

Tutorial Exercises Week 4 - Solutions

* Question 1

Copy the following extended dataset from the course document into Python.

```
dataset = {
    'name' : ["Aiden", "Bella", "Carlos", "Dalia", "Elena", "Farhan", "Geert", "Hadi", "Ian", "Jane"],
    'height (cm)' : [185, 155, 190, 185, 160, 170, 165, 178, 192, 158],
    'weight (kg)' : [80, 60, 100, 85, 62, 75, 75, 85, 90, 70],
    'age (years)' : [23, 23, 23, 21, 19, 25, 26, 19, 18, 22],
    'dietary preference' : ['Veggie', 'Veggie', 'None', 'None', 'Vegan', 'None', 'Veggie', 'None', 'Vegan', 'None']
}
```

a) Turn the dataset above into a Pandas data frame and print it.

```
import pandas as pd

frame = pd.DataFrame(dataset)

print(frame)
```

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
1	Bella	155	60	23	Veggie
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
4	Elena	160	62	19	Vegan
5	Farhan	170	75	25	None
6	Geert	165	75	26	Veggie
7	Hadi	178	85	19	None
8	Ian	192	90	18	Vegan
9	Jane	158	70	22	None

b) Print all the odd numbered rows from the data frame:

	name	height (cm)	weight (kg)	age (years)	dietary preference
1	Bella	155	60	23	Veggie
3	Dalia	185	85	21	None

5	Farhan	170	75	25	None
7	Hadi	178	85	19	None
9	Jane	158	70	22	None

```
# Store the desired rows in a new Pandas data frame x
x = frame.loc[[1,3,5,7,9]]
```

```
print(x)
```

	name	height (cm)	weight (kg)	age (years)	dietary preference
1	Bella	155	60	23	Veggie
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
7	Hadi	178	85	19	None
9	Jane	158	70	22	None

c) Print the following part of the frame:

	name	dietary preference
1	Bella	Veggie
2	Carlos	None
4	Elena	Vegan
6	Geert	Veggie
7	Hadi	None

```
x = frame.loc[[1,2,4,6,7],['name','dietary preference']]
```

```
print(x)
```

	name	dietary preference
1	Bella	Veggie
2	Carlos	None
4	Elena	Vegan
6	Geert	Veggie
7	Hadi	None

d) Print the rows of the people whose weight is greater or equal than 75 kilograms:

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
6	Geert	165	75	26	Veggie
7	Hadi	178	85	19	None
8	Ian	192	90	18	Vegan

```

threshold = 75
weight_75 = frame.loc[:, 'weight (kg)'] >= threshold

x = frame.loc[weight_75, :]

print(x)

```

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
6	Geert	165	75	26	Veggie
7	Hadi	178	85	19	None
8	Ian	192	90	18	Vegan

e) Print the rows of the people whose height is in the interval [170, 190]:

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
7	Hadi	178	85	19	None

Hint: Write a for-loop that checks for every row whether the height of a person is in the desired interval, and, if so, append the row index to an (initially empty) list.

```

a = 170
b = 190

interval = []
for i in frame.index:
    if frame.loc[i, 'height (cm)'] >= a and frame.loc[i, 'height (cm)'] <= b:
        interval.append(i)

x = frame.loc[interval, :]

print(x)

```

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None

7	Hadi	178	85	19	None
---	------	-----	----	----	------

- f) Use a for-loop to print three frames based on the dietary preferences that the people have. So one frame containing all the rows of people whose preference is Veggie, one with people whose preference is Vegan, and one with people whose preference is None:

