# practical2

April 18, 2024

## 1 Create an "Academic performance" dataset of students and perform the following operations using Python.

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df = pd.read_csv('lego_sets.csv')
```

[3]:
```
df.head(10)
```

[3]:

| | set_id | name | year | theme | subtheme | \ |
|---|---|---|---|---|---|---|
| 0 | 1-8 | Small house set | 1970 | Minitalia | NaN | |
| 1 | 2-8 | Medium house set | 1970 | Minitalia | NaN | |
| 2 | 3-6 | Medium house set | 1970 | Minitalia | NaN | |
| 3 | 4-4 | Large house set | 1970 | Minitalia | NaN | |
| 4 | 4-6 | Mini House and Vehicles | 1970 | Samsonite | Model Maker | |
| 5 | 078-1 | Roadway Base Plate 50X50 | 1970 | Samsonite | Supplemental | |
| 6 | 104-1 | 4.5V Replacement Motor | 1970 | Trains | Supplemental / 4.5V | |
| 7 | 126-1 | Steam Locomotive (Push) | 1970 | Trains | 4.5v | |
| 8 | 157-3 | 4 Car Auto Transport | 1970 | Samsonite | Model Maker | |
| 9 | 242-4 | Big Model Book | 1970 | Books | LEGO | |

| | themeGroup | category | pieces | minifigs | agerange_min | US_retailPrice | \ |
|---|---|---|---|---|---|---|---|
| 0 | Vintage | Normal | 67.0 | NaN | NaN | NaN | |
| 1 | Vintage | Normal | 109.0 | NaN | NaN | NaN | |
| 2 | Vintage | Normal | 158.0 | NaN | NaN | NaN | |
| 3 | Vintage | Normal | 233.0 | NaN | NaN | NaN | |
| 4 | Vintage | Normal | NaN | NaN | NaN | NaN | |
| 5 | Vintage | Normal | 1.0 | NaN | NaN | NaN | |
| 6 | Modern day | Normal | 1.0 | NaN | NaN | NaN | |
| 7 | Modern day | Normal | 60.0 | NaN | NaN | NaN | |
| 8 | Vintage | Normal | 65.0 | NaN | NaN | NaN | |
| 9 | Miscellaneous | Book | NaN | NaN | NaN | NaN | |

| | bricksetURL | \ |
|---|---|---|
| 0 | https://brickset.com/sets/1-8 | |
| 1 | https://brickset.com/sets/2-8 | |

```
2    https://brickset.com/sets/3-6
3    https://brickset.com/sets/4-4
4    https://brickset.com/sets/4-6
5  https://brickset.com/sets/078-1
6  https://brickset.com/sets/104-1
7  https://brickset.com/sets/126-1
8  https://brickset.com/sets/157-3
9  https://brickset.com/sets/242-4

                                        thumbnailURL  \
0    https://images.brickset.com/sets/small/1-8.jpg
1    https://images.brickset.com/sets/small/2-8.jpg
2    https://images.brickset.com/sets/small/3-6.jpg
3    https://images.brickset.com/sets/small/4-4.jpg
4                                                NaN
5  https://images.brickset.com/sets/small/078-1.jpg
6  https://images.brickset.com/sets/small/104-1.jpg
7  https://images.brickset.com/sets/small/126-1.jpg
8  https://images.brickset.com/sets/small/157-3.jpg
9  https://images.brickset.com/sets/small/242-4.jpg

                                             imageURL
0    https://images.brickset.com/sets/images/1-8.jpg
1    https://images.brickset.com/sets/images/2-8.jpg
2    https://images.brickset.com/sets/images/3-6.jpg
3    https://images.brickset.com/sets/images/4-4.jpg
4                                                NaN
5  https://images.brickset.com/sets/images/078-1.jpg
6  https://images.brickset.com/sets/images/104-1.jpg
7  https://images.brickset.com/sets/images/126-1.jpg
8  https://images.brickset.com/sets/images/157-3.jpg
9  https://images.brickset.com/sets/images/242-4.jpg
```

## 2 1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them.

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18457 entries, 0 to 18456
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   set_id          18457 non-null  object
 1   name            18457 non-null  object
```

```
 2   year            18457 non-null  int64
 3   theme           18457 non-null  object
 4   subtheme        14901 non-null  object
 5   themeGroup      18455 non-null  object
 6   category        18457 non-null  object
 7   pieces          14533 non-null  float64
 8   minifigs        8399 non-null   float64
 9   agerange_min    6787 non-null   float64
 10  US_retailPrice  6982 non-null   float64
 11  bricksetURL     18457 non-null  object
 12  thumbnailURL    17451 non-null  object
 13  imageURL        17451 non-null  object
dtypes: float64(4), int64(1), object(9)
memory usage: 2.0+ MB
```

