Data cleaning - Students' data

Data cleaning steps that we are going to follow:

1. Deleting redundant columns
2. Renaming columns
3. Dropping duplicates
4. Remove NaN(null) values from the dataset
5. Check for some more transformations

In [1]:
```
import pandas as pd
```

In [2]:
```
pd.__version__
```

Out[2]:
```
'2.2.3'
```

In [4]:
```
data = pd.read_csv(r"C:\Users\balus\OneDrive\Desktop\Projects\Data cleaning - Pandas-student info\Students_info
```

In [5]:
```
data
```

Out[5]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSib
0	0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	
1	1	female	group C	some college	standard	NaN	married	sometimes	yes	
2	2	female	group B	master's degree	standard	none	single	sometimes	yes	
3	3	male	group A	associate's degree	free/reduced	none	married	never	no	
4	4	male	group C	some college	standard	none	married	sometimes	yes	
...	...	...	...	...	...	...	...	...	...	
30636	816	female	group D	high school	standard	none	single	sometimes	no	
30637	890	male	group E	high school	standard	none	single	regularly	no	
30638	911	female	NaN	high school	free/reduced	completed	married	sometimes	no	
30639	934	female	group D	associate's degree	standard	completed	married	regularly	no	
30640	960	male	group B	some college	standard	none	married	never	no	

30641 rows × 15 columns

In [ ]:
```
data.shape ## Data contains 15 columns & 30641 rows
```

Out[ ]:
```
(30641, 15)
```

In [ ]:
```
data.columns ## These are the names of the 15 columns
```

Out[ ]:
```
Index(['Unnamed: 0', 'Gender', 'EthnicGroup', 'ParentEduc', 'LunchType',
       'TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild',
       'NrSiblings', 'TransportMeans', 'WklyStudyHours', 'MathScore',
       'ReadingScore', 'WritingScore'],
      dtype='object')
```

In [ ]:
```
data.head() ## This shows the first 5 rows along with column names
```

Out[ ]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	ParentMaritalStatus	PracticeSport	IsFirstChild	NrSiblings
0	0	female	NaN	bachelor's degree	standard	none	married	regularly	yes	3.0
1	1	female	group C	some college	standard	NaN	married	sometimes	yes	0.0
2	2	female	group B	master's degree	standard	none	single	sometimes	yes	4.0
3	3	male	group A	associate's degree	free/reduced	none	married	never	no	1.0
4	4	male	group C	some college	standard	none	married	sometimes	yes	0.0

In [ ]:
```
data.info() ## It is showing datatypes, no of non-null entries & various other info about each column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30641 entries, 0 to 30640
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           30641 non-null  int64
 1   Gender               30641 non-null  object
 2   EthnicGroup          28801 non-null  object
 3   ParentEduc           28796 non-null  object
 4   LunchType            30641 non-null  object
 5   TestPrep             28811 non-null  object
 6   ParentMaritalStatus  29451 non-null  object
 7   PracticeSport        30010 non-null  object
 8   IsFirstChild         29737 non-null  object
 9   NrSiblings           29069 non-null  float64
 10  TransportMeans       27507 non-null  object
 11  WklyStudyHours       29686 non-null  object
 12  MathScore            30641 non-null  int64
 13  ReadingScore         30641 non-null  int64
 14  WritingScore         30641 non-null  int64
dtypes: float64(1), int64(4), object(10)
memory usage: 3.5+ MB
```

In [12]:
```
'''1) Deleting redundant columns
'Unnamed: 0','TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild' -
These are the redundant columns (not so important columns) that we are deleting'''
data.drop(columns=['Unnamed: 0','TestPrep', 'ParentMaritalStatus', 'PracticeSport', 'IsFirstChild'],inplace=Tru
```

In [13]:
```
data
```

Out[13]:

