Data Exercises II Python Pandas

Exercise #1: Assume we have the following dataset of cities.

	city	black	white	yellow	green	red	orange	purple
0	NY	2	13	2	14	7	12	7
1	SF	10	5	15	25	1	19	30
2	LA	77	23	12	34	45	56	65
3	DC	30	30	10	13	24	26	1

In Google Colab (or Jupyter notebook, etc), load the dataset with the following code:

```
import pandas as pd
df1 = pd.read_csv('http://www.ntu.edu.sg/home/fscheong/public/citycolors.csv')
```

Think of the columns (of colors) as representing various measures of crime rates.
Assume we want to know which crime is most likely to occur in each city.
In other words, in each row, find the column name(s) with the maximum numerical value.

For example, in the first row (NY city), 14 is the largest number, which corresponds to "green".
In the last row (for DC), 30 is the largest number, which corresponds to both "black" and " white".

The output from your Python program should be:

	city	maxcol
0	NY	green
1	SF	purple
2	LA	black
3	DC	black
4	DC	white

Notice that DC has two rows. In other words, the output can have more rows than the input.
Avoid hard-coding the names of variables / columns (i.e., your program should not assume that the "black", "green", etc exists)

If you need inspiration, below is my equivalent Stata code:

```
foreach var of varlist * {
  rename `var' t`var'
}
rename tcity city
reshape long t, i(city) j(maxcol) string
bysort city: egen maxcount = max(t)
drop if t<maxcount
drop t maxcount
```

Exercise #2: linear regression and data subsetting.

Here, we use the auto.dta dataset, and partition them into `rep78` (5 groups) and `foreign` (2 groups).
For all possible subsets (including all possible "subgroups"), run a **linear regression of mpg on weight**, and report the coefficient estimate of weight.

In Google Colab (or Jupyter notebook, etc), load the dataset using the following code:

```
import pandas as pd
df1 = pd.read_stata('http://www.stata-press.com/data/r11/auto.dta')
```

The output from your Python program should be:

	rep78	domestic	foreign	All
0	1	-0.008108	NaN	-0.008108
1	2	-0.008046	NaN	-0.008046
2	3	-0.005066	-0.015476	-0.004410
3	4	-0.005002	-0.005500	-0.004957
4	5	-0.012500	-0.019698	-0.018160
5	All	-0.005999	-0.010747	-0.006009

Exercise #3: reshaping data

In Google Colab (or Jupyter notebook, etc), load the dataset using the following code:

```
import pandas as pd
df1 = pd.read_stata('http://www.stata-press.com/data/r11/reshape1.dta')
df1
```

id	sex	inc80	inc81	inc82	ue80	ue81	ue82
1	0	5000	5500	6000	0	1	0
2	1	2000	2200	3300	1	0	0
3	0	3000	2000	1000	0	0	1

The output from your Python program should be:

id	year	sex	inc	ue
1	80	0	5000	0
1	81	0	5500	1
1	82	0	6000	0
2	80	1	2000	1
2	81	1	2200	0
2	82	1	3300	0
3	80	0	3000	0
3	81	0	2000	0
3	82	0	1000	1

Replicate the above using the following three different methods:
(a) `stack()` function in Python Pandas
(b) `melt()` function in Python Pandas
(c) nested `for` loops (to copy the numbers over to a new dataframe)