DSC 80 Discussion 2 Worksheet

1 WI24 Midterm Problem 3

Jasmine is a veterinarian. Below, you'll find information about some of the dogs in her care, separated by district and breed.

	Beagle		Cocker Spaniel	
	Mean Weight	Count	Mean Weight	Count
District 1	25	3	20	2
District 2	45	1	x	y

What is the mean weight of all beagles in the table above, across both districts?

Notice that the table above has two unknowns, x and y. Find **positive integers** x and y such that the mean weight of all beagles is equal to the mean weight of all cocker spaniels, *where x is as small as possible.*

2 FA23 Midterm Problem 1

	date	name	food	weight
0	2023-01-01	Sam	Ribeye	0.20
1	2023-01-01	Sam	Pinto beans	0.10
2	2023-01-01	Lauren	Mung beans	0.25
3	2023-01-02	Lauren	Lima beans	0.30
4	2023-01-02	Sam	Sirloin	0.30

Find the total kg of food eaten for each day and each person in `df` as a Series.

```
df.groupby(__________________________________________)[________________________].sum()
```

Find all the unique people who did not eat any food containing the word "beans".

```
def foo(x):
    return __________________________________


df.groupby(______________________________).________________________(foo)['name'].unique()
```

3 FA23 Final Problem 1

The bus table (left) records bus arrivals over 11 day for all the bus stops within a 22 mile radius of UCSD.

	time	line	stop	late
0	12pm	201	Gilman Dr & Mandeville Ln	-1.1
1	1:15pm	30	Gilman Dr & Mandeville Ln	2.8
2	11:02am	101	Gilman Dr & Myers Dr	-0.8
3	8:04am	202	Gilman Dr & Myers Dr	NaN
4	9am	30	Gilman Dr & Myers Dr	-3.0

`time` Time of arrival (`str`). Note that the times are inconsistently entered (e.g. `12pm` vs. `1:15pm`).
`line` Bus line (`int`). There are multiple buses per bus line each day.
`stop` Bus stop (`str`).
`late` The number of minutes the bus arrived after its scheduled time. Negative numbers mean that the bus arrived early (`float`). Some entries in this column are missing.

The stop table (left) contains information for all the bus lines in San Diego (not just the ones near UCSD).

	line	stop	next
0	201	Gilman Dr & Mandeville Ln	VA Hospital
1	201	VA Hospital	La Jolla Village Dr & Lebon Dr
2	30	VA Hospital	Villa La Jolla Dr & Holiday Ct
3	30	UTC	NaN

`line` Bus line (`int`).
`stop` Bus stop (`str`).
`next` The next bus stop for a particular bus line (`str`). For example, the first row of the table shows that after the `201` stops at `Gilman Dr & Mandeville Ln`, it will stop at the `VA Hospital` next. A missing value represents the end of a line.

Compute the number of buses in `bus` whose next stop is `'UTC'`.

```
x = stop.merge(___________, on = ____________________________, how = ________________)

x[_____________________________________________].shape[0]
```

Compute the number of unique pairs of bus stops that are exactly two stops away from each other. For example, if you only use the first four rows of the `stop` table, then your code should evaluate to the number 2, since you can go from `'Gilman Dr & Mandeville Ln'` to `'La Jolla Village Dr & Lebon Dr'` and from `'Gilman Dr & Mandeville Ln'` to `'Villa La Jolla Dr & Holiday Ct'` in two stops. Hint: The `suffixes = (1, 2)` argument to merge appends a `1` to column labels in the left table and a `2` to column labels in the right table whenever the merged tables share column labels.

```
m = ___________.merge(___________, left_on = ___________, right_on = ___________,
      how = ____________, suffixes=(1, 2))
(m[____________________________].drop_duplicates().shape[0])
```

4 FA22 Midterm Problem 7

	category	completed	minutes	urgency	client
0	work	False	NaN	2.0	NaN
1	work	False	NaN	1.0	NaN
2	work	True	13.5	2.0	NaN
3	work	False	NaN	1.0	NaN
4	relationship	True	5.3	NaN	NaN
...	...	...	...	...	...
9831	consulting	True	71.7	2.0	San Diego Financial Analysts
9832	finance	True	36.4	1.0	NaN
9833	work	True	31.1	1.0	NaN
9834	work	True	24.8	3.0	NaN
9835	work	False	NaN	2.0	NaN

The code below creates a pivot table.

```
pt = tasks.pivot_table(index='urgency', columns='category', values='completed', aggfunc='sum')
```

Which of the below snippets of code will produce the same result as `pt.loc[3.0, 'consulting']`? **Select all that apply.**

☐ Snippet 1:

```
tasks[(tasks['category'] == 'consulting') & (tasks['urgency'] == 3.0)]['completed'].sum()
```

☐ Snippet 2:

```
tasks[tasks['urgency'] == 3].groupby('category')['completed'].sum().loc['consulting']
```

☐ Snippet 3:

```
tasks.groupby('urgency')['completed'].sum().loc[3.0, 'consulting']
```

☐ Snippet 4:

```
tasks.groupby(['urgency', 'category'])['completed'].sum().loc[(3.0, 'consulting')]
```

☐ Snippet 5

```
tasks.groupby('completed').sum().loc[(3.0, 'consulting')]
```