

Out[278]:

```
0        Roger Voss
1    Virginie Boone
Name: taster_name, dtype: object
```

h) What are the minimum and maximum prices for each variety of wine?

In [295]:

```
#your solution
reviews.groupby('variety').price.agg(['min', 'max']).head()
```

Out[295]:

	min	**max**
variety		
Aglianico	32.0	32.0
Albariño	16.0	20.0
Bordeaux-style Red Blend	46.0	75.0
Bordeaux-style White Blend	15.0	15.0
Cabernet Franc	25.0	25.0

i) What are the most expensive wine varieties? Create a variable `sorted_varieties` containing a copy of the dataframe from the previous question where varieties are sorted in descending order based on minimum price, then on maximum price (to break ties).

In [302]:

```
#your solution
sorted_varieties = reviews.groupby('variety').price.agg(['min', 'max']).sort_values(by =
sorted_varieties.head()
```

Out[302]:

variety	min	max
Champagne Blend	55.0	58.0
Petite Sirah	55.0	55.0
Bordeaux-style Red Blend	46.0	75.0
Meritage	32.0	55.0
Aglianico	32.0	32.0

j) For each taster, find the average review score given out by that reviewer.

In [305]:

```
#your solution
reviews.groupby('taster_name').points.agg('mean')
```

Out[305]:

```
taster_name
Alexander Peartree    87.000000
Anna Lee C. Iijima    86.800000
Anne Krebiehl MW      88.000000
Jim Gordon            86.000000
Joe Czerwinski        86.000000
Kerin O’Keefe         86.923077
Matt Kettmann         86.666667
Michael Schachner     86.200000
Paul Gregutt          86.500000
Roger Voss            86.437500
Sean P. Sullivan      86.333333
Virginie Boone        86.625000
Name: points, dtype: float64
```

k) Count the number of reviews available for each combination of countries and varieties.

For example, a Gamay produced in France has 5 reviews. Sort the values in descending order based on wine count.

In [361]:

```
#your solution
test = reviews.groupby(['country', 'variety']).count().sort_values(by = ['title'],ascend
test.head()
```

Out[361]:

		title
country	**variety**	
US	**Pinot Noir**	6
	Red Blend	5
	Cabernet Sauvignon	5
France	**Gamay**	5
Italy	**Red Blend**	4

Hence, which combination of countries and varieties is most common?

In [362]:

```
#your solution
test.idxmax()
```

Out[362]:

```
title    (US, Pinot Noir)
dtype: object
```

l) Find the average price for each combination of countries and varieties.

In [383]:

```
#your solution
temp = reviews.groupby(['country', 'variety']).price.mean().to_frame()
temp.head()
```

Out[383]:

		price
country	**variety**	
Argentina	**Malbec**	21.5
Australia	**Chardonnay**	18.0
	Rosé	20.0
Austria	**Grüner Veltliner**	12.0
Chile	**Carmenère**	12.0

Hence for each country, find the variety with the highest average price.

In [384]:

```
#your solution
temp.groupby('country').idxmax()
```

Out[384]:

	price
country	
Argentina	(Argentina, Malbec)
Australia	(Australia, Rosé)
Austria	(Austria, Grüner Veltliner)
Chile	(Chile, Petit Verdot)
France	(France, Champagne Blend)
Germany	(Germany, Riesling)
Italy	(Italy, Aglianico)
Portugal	(Portugal, Portuguese Red)
Spain	(Spain, Tempranillo Blend)
US	(US, Bordeaux-style Red Blend)

Q5

Use the data below to answer the questions.

In [385]:

```
import pandas as pd;
DFAmount = pd.DataFrame(data = [[4,19,13,9],[7,13,20,12],[11,16,13,5],[4,3,4,5]], column
DFAmount
```

Out[385]:

	XS	S	M	L
Shirt	4	19	13	9
Tee Shirt	7	13	20	12
Polo Shirt	11	16	13	5
V Neck Shirt	4	3	4	5

In [386]:

```
DFSellingprice = pd.DataFrame(data = [[21.9,22.9,23.9,24.9],[23.9,23.9,25.9,25.9],[29,29
DFSellingprice
```

Out[386]:

	XS	S	M	L
Shirt	21.9	22.9	23.9	24.9
Tee Shirt	23.9	23.9	25.9	25.9
Polo Shirt	29.0	29.0	32.0	32.0
V Neck Shirt	28.0	28.0	28.0	28.0

In [387]:

```
DFCostprice = pd.DataFrame(data = [[18.0,18,19,19],[20,20,21,21],[25,25,25,25],[20,20,22
DFCostprice
```

Out[387]:

	XS	S	M	L
Shirt	18.0	18	19	19
Tee Shirt	20.0	20	21	21
Polo Shirt	25.0	25	25	25
V Neck Shirt	20.0	20	22	22

a) Find the total cost price of the all the item in the shop.

In [394]:

```
#your solution
cost = (DFAmount * DFCostprice).sum().sum()
cost
```

Out[394]:

```
3367.0
```

b) Find the total profit if he sold all the item.

In [395]:

```
#your solution
sell = DFAmount * DFSellingprice
sell.sum().sum() - cost
```

Out[395]:

```
804.2999999999993
```

c) He realized that he had entered the selling price of the shirt wrongly. The selling price of the shirt in increasing sizes is 22.90, 24.90, 25.90 and 26.90 respectively. Make the necessary amendment to the DFSellingprice.

In [398]:

```
#your solution
DFSellingprice.loc['Shirt'] = [22.90, 24.90, 25.90, 26.90]
DFSellingprice
```

Out[398]:

	XS	S	M	L
Shirt	22.9	24.9	25.9	26.9
Tee Shirt	23.9	23.9	25.9	25.9
Polo Shirt	29.0	29.0	32.0	32.0
V Neck Shirt	28.0	28.0	28.0	28.0