Pandas - Assignment 01

November 23, 2021

1 Assignment 01: Evaluate the FAA Dataset

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

1: VIew and import the dataset

[1]:
```
#Import necessary libraries
import pandas as pd
```

[2]:
```
#Import the FAA (Federal Aviation Authority) dataset
df_faa_dataset = pd.read_csv("D:/COURSES/Artificial Intellegence Engineer/Data␣
↪Analytics With Python/Analyse the Federal Aviation Authority Dataset using␣
↪Pandas/SUBMISSION/SOURCE CODE/faa_ai_prelim.csv")
```

2: View and understand the dataset

[3]:
```
#View the dataset shape
df_faa_dataset.shape
```

[3]:
```
(83, 42)
```

[4]:
```
#View the first five observations
df_faa_dataset.head()
```

[4]:
```
  UPDATED ENTRY_DATE EVENT_LCL_DATE EVENT_LCL_TIME LOC_CITY_NAME  \
0      No  19-FEB-16      19-FEB-16      00:45:00Z    MARSHVILLE
1      No  19-FEB-16      18-FEB-16      23:55:00Z     TAVERNIER
2      No  19-FEB-16      18-FEB-16      22:14:00Z       TRENTON
3      No  19-FEB-16      18-FEB-16      17:10:00Z     ASHEVILLE
4      No  19-FEB-16      18-FEB-16      00:26:00Z     TALKEETNA

   LOC_STATE_NAME LOC_CNTRY_NAME  \
```

```
0  North Carolina            NaN
1         Florida            NaN
2      New Jersey            NaN
3  North Carolina            NaN
4          Alaska            NaN

                                          RMK_TEXT EVENT_TYPE_DESC  \
0  AIRCRAFT CRASHED INTO TREES, THE 1 PERSON ON B…        Accident
1  AIRCRAFT ON LANDING WENT OFF THE END OF THE RU…        Incident
2  AIRCRAFT ON FINAL SUSTAINED A BIRD STRIKE, LAN…        Incident
3  AIRCRAFT ON LANDING, GEAR COLLAPSED, ASHEVILLE…        Incident
4  AIRCRAFT ON LANDING, NOSE GEAR COLLAPSED, TALK…        Incident

                  FSDO_DESC  … PAX_INJ_NONE PAX_INJ_MINOR PAX_INJ_SERIOUS  \
0     FAA Charlotte FSDO-68  …          NaN           NaN              NaN
1         FAA Miami FSDO-19  …          NaN           NaN              NaN
2  FAA Philadelphia FSDO-17  …          NaN           NaN              NaN
3     FAA Charlotte FSDO-68  …          NaN           NaN              NaN
4     FAA Anchorage FSDO-03  …          NaN           1.0              NaN

  PAX_INJ_FATAL PAX_INJ_UNK  GRND_INJ_NONE GRND_INJ_MINOR GRND_INJ_SERIOUS  \
0           NaN         NaN            NaN            NaN               NaN
1           NaN         NaN            NaN            NaN               NaN
2           NaN         NaN            NaN            NaN               NaN
3           NaN         NaN            NaN            NaN               NaN
4           NaN         NaN            NaN            NaN               NaN

  GRND_INJ_FATAL  GRND_INJ_UNK
0            NaN           NaN
1            NaN           NaN
2            NaN           NaN
3            NaN           NaN
4            NaN           NaN

[5 rows x 42 columns]
```