[5]:
```
df.isnull().sum()
```

[5]:
```
set_id               0
name                 0
year                 0
theme                0
subtheme          3556
themeGroup           2
category             0
pieces            3924
minifigs         10058
agerange_min     11670
US_retailPrice   11475
bricksetURL          0
thumbnailURL      1006
imageURL          1006
dtype: int64
```

[6]:
```
new_df = df['subtheme'].replace(np.nan,0)    #fill null avalue with 0
```

[7]:
```
new_df.isnull().sum()
```

[7]:
```
0
```

[8]:
```
df.dropna(axis=0,inplace=True)
```

[9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2669 entries, 6107 to 18029
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
```

```
 0   set_id          2669 non-null   object
 1   name            2669 non-null   object
 2   year            2669 non-null   int64
 3   theme           2669 non-null   object
 4   subtheme        2669 non-null   object
 5   themeGroup      2669 non-null   object
 6   category        2669 non-null   object
 7   pieces          2669 non-null   float64
 8   minifigs        2669 non-null   float64
 9   agerange_min    2669 non-null   float64
 10  US_retailPrice  2669 non-null   float64
 11  bricksetURL     2669 non-null   object
 12  thumbnailURL    2669 non-null   object
 13  imageURL        2669 non-null   object
dtypes: float64(4), int64(1), object(9)
memory usage: 312.8+ KB
```

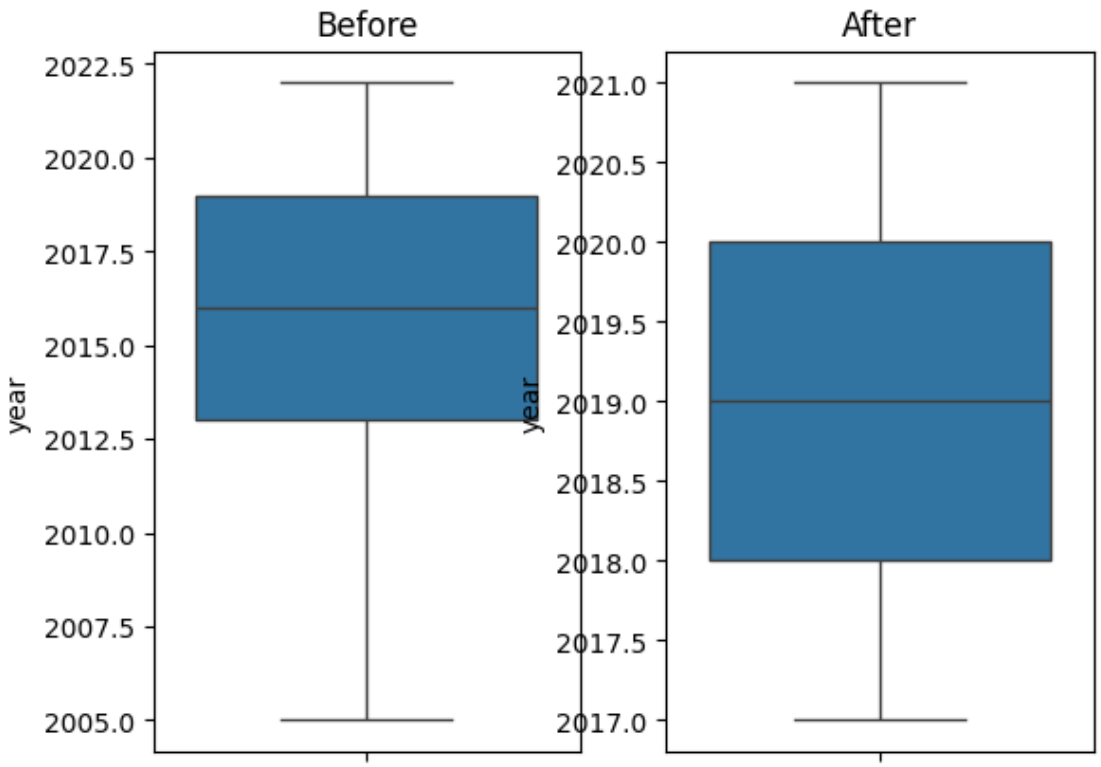
## 3 2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them.

[10]:
```
import seaborn as sb
import matplotlib.pyplot as plt
```

[22]:
```
###warnings.filterwarnings("ignore")
fig,axis = plt.subplots(1,2)
max_val = df.year.quantile(0.95)
min_val = df.year.quantile(0.5)
print("Before Shape", df.shape)
df2 = df[(df['year']>min_val) & (df['year']<max_val)]
print("After Shape", df2.shape)
sb.boxplot(df['year'], orient = 'v', ax=axis[0])
axis[0].title.set_text("Before")
sb.boxplot(df2['year'], orient = 'v', ax=axis[1])
axis[1].title.set_text("After")
plt.show
```

