Missing Values Exercises - due thu 09/21 | *by Charlotte*

Exercise 1

To begin, load the ACS Data we used in our first pandas exercise. That data can be found here. We'll be working with US_ACS_2017_10pct_sample.dta.

In [108…
```
import pandas as pd
acs = pd.read_stata('US_ACS_2017_10pct_sample.dta')
```

Exercise 2

Let's begin by calculating the mean US incomes from this data (recall that income is stored in the inctot variable).

In [109…
```
f'Mean US incomes is ${acs["inctot"].mean().round(2)}'
```

Out[109…
```
'Mean US incomes is $1723646.27'
```

Exercise 3

Hmmm... That doesn't look right. The average American is definitely not earning 1.7 million dollars a year. Let's look at the values of inctot using value_counts(). Do you see a problem?

- Yes, most likely we have missing values in the dataset that are still coded as 9999999 or 8888888, instead of na's.

Now use value_counts() with the argument normalize=True to see proportions of the sample that report each value instead of the count of people in each category. What percentage of our sample has an income of 9,999,999? What percentage has an income of 0?

In [110…
```
acs['inctot'].value_counts(normalize=True)
```

Out[110…
```
9999999    0.168967
0          0.105575
30000      0.014978
50000      0.013837
40000      0.013834
             ...
70520      0.000003
76680      0.000003
57760      0.000003
200310     0.000003
505400     0.000003
Name: inctot, Length: 8471, dtype: float64
```

- 16.9% of the sample reportedly has an income of $9,999,999.
- 10.6% has an income of $0.

Exercise 4

To help out pandas, use the replace command to replace all values of 9999999 with np.nan.

In [111…
```
import numpy as np
acs['inctot'] = acs['inctot'].replace(9999999, np.nan)
acs['inctot']
```

Out[111…
```
0             NaN
1          6000.0
2          6150.0
3         14000.0
4             NaN
            ...
318999    22130.0
319000        NaN
319001     5000.0
319002   240000.0
319003    48000.0
Name: inctot, Length: 319004, dtype: float64
```

Exercise 5

Now that we've properly labeled our missing data as np.nan, let's calculate the average US income once more.

In [112…
```
f'Mean US incomes is ${acs["inctot"].mean().round(2)}'
```

Out[112…
```
'Mean US incomes is $40890.18'
```

Exercise 6

So let's make sure we understand why data is missing for some people. If you recall from our last exercise, it seemed to be the case that most of the people who had incomes of 9999999 were children. Let's make sure that's true by looking at the distribution of the variable age for people for whom inctot is missing (i.e. subset the data to people with inctot missing, then look at the values of age with value_counts()).

In [113…

```
acs_sub = acs.loc[acs['inctot'].isnull()]
acs_sub
acs_sub['age'].value_counts()
```

Out[113…

```
10    3997
9     3977
14    3847
12    3845
13    3800
      ...
39       0
38       0
37       0
36       0
96       0
Name: age, Length: 97, dtype: int64
```

It seems that those who income is missing for are children.

Then do the opposite: look at the distribution of the age variable for people for whom inctot is not missing.

In [114…

```
acs_sub2 = acs.loc[acs['inctot'].notnull()]
acs_sub2
acs_sub2['age'].value_counts()
```

Out[114…

```
60                      4950
54                      4821
59                      4776
56                      4776
58                      4734
                        ...
5                          0
4                          0
3                          0
2                          0
less than 1 year old       0
Name: age, Length: 97, dtype: int64
```

The opposite is true. Those where income is reported are adults. Which makes sense.

Can you determine when 9999999 was being used? Is it ok we're excluding those people from our analysis?

- 9999999 was used for children. It is okay if we exclude them from our analysis as we look at the income distribution and how it varies by race. Children by default and in the normal case should not have income on their own and are unemployed. Excluding them from our analysis should therefore be okay. However, in order not to skew our analysis we must assume that this (i.e. children not having income and not being employed) holds true across different races.

Exercise 7

Let's limit our attention to people who are currently working. We can do this using empstat. Remember you can use value_counts() to see what values of empstat are in the data!

In [115…
```
acs_sub2['empstat'].value_counts()
```

Out[115…
```
employed              148758
not in labor force    104676
unemployed              7727
n/a                     3942
Name: empstat, dtype: int64
```

In [116…
```
print("Subsetting the dataset for the people for whom empstat is equal to '
acs_emp = acs.loc[acs['empstat'] == "employed"]
acs_emp
```

```
Subsetting the dataset for the people for whom empstat is equal to "employe
d", i.e. people that are employed.
```

Out[116…

	year	datanum	serial	cbserial	numprec	subsamp	hhwt	hhtype	
1	2017	1	1200045	2.017001e+12	6	79	25	male householder, no wife present	2
2	2017	1	70831	2.017000e+12	1 person record	36	57	male householder, living alone	2
5	2017	1	563897	2.017001e+12	3	19	66	female householder, no husband present	2.

9	2017	1	856859	2.017001e+12	5	69	12	married-couple family household	2.
10	2017	1	175930	2.017001e+12	9	72	171	married-couple family household	2.
...	...	...	...	...	...	...	...	...	
318995	2017	1	46231	2.017001e+12	3	36	104	married-couple family household	2.
318999	2017	1	734396	2.017001e+12	4	78	100	married-couple family household	2
319001	2017	1	510444	2.017001e+12	2	43	152	female householder, no husband present	2.
319002	2017	1	1220474	2.017001e+12	4	16	148	married-couple family household	2
319003	2017	1	219435	2.017000e+12	2	17	47	married-couple family household	2.

148758 rows × 104 columns

Exercise 8

Now let's estimate the racial income gap in the United States.

In [117…

```
print("Find race variable and thus print out all columns")
pd.set_option('display.max_columns', None)
acs.head()
```

```
Find race variable and thus print out all columns
```

Out[117…

	year	datanum	serial	cbserial	numprec	subsamp	hhwt	hhtype	cl
0	2017	1	177686	2.017001e+12	9	64	55	female householder, no husband present	2.01700
1	2017	1	1200045	2.017001e+12	6	79	25	male householder, no wife present	2.01701
2	2017	1	70831	2.017000e+12	1 person record	36	57	male householder, living alone	2.01700
3	2017	1	557128	2.017001e+12	2	10	98	married-couple family household	2.01700
4	2017	1	614890	2.017001e+12	4	96	54	married-couple family household	2.01700

What is the average salary for employed Black Americans, and what is the average salary for employed White Americans? In percentage terms, how much more does the average White American make than the average Black American?

In [118…

```
acs_emp.groupby('race', as_index=False)['inctot'].mean().round(2)
```

Out[118…

	race	inctot
0	white	60473.15
1	black/african american/negro	41747.95
2	american indian or alaska native	37996.52
3	chinese	72804.92
4	japanese	78906.74
5	other asian or pacific islander	66647.74
6	other race, nec	34989.40
7	two major races	49021.15
8	three or more major races	49787.18

- The average salary for employed Black Americans is USD 41,747.95, while the average income for employed White Americans is USD 60,473.15.

In [119…

```
k = 41747.95/100
pd = 60473.15/k
pdr = pd -100
pdrr = round(pdr, 2)
f'On average employed White Americans make {pdrr}% more money than employed
```

Out[119…

```
'On average employed White Americans make 44.85% more money than employed B
lack Americans, which means their income is almost twice as high.'
```