fix_datatypes_air_pollution

October 26, 2017

1 Fixing `air_pollution_score` Data Type

- 2008: convert string to float
- 2018: convert int to float

```
In [1]: # load datasets
        import pandas as pd

In [2]: df_08 = pd.read_csv('data_08.csv')
        df_08.head(1)

Out[2]:        model  displ  cyl    trans drive      fuel veh_class  \
        0  ACURA MDX    3.7    6  Auto-S5   4WD  Gasoline       SUV

          air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
        0                   7       15      20      17                    4       no

In [3]: df_18 = pd.read_csv('data_18.csv')
        df_18.head(1)

Out[3]:        model  displ  cyl       trans drive      fuel  veh_class  \
        0  ACURA RDX    3.5    6  SemiAuto-6   2WD  Gasoline  small SUV

           air_pollution_score city_mpg hwy_mpg cmb_mpg  greenhouse_gas_score smartway
        0                    3       20      28      23                     5       No

In [4]: df_08['air_pollution_score'].value_counts()

Out[4]: 6      500
        7      398
        9.5     80
        9        7
        6/4      1
        Name: air_pollution_score, dtype: int64

In [6]: df_18['air_pollution_score'].value_counts()
```

```
Out[6]: 3    372
        5    187
        7    139
        1     89
        6      7
        Name: air_pollution_score, dtype: int64
```

```
In [9]: # try using Pandas to_numeric or astype function to convert the
        # 2008 air_pollution_score column to float -- this won't work
        df_08['air_pollution_score'].str.contains('/')
```

```
Out[9]: 0      False
        1      False
        2      False
        3      False
        4      False
        5      False
        6      False
        7      False
        8      False
        9      False
        10     False
        11     False
        12     False
        13     False
        14     False
        15     False
        16     False
        17     False
        18     False
        19     False
        20     False
        21     False
        22     False
        23     False
        24     False
        25     False
        26     False
        27     False
        28     False
        29     False
               ...
        956    False
        957    False
        958    False
        959    False
        960    False
        961    False
```

```
962    False
963    False
964    False
965    False
966    False
967    False
968    False
969    False
970    False
971    False
972    False
973    False
974    False
975    False
976    False
977    False
978    False
979    False
980    False
981    False
982    False
983    False
984    False
985    False
Name: air_pollution_score, Length: 986, dtype: bool
```

2 Figuring out the issue

Looks like this isn't going to be as simple as converting the datatype. According to the error above, the value at row 582 is "6/4" - let's check it out.

```
In [5]: df_08.iloc[582]
```

```
Out[5]: model                       MERCEDES-BENZ C300
        displ                                        3
        cyl                                          6
        trans                                  Auto-L7
        drive                                      2WD
        fuel                               ethanol/gas
        veh_class                            small car
        air_pollution_score                        6/4
        city_mpg                                 13/18
        hwy_mpg                                  19/25
        cmb_mpg                                  15/21
        greenhouse_gas_score                       7/6
        smartway                                    no
        Name: 582, dtype: object
```

3 It's not just the air pollution score!

The mpg columns and greenhouse gas scores also seem to have the same problem - maybe that's why these were all saved as strings! According to this link, which I found from the PDF documentation:

```
"If a vehicle can operate on more than one type of fuel, an estimate is provided for each fuel t
```

Ohh.. so all vehicles with more than one fuel type, or hybrids, like the one above (it uses ethanol AND gas) will have a string that holds two values - one for each. This is a little tricky, so I'm going to show you how to do it with the 2008 dataset, and then you'll try it with the 2018 dataset.

```
In [8]: # First, let's get all the hybrids in 2008
        hb_08 = df_08[df_08['fuel'].str.contains('/')]
        hb_08
```

```
Out[8]:                   model  displ  cyl    trans drive         fuel  veh_class  \
        582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD  ethanol/gas  small car

            air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
        582                 6/4    13/18   19/25   15/21                  7/6       no
```