```
Before Shape (2669, 14)
After Shape (1003, 14)
```

[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```




3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution.

[12]:
```
df.head()
```

[12]:

| | set_id | name | year | theme | subtheme | themeGroup | category | pieces | minifigs | agerange_min | US_retailPrice | bricksetURL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6107 | 7236-1 | Police Car | 2005 | City | Police | Modern day | Normal | 59.0 | 1.0 | 5.0 | 5.99 | https://brickset.com/sets/7236-1 |
| 6254 | 10144-1 | Sandcrawler | 2005 | Star Wars | Episode IV | Licensed | Normal | 1669.0 | 11.0 | 12.0 | 139.99 | |
| 6407 | 4962-1 | Baby Zoo | 2006 | Duplo | LEGO Ville | Pre-school | Normal | 18.0 | 1.0 | 2.0 | 9.99 | |
| 6445 | 6209-1 | Slave I | 2006 | Star Wars | Episode V | Licensed | Normal | 537.0 | 5.0 | 8.0 | 49.99 | |
| 6447 | 6211-1 | Imperial Star Destroyer | 2006 | Star Wars | Episode IV | Licensed | Normal | 1367.0 | 9.0 | 9.0 | 99.99 | |

```
6254  https://brickset.com/sets/10144-1
6407   https://brickset.com/sets/4962-1
6445   https://brickset.com/sets/6209-1
6447   https://brickset.com/sets/6211-1

                                              thumbnailURL  \
6107  https://images.brickset.com/sets/small/7236-1.jpg
6254  https://images.brickset.com/sets/small/10144-1…
6407  https://images.brickset.com/sets/small/4962-1.jpg
6445  https://images.brickset.com/sets/small/6209-1.jpg
6447  https://images.brickset.com/sets/small/6211-1.jpg

                                                 imageURL
6107  https://images.brickset.com/sets/images/7236-1…
6254  https://images.brickset.com/sets/images/10144-…
6407  https://images.brickset.com/sets/images/4962-1…
6445  https://images.brickset.com/sets/images/6209-1…
6447  https://images.brickset.com/sets/images/6211-1…
```

[13]:
```
import sklearn
```

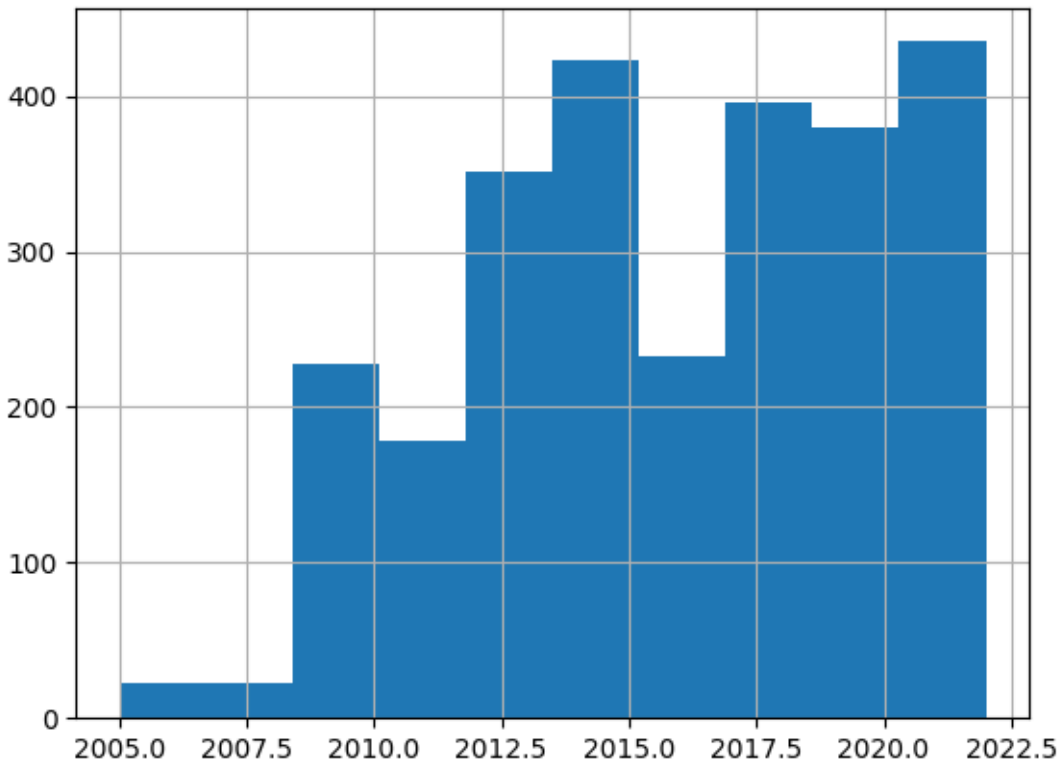
[14]:
```
from sklearn.preprocessing import StandardScaler
```

[15]:
```
scaler = StandardScaler()
x = df[['agerange_min', 'year','US_retailPrice','minifigs']]
scaledf = scaler.fit_transform(x)
print(scaledf)
```