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
1	Bella	155	60	23	Veggie
6	Geert	165	75	26	Veggie
	name	height (cm)	weight (kg)	age (years)	dietary preference
4	Elena	160	62	19	Vegan
8	Ian	192	90	18	Vegan
	name	height (cm)	weight (kg)	age (years)	dietary preference
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
7	Hadi	178	85	19	None
9	Jane	158	70	22	None

```
preferences = ['Veggie', 'Vegan', 'None']
```

```
# Alternatively: preferences = frame.loc[:, 'dietary preference'].unique()
# This would be quicker if there are many different preferences.
```

```
for i in preferences:
    pref = frame.loc[:, 'dietary preference'] == i
    print(frame.loc[pref, :])
```

	name	height (cm)	weight (kg)	age (years)	dietary preference
0	Aiden	185	80	23	Veggie
1	Bella	155	60	23	Veggie
6	Geert	165	75	26	Veggie
	name	height (cm)	weight (kg)	age (years)	dietary preference
4	Elena	160	62	19	Vegan
8	Ian	192	90	18	Vegan
	name	height (cm)	weight (kg)	age (years)	dietary preference
2	Carlos	190	100	23	None
3	Dalia	185	85	21	None
5	Farhan	170	75	25	None
7	Hadi	178	85	19	None
9	Jane	158	70	22	None

* Question 2

In this exercise, we will replace the metrics that are used to compute the height

and weight.

- a) Adjust the weight column so that the weights are displayed in pounds (lbs) instead of kilograms (kg). For this, you may use that $1 \text{ kg} = 2.2205 \text{ lbs}$. The column name should be changed to 'weight (lbs)'. The frame should look like this in the end:

	name	height (cm)	weight (lbs)	age (years)	dietary preference
0	Aiden	185	177.6400	23	Veggie
1	Bella	155	133.2300	23	Veggie
2	Carlos	190	222.0500	23	None
3	Dalia	185	188.7425	21	None
4	Elena	160	137.6710	19	Vegan
5	Farhan	170	166.5375	25	None
6	Geert	165	166.5375	26	Veggie
7	Hadi	178	188.7425	19	None
8	Ian	192	199.8450	18	Vegan
9	Jane	158	155.4350	22	None

Hint: Define a function that does the conversion from kilograms to pounds and apply it to the weight column. To rename the column, rename all columns using `frame.columns`.

```
import pandas as pd

def kg_to_lbs(y):
    return 2.2205*y

frame['weight (kg)'] = frame['weight (kg)'].apply(kg_to_lbs)

# We rename the entire set of columns, because it is not possible to
# change one column name at a time. There is a function rename() that
# can do this, but we omit it here.
frame.columns = ['name', 'height (cm)', 'weight (lbs)', 'age (years)',
                 'dietary preference']

print(frame)
```

	name	height (cm)	weight (lbs)	age (years)	dietary preference
0	Aiden	185	177.6400	23	Veggie
1	Bella	155	133.2300	23	Veggie
2	Carlos	190	222.0500	23	None
3	Dalia	185	188.7425	21	None
4	Elena	160	137.6710	19	Vegan
5	Farhan	170	166.5375	25	None
6	Geert	165	166.5375	26	Veggie
7	Hadi	178	188.7425	19	None

8	Ian	192	199.8450	18	Vegan
9	Jane	158	155.4350	22	None

A common height measure used in the US is that of feet and inches. If a person is $f'i''$ it means they are f feet plus i inches long. Here 1 foot = 30.48 cm and 1 inch = 2.54 cm. For example, if you are 6'4" tall, you are $6 \cdot 30.48 + 4 \cdot 2.54 \approx 193$ cm tall.

Conversely, if you are 193 cm tall, then the $f'i''$ height can be computed as follows. First divide $193/30.48 \approx 6.33$, meaning that 6 whole feet fit in 193 cm. Then the remainder $193 - 6 \cdot 30.48 = 10.12$ can be divided by 2.54 to obtain the number of inches, namely $10.12/2.54 = 4$.

- b) Adjust the heights in the frame so that they are displayed as $[f, i]$, meaning a height of $f'i''$. Adjust the column name to `'height (f'i'')`. You may round the number of inches to an integer, which you can do with `round()`.

Hint: The same hint as in part a) applies. Also recall that you can use `//` to do modulo operations. For example `14 // 3` gives 4 because 3 fits 4 times in 14.