[5]:
```
#View all the columns present in the dataset
df_faa_dataset.columns
```

[5]:
```
Index(['UPDATED', 'ENTRY_DATE', 'EVENT_LCL_DATE', 'EVENT_LCL_TIME',
       'LOC_CITY_NAME', 'LOC_STATE_NAME', 'LOC_CNTRY_NAME', 'RMK_TEXT',
       'EVENT_TYPE_DESC', 'FSDO_DESC', 'REGIST_NBR', 'FLT_NBR', 'ACFT_OPRTR',
       'ACFT_MAKE_NAME', 'ACFT_MODEL_NAME', 'ACFT_MISSING_FLAG',
       'ACFT_DMG_DESC', 'FLT_ACTIVITY', 'FLT_PHASE', 'FAR_PART', 'MAX_INJ_LVL',
       'FATAL_FLAG', 'FLT_CRW_INJ_NONE', 'FLT_CRW_INJ_MINOR',
       'FLT_CRW_INJ_SERIOUS', 'FLT_CRW_INJ_FATAL', 'FLT_CRW_INJ_UNK',
       'CBN_CRW_INJ_NONE', 'CBN_CRW_INJ_MINOR', 'CBN_CRW_INJ_SERIOUS',
```

```
       'CBN_CRW_INJ_FATAL', 'CBN_CRW_INJ_UNK', 'PAX_INJ_NONE', 'PAX_INJ_MINOR',
       'PAX_INJ_SERIOUS', 'PAX_INJ_FATAL', 'PAX_INJ_UNK', 'GRND_INJ_NONE',
       'GRND_INJ_MINOR', 'GRND_INJ_SERIOUS', 'GRND_INJ_FATAL', 'GRND_INJ_UNK'],
      dtype='object')
```

3: Extract the following attributes from the dataset:

1. Aircraft make name
2. State name
3. Aircraft model name
4. Text information
5. Flight phase
6. Event description type
7. Fatal flag

[6]:
```
#Create a new dataframe with only the required columns
df_analyze_dataset = df_faa_dataset[['LOC_STATE_NAME', 'RMK_TEXT',␣
↪'EVENT_TYPE_DESC', 'ACFT_MAKE_NAME',
                                    'ACFT_MODEL_NAME', 'FLT_PHASE',␣
↪'FATAL_FLAG']]
```

[7]:
```
#View the type of the object
type(df_analyze_dataset)
```

[7]:
```
pandas.core.frame.DataFrame
```

[8]:
```
#Check if the dataframe contains all the required attributes
df_analyze_dataset.head()
```

[8]:

	LOC_STATE_NAME	RMK_TEXT	\
0	North Carolina	AIRCRAFT CRASHED INTO TREES, THE 1 PERSON ON B…	
1	Florida	AIRCRAFT ON LANDING WENT OFF THE END OF THE RU…	
2	New Jersey	AIRCRAFT ON FINAL SUSTAINED A BIRD STRIKE, LAN…	
3	North Carolina	AIRCRAFT ON LANDING, GEAR COLLAPSED, ASHEVILLE…	
4	Alaska	AIRCRAFT ON LANDING, NOSE GEAR COLLAPSED, TALK…	

	EVENT_TYPE_DESC	ACFT_MAKE_NAME	ACFT_MODEL_NAME	FLT_PHASE	FATAL_FLAG
0	Accident	BEECH	36	UNKNOWN (UNK)	Yes
1	Incident	VANS	RV7	LANDING (LDG)	NaN
2	Incident	CESSNA	172	APPROACH (APR)	NaN
3	Incident	LANCAIR	235	LANDING (LDG)	NaN
4	Incident	CESSNA	172	LANDING (LDG)	NaN

4. Clean the dataset and replace the fatal flag NaN with "No"

[9]:
```
#Replace all Fatal Flag missing values with the required output
df_analyze_dataset['FATAL_FLAG'].fillna(value="No",inplace=True)
```

```
C:\Users\amalp\AppData\Local\Programs\Python\Python310\lib\site-
packages\pandas\core\generic.py:6392: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
```

[10]:
```
#Verify if the missing values are replaced
df_analyze_dataset.head()
```

[10]:

	LOC_STATE_NAME	RMK_TEXT	\
0	North Carolina	AIRCRAFT CRASHED INTO TREES, THE 1 PERSON ON B…	
1	Florida	AIRCRAFT ON LANDING WENT OFF THE END OF THE RU…	
2	New Jersey	AIRCRAFT ON FINAL SUSTAINED A BIRD STRIKE, LAN…	
3	North Carolina	AIRCRAFT ON LANDING, GEAR COLLAPSED, ASHEVILLE…	
4	Alaska	AIRCRAFT ON LANDING, NOSE GEAR COLLAPSED, TALK…	

