DSHBA - Data Wrangling

Student: Gabriel Yugo Nascimento Kishida
SIT ID: Z122232

Code for Setup:

```
import pandas as pd
df_raw = pd.read_csv('tobacco.csv')
```

Task 1

Print the ratio of the number of missing records over the number of all records for "Response" column. Format: 0.XXX (3 decimal places)

Code for this task:

```
print("Task 1: Find the ratio of the number of missing records over the
number of all records:\n")
df_missing = df_raw[df_raw['Response'].isnull()]
ratio = len(df_missing)/len(df_raw)
print("Ratio is:", round(ratio,3))
```

Output obtained:

```
Task 1: Find the ratio of the number of missing records over the number
of all records:

Ratio is: 0.653
```

Task 2

Drop the missing records in the "Response" column and print the output (ratio) in the previous task again.

Code for this task:

```
print("Task 2: Drop the missing records in \"Response\" column and print the
ratio again:\n")
df_clean = df_raw.dropna(subset="Response")
df_missing = df_clean[df_clean['Response'].isnull()]
ratio = len(df_missing)/len(df_clean)
print("Ratio is:", round(ratio,3))
```

Output obtained:

```
Task 2: Drop the missing records in "Response" column and print the ratio again:
Ratio is: 0.0
```

Comment: this task was a bit ambiguous. Were we supposed to calculate the ratio using the amount of null responses after the cleaning? Compare it to the data before the cleaning? In this case, I compared the data that was null after the cleaning, and the complete data after cleaning – which is why the output is 0, since all null responses were removed.

Task 3

Print the unique values of the "Race" column and replace it with numeric ID (0,1,2,...) for each unique value.

Code for this task:

```
print("Task 3: Print the unique values of the \"Race\" column and replace
them with numeric ID:\n")
races = df_raw["Race"].unique()
races_id = [i for i in range(0, len(races))]
df_races_sorted = df_raw
print("Before ID attribution:")
print(df_races_sorted["Race"],"\n")
df_races_sorted["Race"] = df_races_sorted["Race"].replace(races,races_id)
print("After ID attribution:")
print(df_races_sorted["Race"])
```

Output obtained:

```
Task 3: Print the unique values of the "Race" column and replace them with
numeric ID:
Before ID attribution:
0 All Races
1 Hispanic
2 African American
3 African American
4 White
...
43336 All Races
43337 All Races
43338 All Races
43339 All Races
43340 All Races
Name: Race, Length: 43341, dtype: object

After ID attribution:
0 0
1 1
```

```
2 2
3 2
4 3
..
43336 0
43337 0
43338 0
43339 0
43340 0
Name: Race, Length: 43341, dtype: int64
```