The new frame should look like this:

	name	height (f'i'')	weight (lbs)	age (years)	dietary preference
0	Aiden	[6.0, 1]	177.6400	23	Veggie
1	Bella	[5.0, 1]	133.2300	23	Veggie
2	Carlos	[6.0, 3]	222.0500	23	None
3	Dalia	[6.0, 1]	188.7425	21	None
4	Elena	[5.0, 3]	137.6710	19	Vegan
5	Farhan	[5.0, 7]	166.5375	25	None
6	Geert	[5.0, 5]	166.5375	26	Veggie
7	Hadi	[5.0, 10]	188.7425	19	None
8	Ian	[6.0, 4]	199.8450	18	Vegan
9	Jane	[5.0, 2]	155.4350	22	None

```
def cm_to_fi(y):
    foot = y // 30.48
    inch = (y - foot*30.48)/2.54
    inch = round(inch)
    return [foot, inch]

frame['height (cm)'] = frame['height (cm)'].apply(cm_to_fi)

frame.columns = ['name', "'height (f'i'')'", 'weight (lbs)', 'age (years)', 'dietary pr

print(frame)
```

	name	height (f'i'')	weight (lbs)	age (years)	dietary preference
0	Aiden	[6.0, 1]	177.6400	23	Veggie

1	Bella	[5.0, 1]	133.2300	23	Veggie
2	Carlos	[6.0, 3]	222.0500	23	None
3	Dalia	[6.0, 1]	188.7425	21	None
4	Elena	[5.0, 3]	137.6710	19	Vegan
5	Farhan	[5.0, 7]	166.5375	25	None
6	Geert	[5.0, 5]	166.5375	26	Veggie
7	Hadi	[5.0, 10]	188.7425	19	None
8	Ian	[6.0, 4]	199.8450	18	Vegan
9	Jane	[5.0, 2]	155.4350	22	None

* Question 3

The Body Mass Index (BMI) is a well-known, and sometimes critized, measure used to quantify how healthy a person is. It is a function of the height (m) and weight (kg) of a person, given by the formula

$$\text{BMI}(\text{height}, \text{weight}) = \frac{\text{weight}}{\text{height}^2}$$

- a) Start with the frame in Question 1a (with height in centimeters and weight in kilograms) and add a column right of the weight column that contains the BMI of every person.

Hint: Define a function that computes the BMI for a given weight and height, and then use a for-loop to compute the BMI values.

The new frame should look like this.

	name	height (cm)	weight (kg)	BMI	age (years)	dietary preference
0	Aiden	185	80	23.374726	23	Veggie
1	Bella	155	60	24.973985	23	Veggie
2	Carlos	190	100	27.700831	23	None
3	Dalia	185	85	24.835646	21	None
4	Elena	160	62	24.218750	19	Vegan
5	Farhan	170	75	25.951557	25	None
6	Geert	165	75	27.548209	26	Veggie
7	Hadi	178	85	26.827421	19	None
8	Ian	192	90	24.414062	18	Vegan
9	Jane	158	70	28.040378	22	None

```
import pandas as pd
```

```
dataset = {
```

```
    'name' : ["Aiden", "Bella", "Carlos", "Dalia", "Elena", "Farhan", "Geert", "Hadi", "Ian"],
    'height (cm)' : [185, 155, 190, 185, 160, 170, 165, 178, 192, 158],
    'weight (kg)' : [80, 60, 100, 85, 62, 75, 75, 85, 90, 70],
    'age (years)' : [23, 23, 23, 21, 19, 25, 26, 19, 18, 22],
```

```

    'dietary preference' : ['Veggie','Veggie','None','None','Vegan','None', 'Veggie', 'None']
}

frame = pd.DataFrame(dataset)

def BMI(weight, height):
    return weight/(height/100)**2

bmi = []
for i in frame.index:
    x = BMI(frame.loc[i,'weight (kg)'],frame.loc[i,'height (cm)'])
    bmi.append(x)

frame.insert(3,'BMI',bmi)

print(frame)

```

	name	height (cm)	weight (kg)	BMI	age (years)	dietary preference
0	Aiden	185	80	23.374726	23	Veggie
1	Bella	155	60	24.973985	23	Veggie
2	Carlos	190	100	27.700831	23	None
3	Dalia	185	85	24.835646	21	None
4	Elena	160	62	24.218750	19	Vegan
5	Farhan	170	75	25.951557	25	None
6	Geert	165	75	27.548209	26	Veggie
7	Hadi	178	85	26.827421	19	None
8	Ian	192	90	24.414062	18	Vegan
9	Jane	158	70	28.040378	22	None

b) Compute the average BMI value of the people who are 22 or younger.