	EVENT_TYPE_DESC	ACFT_MAKE_NAME	ACFT_MODEL_NAME	FLT_PHASE	FATAL_FLAG
0	Accident	BEECH	36	UNKNOWN (UNK)	Yes
1	Incident	VANS	RV7	LANDING (LDG)	No
2	Incident	CESSNA	172	APPROACH (APR)	No
3	Incident	LANCAIR	235	LANDING (LDG)	No
4	Incident	CESSNA	172	LANDING (LDG)	No

[11]:
```
#Check the number of observations
df_analyze_dataset.shape
```

[11]:
```
(83, 7)
```

5. Remove all the observations where aircraft names are not available

[12]:
```
#Drop the unwanted values/observations from the dataset
df_final_dataset = df_analyze_dataset.dropna(subset=['ACFT_MAKE_NAME'])
```

6. Find the aircraft types and their occurrences in the dataset

[13]:
```
#Check the number of observations now to compare it with the original dataset␣
↪and see how many values have been dropped
df_final_dataset.shape
```

[13]:
```
(78, 7)
```

[14]:
```
#Group the dataset by aircraft name
aircraftType = df_final_dataset.groupby('ACFT_MAKE_NAME')
```

[15]:
```
#View the number of times each aircraft type appears in the dataset (Hint: use␣
↪the size() method)
aircraftType.size()
```

[15]:
```
ACFT_MAKE_NAME
AERO COMMANDER              1
AERONCA                     1
AEROSTAR INTERNATIONAL      1
AIRBUS                      1
BEECH                       9
BELL                        2
BOEING                      3
CESSNA                     23
CHAMPION                    2
CHRISTEN                    1
CONSOLIDATED VULTEE         1
EMBRAER                     1
ENSTROM                     1
FAIRCHILD                   1
FLIGHT DESIGN               1
GLOBE                       1
GREAT LAKES                 1
GRUMMAN                     1
GULFSTREAM                  1
HUGHES                      1
LANCAIR                     2
MAULE                       1
MOONEY                      4
NORTH AMERICAN              1
PIPER                      10
PITTS                       1
SAAB                        1
SABRELINER                  1
SOCATA                      2
VANS                        1
dtype: int64
```

7: Display the observations where fatal flag is "Yes"

[16]:
```
#Group the dataset by fatal flag
fatalAccedents = df_final_dataset.groupby('FATAL_FLAG')
```

[17]:
```
#View the total number of fatal and non-fatal accidents
fatalAccedents.size()
```

[17]:
```
FATAL_FLAG
No     71
Yes     7
dtype: int64
```

[18]:
```
#Create a new dataframe to view only the fatal accidents (Fatal Flag values =␣
↪Yes)
```

```
accidents_with_fatality = fatalAccedents.get_group('Yes')
```

[19]:
```
accidents_with_fatality.head()
```

[19]:

	LOC_STATE_NAME	RMK_TEXT	\
0	North Carolina	AIRCRAFT CRASHED INTO TREES, THE 1 PERSON ON B…	
53	Florida	AIRCRAFT CRASHED UNDER UNKNOWN CIRCUMSTANCES. …	
55	California	AIRCRAFT CRASHED UNDER UNKNOWN CIRCUMSTANCES A…	
79	Arizona	AIRCRAFT CRASHED UNDER UNKNOWN CIRCUMSTANCES, …	
80	California	N9872R, BEECH M35 AIRCRAFT, AND N5057G, BELLAN…	

	EVENT_TYPE_DESC	ACFT_MAKE_NAME	ACFT_MODEL_NAME	FLT_PHASE	FATAL_FLAG
0	Accident	BEECH	36	UNKNOWN (UNK)	Yes
53	Accident	PIPER	PA28	UNKNOWN (UNK)	Yes
55	Accident	FLIGHT DESIGN	CTLS	UNKNOWN (UNK)	Yes
79	Accident	NORTH AMERICAN	F51	UNKNOWN (UNK)	Yes
80	Accident	CHAMPION	8KCAB	UNKNOWN (UNK)	Yes

[ ]:
```

```