Looks like this dataset only has one! The 2018 has MANY more - but don't worry - the steps I'm taking here will work for that as well!

```
In [10]: # hybrids in 2018
         hb_18 = df_18[df_18['fuel'].str.contains('/')]
         hb_18
```

```
Out[10]:                                model  displ  cyl       trans drive  \
         52                          BMW 330e    2.0    4  SemiAuto-8   2WD
         78                          BMW 530e    2.0    4  SemiAuto-8   2WD
         79                          BMW 530e    2.0    4  SemiAuto-8   4WD
         92                          BMW 740e    2.0    4  SemiAuto-8   4WD
         189                CHEVROLET Impala    3.6    6  SemiAuto-6   2WD
         195          CHEVROLET Silverado 15    4.3    6      Auto-6   2WD
         196          CHEVROLET Silverado 15    4.3    6      Auto-6   4WD
         197          CHEVROLET Silverado 15    5.3    8      Auto-6   2WD
         212         CHEVROLET Suburban 1500    5.3    8      Auto-6   2WD
         214         CHEVROLET Suburban 1500    5.3    8      Auto-6   4WD
         216            CHEVROLET Tahoe 1500    5.3    8      Auto-6   2WD
         218            CHEVROLET Tahoe 1500    5.3    8      Auto-6   4WD
         225                  CHEVROLET Volt    1.5    4         CVT   2WD
         226                  CHEVROLET Volt    1.5    4         CVT   2WD
         227                    CHRYSLER 300    3.6    6      Auto-8   2WD
         229                    CHRYSLER 300    3.6    6      Auto-8   4WD
         244                  DODGE Charger    3.6    6      Auto-8   2WD
         246                  DODGE Charger    3.6    6      Auto-8   4WD
```

```
300  FORD Fusion Energi Plug-in Hybrid    2.0    4         CVT   2WD
326                   GMC Sierra 15    4.3    6      Auto-6   2WD
327                   GMC Sierra 15    4.3    6      Auto-6   4WD
328                   GMC Sierra 15    5.3    8      Auto-6   2WD
345                  GMC Yukon 1500    5.3    8      Auto-6   2WD
347                  GMC Yukon 1500    5.3    8      Auto-6   4WD
351               GMC Yukon 1500 XL    5.3    8      Auto-6   2WD
354               GMC Yukon XL 1500    5.3    8      Auto-6   4WD
442                   JEEP Cherokee    2.4    4      Auto-9   2WD
444                   JEEP Cherokee    2.4    4      Auto-9   4WD
462                    KARMA Revero    2.0    4      Auto-1   2WD
571        MERCEDES-BENZ CLA250 4Matic    2.0    4   AutoMan-7   4WD
578        MERCEDES-BENZ GLA250 4Matic    2.0    4   AutoMan-7   4WD
584        MERCEDES-BENZ GLE350 4Matic    3.5    6      Auto-7   4WD
616     MINI Cooper SE Countryman All4    1.5    3  SemiAuto-6   4WD
742              TOYOTA Sequoia FFV    5.7    8  SemiAuto-6   4WD
747               TOYOTA Tundra FFV    5.7    8  SemiAuto-6   4WD
777                       VOLVO S90    2.0    4  SemiAuto-8   4WD
789                     VOLVO XC 60    2.0    4  SemiAuto-8   4WD
793                     VOLVO XC 90    2.0    4  SemiAuto-8   4WD

                    fuel     veh_class  air_pollution_score city_mpg hwy_mpg  \
52   Gasoline/Electricity     small car                    3    28/66   34/78
78   Gasoline/Electricity     small car                    7    27/70   31/75
79   Gasoline/Electricity     small car                    7    27/66   31/68
92   Gasoline/Electricity     large car                    3    25/62   29/68
189           Ethanol/Gas     large car                    5    14/18   20/28
195           Ethanol/Gas        pickup                    5    12/18   16/24
196           Ethanol/Gas        pickup                    5    12/17   15/22
197           Ethanol/Gas        pickup                    3    12/16   17/23
212           Ethanol/Gas  standard SUV                    3    12/16   17/23
214           Ethanol/Gas  standard SUV                    3    11/16   15/22