25.66725157640014

```

below_22 = frame.loc[:,'age (years)'] <= 22

frame_22 = frame.loc[below_22,:]

x = frame_22.loc[:,'BMI'].mean()

print(x)

```

25.66725157640014

* Question 4

In this exercise we will merge two comma-separated value files together. We

start with two comma-separated values files that can be downloaded here: Week5_Q4_Part1.csv and Week5_Q4_Part2.csv. Both files contain a list of 100 names.

- a) Load both files into a (separate) Pandas data frame and print the first 10 rows from both. The output should look like this:

```
      0
0    Amina
1  Hiroshi
2  Isabella
3  Alejandro
4    Fatima
5    Johan
6    Priya
7    Carlos
8  Chiamaka
9     Mei
```

```
      0
0    Maria
1    Arun
2    Zhang
3  Esteban
4    Yasmin
5  Mustafa
6  Natalia
7    Omar
8    Aisha
9    Lucia
```

```
import pandas as pd

data_one = pd.read_csv('Week5_Q4_Part1.csv', header=None)
frame_one = pd.DataFrame(data_one)

data_two = pd.read_csv('Week5_Q4_Part2.csv', header=None)
frame_two = pd.DataFrame(data_two)

print(data_one.head(10))
print(data_two.head(10))
```

```
      0
0    Amina
1  Hiroshi
2  Isabella
3  Alejandro
```

```

4      Fatima
5      Johan
6      Priya
7      Carlos
8      Chiamaka
9      Mei
0
0      Maria
1      Arun
2      Zhang
3      Esteban
4      Yasmin
5      Mustafa
6      Natalia
7      Omar
8      Aisha
9      Lucia

```

We want to pair up the persons in the different sheets with each other so that they can form a team for a group project. That is, we want to pair up Amina with Maria, Hiroshi with Arun, Isabella with Zhang, Alejandro with Esteban, Fatima with Yasmin etc.

- b) Create a new data frame that contains three columns: One with the team number (ranging from 1 to 100), one with the first team member from the first dataset, and one with the second team member from the second dataset. The first five rows of the new frame should look like this:

	Team number	Member 1	Member 2
0	1	Amina	Maria
1	2	Hiroshi	Arun
2	3	Isabella	Zhang
3	4	Alejandro	Esteban
4	5	Fatima	Yasmin

Hint: Create a dictionary for the first team like this: `team1 = {'Team number' : [1], 'Member 1' : ['Amina'], 'Member 2' : ['Maria'] }` and turn it into a Pandas frame. Then use a for-loop to add the other teams to the frame.

```

import numpy as np

# Define number of teams
num_teams = 100

# Create dictionary with first team only
team1 = {'Team number' : [1], 'Member 1' : [frame_one.loc[0,0]], 'Member 2' : [frame_two.

```

```
# Create frame for first team
teams_frame = pd.DataFrame(team1)

for i in range(num_teams):
    if i > 0: # We already have the first team (with index 0) so can skip it.
        teams_frame.loc[i] = [i+1, frame_one.loc[i,0], frame_two.loc[i,0]]

print(teams_frame.head(5))
```

	Team number	Member 1	Member 2
0	1	Amina	Maria
1	2	Hiroshi	Arun
2	3	Isabella	Zhang
3	4	Alejandro	Esteban
4	5	Fatima	Yasmin

- c) Export the frame with teams to a new comma-separated values file called teams.csv, without the index numbers of the frame.

```
teams_frame.to_csv('teams.csv', index=False)
```

The folder in which you have stored your Python file should not contain a new file called teams.csv which should look like this.