	Gender	EthnicGroup	ParentEduc	LunchType	NrSiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore	W
0	female	NaN	bachelor's degree	standard	3.0	school_bus	< 5	71	71	
1	female	group C	some college	standard	0.0	NaN	5 - 10	69	90	
2	female	group B	master's degree	standard	4.0	school_bus	< 5	87	93	
3	male	group A	associate's degree	free/reduced	1.0	NaN	5 - 10	45	56	
4	male	group C	some college	standard	0.0	school_bus	5 - 10	76	78	
...	...	...	...	...	...	...	...	...	...	
30636	female	group D	high school	standard	2.0	school_bus	5 - 10	59	61	
30637	male	group E	high school	standard	1.0	private	5 - 10	58	53	
30638	female	NaN	high school	free/reduced	1.0	private	5 - 10	61	70	
30639	female	group D	associate's degree	standard	3.0	school_bus	5 - 10	82	90	
30640	male	group B	some college	standard	1.0	school_bus	5 - 10	64	60	

30641 rows × 10 columns

In [17]:
```
'''2) Renaming columns, we are going to rename column NrSiblings as Numberofsiblings '''
data.rename(columns={"NrSiblings":"Noofsiblings"},inplace=True)
```

In [18]:
```
data
```

Out[18]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
0	female	NaN	bachelor's degree	standard	3.0	school_bus	< 5	71	71
1	female	group C	some college	standard	0.0	NaN	5 - 10	69	90
2	female	group B	master's degree	standard	4.0	school_bus	< 5	87	93
3	male	group A	associate's degree	free/reduced	1.0	NaN	5 - 10	45	56
4	male	group C	some college	standard	0.0	school_bus	5 - 10	76	78
...	...	...	...	...	...	...	...	...	...
30636	female	group D	high school	standard	2.0	school_bus	5 - 10	59	61
30637	male	group E	high school	standard	1.0	private	5 - 10	58	53
30638	female	NaN	high school	free/reduced	1.0	private	5 - 10	61	70
30639	female	group D	associate's degree	standard	3.0	school_bus	5 - 10	82	90
30640	male	group B	some college	standard	1.0	school_bus	5 - 10	64	60

30641 rows × 10 columns

In [ ]:
```
'''3) Dropping duplicates - we are going to check how many duplicate records are there & delete them'''
data.duplicated()
```

Out[ ]:
```
np.int64(9)
```

In [ ]:
```
data.duplicated().sum() ## There are 9 duplicate rows in the data
```

Out[ ]:
```
np.int64(9)
```

In [22]:
```
data.drop_duplicates(inplace=True) ## deleting those duplicate rows
```

In [ ]:
```
data.duplicated().sum() ## now there are no duplicate rows
```

Out[ ]:
```
np.int64(0)
```

In [30]:
```
'''4) Remove NaN(null) values from the dataset - we are going to find Null values & remove them'''
data.isna() ## Showing True wherever there is null values & false wherever there non-null values
```

Out[30]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
0	False	True	False	False	False	False	False	False	False
1	False	False	False	False	False	True	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	True	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
30635	False	False	False	False	False	False	False	False	False
30636	False	False	False	False	False	False	False	False	False
30637	False	False	False	False	False	False	False	False	False
30638	False	True	False	False	False	False	False	False	False
30639	False	False	False	False	False	False	False	False	False

30632 rows × 10 columns

In [29]:
```
data.isna().sum() ## Shows No of null values column wise
```

Out[29]:
```
Gender              0
EthnicGroup      1840
ParentEduc       1845
LunchType           0
Noofsiblings     1571
TransportMeans   3134
WklyStudyHours    955
MathScore           0
ReadingScore        0
WritingScore        0
dtype: int64
```

In [33]:
```
data.isna().sum().sum() ##Shows total No of null values across all columns
```

Out[33]:
```
np.int64(9345)
```

In [ ]:
```
data.dropna(inplace=True) ## removing null records & saving it
```

In [ ]:
```
data.isna().sum().sum() ## now there are no null values
```

Out[ ]:
```
np.int64(0)
```

In [ ]:
```
## 5) Check for some more transformations
data["Gender"].unique() ## checking unique entries in Gender column
```

Out[ ]:
```
array(['female', 'male'], dtype=object)
```

In [ ]:
```
## Changing the gender values to 1 for female & 0 for male
data["Gender"] = data["Gender"].apply(lambda x:1 if x == "female" else 0)
```