216           Ethanol/Gas  standard SUV                    3    12/16   17/23
218           Ethanol/Gas  standard SUV                    3    11/16   16/22
225  Gasoline/Electricity     small car                    3   43/113   42/99
226  Gasoline/Electricity     small car                    7   43/113   42/99
227           Ethanol/Gas     large car                    3    14/19   22/30
229           Ethanol/Gas     large car                    3    13/18   20/27
244           Ethanol/Gas     large car                    3    14/19   22/30
246           Ethanol/Gas     large car                    3    13/18   20/27
300  Gasoline/Electricity   midsize car                    7   43/102   41/91
326           Ethanol/Gas        pickup                    5    12/18   16/24
327           Ethanol/Gas        pickup                    5    12/17   15/22
328           Ethanol/Gas        pickup                    3    12/16   17/23
345           Ethanol/Gas  standard SUV                    3    12/16   17/23
347           Ethanol/Gas  standard SUV                    3    11/16   16/22
351           Ethanol/Gas  standard SUV                    3    12/16   17/23
354           Ethanol/Gas  standard SUV                    3    11/16   15/22
```

```
442          Ethanol/Gas     small SUV                       3    15/21   22/30
444          Ethanol/Gas     small SUV                       3    14/21   21/28
462  Gasoline/Electricity    small car                       1    20/59   21/61
571          Ethanol/Gas     small car                       5    17/24   24/32
578          Ethanol/Gas     small SUV                       5    17/23   23/31
584          Ethanol/Gas  standard SUV                       3    13/18   17/22
616  Gasoline/Electricity   midsize car                      3    28/63   27/66
742          Ethanol/Gas  standard SUV                       5     9/13   13/17
747          Ethanol/Gas        pickup                       5     9/13   12/17
777  Gasoline/Electricity   midsize car                      7    26/70   33/72
789  Gasoline/Electricity     small SUV                      7    26/60   28/58
793  Gasoline/Electricity  standard SUV                      7    26/63   30/61

    cmb_mpg  greenhouse_gas_score smartway
52   30/71                     10      Yes
78   29/72                     10    Elite
79   28/67                     10    Elite
92   27/64                      9      Yes
189  16/22                      4       No
195  14/20                      4       No
196  13/19                      3       No
197  14/19                      3       No
212  14/19                      3       No
214  12/18                      3       No
216  14/19                      3       No
218  13/18                      3       No
225  42/106                    10      Yes
226  42/106                    10    Elite
227  17/23                      5       No
229  16/21                      4       No
244  17/23                      5       No
246  16/21                      4       No
300  42/97                     10    Elite
326  14/20                      4       No
327  13/19                      3       No
328  14/19                      3       No
345  14/19                      3       No
347  13/18                      3       No
351  14/19                      3       No
354  12/18                      3       No
442  18/25                      5       No
444  17/23                      5       No
462  20/60                     10       No
571  20/27                      6       No
578  19/26                      5       No
584  14/19                      3       No
616  27/65                      9      Yes
742  10/14                      1       No
```

```
747   10/15                          2       No
777   29/71                         10    Elite
789   26/59                         10    Elite
793   27/62                         10    Elite
```

We're going to take each hybrid row and split them into two new rows - one with values for the first fuel type (values before the "/"), and the other with values for the second fuel type (values after the "/"). Let's separate them with two dataframes!

```
In [11]: # create two copies of the 2008 hybrids dataframe
         df1 = hb_08.copy()  # data on first fuel type of each hybrid vehicle
         df2 = hb_08.copy()  # data on second fuel type of each hybrid vehicle

         # Each one should look like this
         df1
```

```
Out[11]:                    model  displ  cyl    trans drive         fuel  veh_class  \
         582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD  ethanol/gas  small car

             air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
         582                 6/4    13/18   19/25   15/21                  7/6       no
```

For this next part, we're going use Pandas' apply function. See the docs here.

```
In [12]: # columns to split by "/"
         split_columns = ['fuel', 'air_pollution_score', 'city_mpg', 'hwy_mpg', 'cmb_mpg', 'gree

         # apply split function to each column of each dataframe copy
         for c in split_columns:
             df1[c] = df1[c].apply(lambda x: x.split("/")[0])
             df2[c] = df2[c].apply(lambda x: x.split("/")[1])
```

```
In [13]: # this dataframe holds info for the FIRST fuel type of the hybrid
         # aka the values before the "/"s
         df1
```

```
Out[13]:                    model  displ  cyl    trans drive     fuel  veh_class  \
         582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD  ethanol  small car

             air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
         582                   6       13      19      15                    7       no
```

```
In [14]: # this dataframe holds info for the SECOND fuel type of the hybrid
         # aka the values before the "/"s
         df2
```

```
Out[14]:                    model  displ  cyl    trans drive fuel  veh_class  \
         582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD  gas  small car

             air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
         582                   4       18      25      21                    6       no
```

```
In [15]: # combine dataframes to add to the original dataframe
         new_rows = df1.append(df2)

         # now we have separate rows for each fuel type of each vehicle!
         new_rows
```

```
Out[15]:                    model  displ  cyl    trans drive     fuel  veh_class  \
         582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD  ethanol  small car
         582  MERCEDES-BENZ C300    3.0    6  Auto-L7   2WD      gas  small car

             air_pollution_score city_mpg hwy_mpg cmb_mpg greenhouse_gas_score smartway
         582                   6       13      19      15                    7       no
         582                   4       18      25      21                    6       no
```

```
In [16]: hb_08.index
```

```
Out[16]: Int64Index([582], dtype='int64')
```

```
In [17]: hb_18.index
```

```
Out[17]: Int64Index([ 52,  78,  79,  92, 189, 195, 196, 197, 212, 214, 216, 218, 225,
                     226, 227, 229, 244, 246, 300, 326, 327, 328, 345, 347, 351, 354,
                     442, 444, 462, 571, 578, 584, 616, 742, 747, 777, 789, 793],
                    dtype='int64')
```

```
In [18]: # drop the original hybrid rows
         df_08.drop(hb_08.index, inplace=True)

         # add in our newly separated rows
         df_08 = df_08.append(new_rows, ignore_index=True)
```

```
In [19]: # check that all the original hybrid rows with "/"s are gone
         df_08[df_08['fuel'].str.contains('/')]
```

```
Out[19]: Empty DataFrame
         Columns: [model, displ, cyl, trans, drive, fuel, veh_class, air_pollution_score, city_m
         Index: []
```

```
In [20]: df_08.shape
```

```
Out[20]: (987, 13)
```

4 Repeat this process for the 2018 dataset

```
In [21]: # create two copies of the 2018 hybrids dataframe, hb_18
         df1 = hb_18.copy()  # data on first fuel type of each hybrid vehicle
         df2 = hb_18.copy()
```

4.0.1 Split values for `fuel`, `city_mpg`, `hwy_mpg`, `cmb_mpg`

You don't need to split for `air_pollution_score` or `greenhouse_gas_score` here because these columns are already ints in the 2018 dataset.

```
In [22]: # list of columns to split
         split_columns = ['fuel', 'city_mpg', 'hwy_mpg', 'cmb_mpg']

         # apply split function to each column of each dataframe copy
         for c in split_columns:
             df1[c] = df1[c].apply(lambda x: x.split("/")[0])
             df2[c] = df2[c].apply(lambda x: x.split("/")[1])
```

```
In [23]: df1.shape
```

```
Out[23]: (38, 13)
```

```
In [24]: df2.shape
```

```
Out[24]: (38, 13)
```

```
In [25]: # append the two dataframes
         new_rows = df1.append(df2)

         # drop each hybrid row from the original 2018 dataframe
         # do this by using Pandas drop function with hb_18's index
         df_18.drop(hb_18.index, inplace=True)

         # append new_rows to df_18
         df_18 = df_18.append(new_rows, ignore_index=True)
```

```
In [26]: # check that they're gone
         df_18[df_18['fuel'].str.contains('/')]
```

```
Out[26]: Empty DataFrame
         Columns: [model, displ, cyl, trans, drive, fuel, veh_class, air_pollution_score, city_m
         Index: []
```

```
In [27]: df_18.shape
```

```
Out[27]: (832, 13)
```

4.0.2 Now we can comfortably continue the changes needed for `air_pollution_score`! Here they are again:

- 2008: convert string to float
- 2018: convert int to float

```
In [28]: df_08['air_pollution_score'].value_counts()
```

```
Out[28]: 6      501
         7      398
         9.5     80
         9        7
         4        1
         Name: air_pollution_score, dtype: int64
```

```
In [29]: # convert string to float for 2008 air pollution column
         df_08['air_pollution_score'] = df_08['air_pollution_score'].astype(float)
```

```
In [30]: df_08['air_pollution_score'].value_counts()
```

```
Out[30]: 6.0    501
         7.0    398
         9.5     80
         9.0      7
         4.0      1
         Name: air_pollution_score, dtype: int64
```

```
In [31]: # convert int to float for 2018 air pollution column
         df_18['air_pollution_score'] = df_18['air_pollution_score'].astype(float)
```

```
In [32]: df_08['air_pollution_score'].value_counts()
```

```
Out[32]: 6.0    501
         7.0    398
         9.5     80
         9.0      7
         4.0      1
         Name: air_pollution_score, dtype: int64
```

```
In [33]: df_08.to_csv('data_08.csv', index=False)
         df_18.to_csv('data_18.csv', index=False)
```

```
In [ ]:
```