In [ ]:
```
data ## we can see Index is not right order
```

Out[ ]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
2	1	group B	master's degree	standard	4.0	school_bus	< 5	87	93
4	0	group C	some college	standard	0.0	school_bus	5 - 10	76	78
5	1	group B	associate's degree	standard	1.0	school_bus	5 - 10	73	84
6	1	group B	some college	standard	1.0	private	5 - 10	85	93
7	0	group B	some college	free/reduced	1.0	private	> 10	41	43
...	...	...	...	...	...	...	...	...	...
30634	0	group A	associate's degree	free/reduced	2.0	school_bus	5 - 10	65	60
30635	0	group C	some college	standard	2.0	school_bus	5 - 10	58	53
30636	1	group D	high school	standard	2.0	school_bus	5 - 10	59	61
30637	0	group E	high school	standard	1.0	private	5 - 10	58	53
30639	1	group D	associate's degree	standard	3.0	school_bus	5 - 10	82	90

22343 rows × 10 columns

In [43]:
```
data.reset_index(inplace=True)
```

In [42]:
```
data
```

Out[42]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
2	1	group B	master's degree	standard	4.0	school_bus	< 5	87	93
4	0	group C	some college	standard	0.0	school_bus	5 - 10	76	78
5	1	group B	associate's degree	standard	1.0	school_bus	5 - 10	73	84
6	1	group B	some college	standard	1.0	private	5 - 10	85	93
7	0	group B	some college	free/reduced	1.0	private	> 10	41	43
...	...	...	...	...	...	...	...	...	...
30634	0	group A	associate's degree	free/reduced	2.0	school_bus	5 - 10	65	60
30635	0	group C	some college	standard	2.0	school_bus	5 - 10	58	53
30636	1	group D	high school	standard	2.0	school_bus	5 - 10	59	61
30637	0	group E	high school	standard	1.0	private	5 - 10	58	53
30639	1	group D	associate's degree	standard	3.0	school_bus	5 - 10	82	90

22343 rows × 10 columns

In [44]:
```
data.drop(columns="index",inplace=True)
```

In [45]:
```
data
```

Out[45]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
0	1	group B	master's degree	standard	4.0	school_bus	< 5	87	93
1	0	group C	some college	standard	0.0	school_bus	5 - 10	76	78
2	1	group B	associate's degree	standard	1.0	school_bus	5 - 10	73	84
3	1	group B	some college	standard	1.0	private	5 - 10	85	93
4	0	group B	some college	free/reduced	1.0	private	> 10	41	43
...	...	...	...	...	...	...	...	...	...
22338	0	group A	associate's degree	free/reduced	2.0	school_bus	5 - 10	65	60
22339	0	group C	some college	standard	2.0	school_bus	5 - 10	58	53
22340	1	group D	high school	standard	2.0	school_bus	5 - 10	59	61
22341	0	group E	high school	standard	1.0	private	5 - 10	58	53
22342	1	group D	associate's degree	standard	3.0	school_bus	5 - 10	82	90

22343 rows × 10 columns

In [ ]:
```
type(data["Noofsiblings"][0]) ## checking the datatype of the column Noofsiblings
```

Out[ ]:
```
numpy.float64
```

In [50]:
```
data["Noofsiblings"] =  data["Noofsiblings"].astype(int)
```

In [51]:
```
data
```

Out[51]:

	Gender	EthnicGroup	ParentEduc	LunchType	Noofsiblings	TransportMeans	WklyStudyHours	MathScore	ReadingScore
0	1	group B	master's degree	standard	4	school_bus	< 5	87	93
1	0	group C	some college	standard	0	school_bus	5 - 10	76	78
2	1	group B	associate's degree	standard	1	school_bus	5 - 10	73	84
3	1	group B	some college	standard	1	private	5 - 10	85	93
4	0	group B	some college	free/reduced	1	private	> 10	41	43
...	...	...	...	...	...	...	...	...	...
22338	0	group A	associate's degree	free/reduced	2	school_bus	5 - 10	65	60
22339	0	group C	some college	standard	2	school_bus	5 - 10	58	53
22340	1	group D	high school	standard	2	school_bus	5 - 10	59	61
22341	0	group E	high school	standard	1	private	5 - 10	58	53
22342	1	group D	associate's degree	standard	3	school_bus	5 - 10	82	90

22343 rows × 10 columns

Now our Data cleaning is done. It is time to export the cleaned data

In [52]:
```
data.to_excel("Cleaned_data.xlsx")
```

In [53]:
```
data.to_csv("Cleaned_data.csv")
```

Thank you