```
[[-0.62750576 -2.73682016 -0.67971696 -0.86203015]
 [ 2.00112431 -2.73682016  1.57016224  2.95855848]
 [-1.7540615  -2.48543244 -0.61255639 -0.86203015]
 …
 [ 1.25008715  1.5367711   2.40966941  1.81238189]
 [ 0.49904998  1.5367711   0.56275364  3.34061735]
 [ 0.49904998  1.5367711   1.23435938  3.72267621]]
```

[16]:
```
df.year.hist()
```

[16]:
```
<Axes: >
```

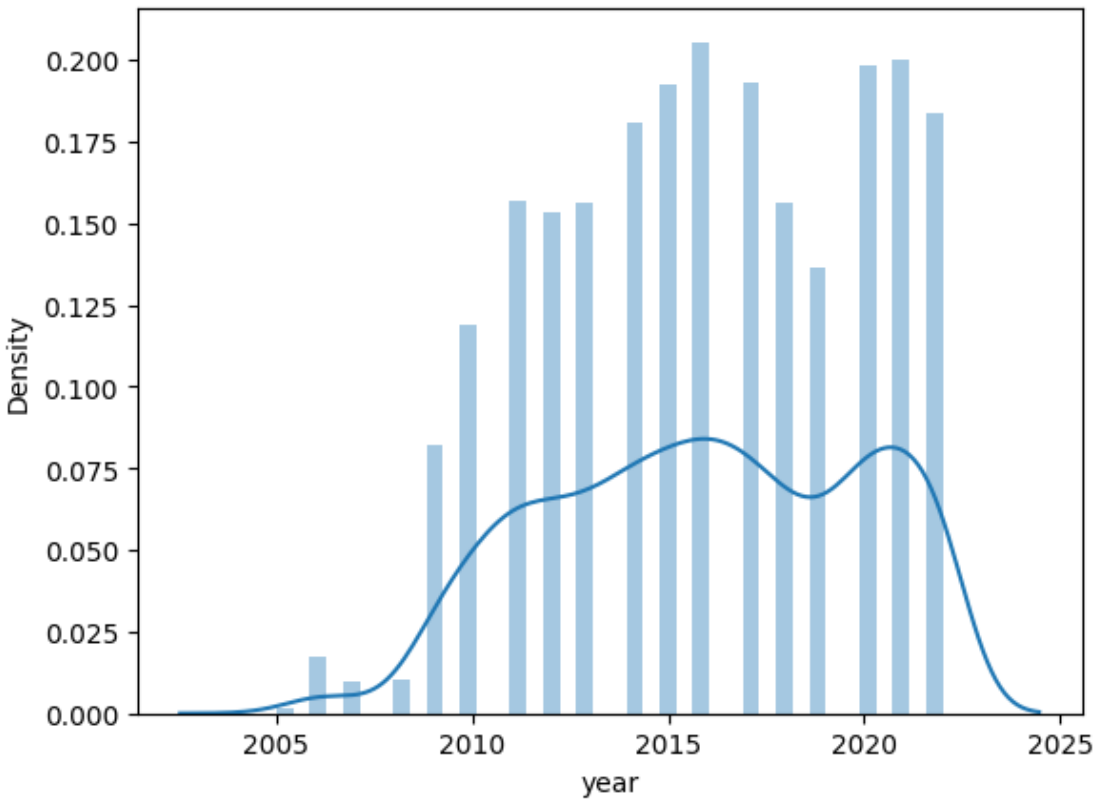



[17]:
```
import scipy.stats as stats
```

[18]:
```
sb.distplot(df['year'], bins=40)
```

```
C:\Users\kumar\AppData\Local\Temp\ipykernel_9060\3404656447.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sb.distplot(df['year'], bins=40)
```

[18]:
```
<Axes: xlabel='year', ylabel='Density'>
```




[19]:
```
df['year'].skew()        #check the skewness
```

[19]:
```
-0.17892940223703785
```

to reduce skewness we have 4 methods. 1. log

[20]:
```
log = np.log(df['year'])
print(log.skew())
```

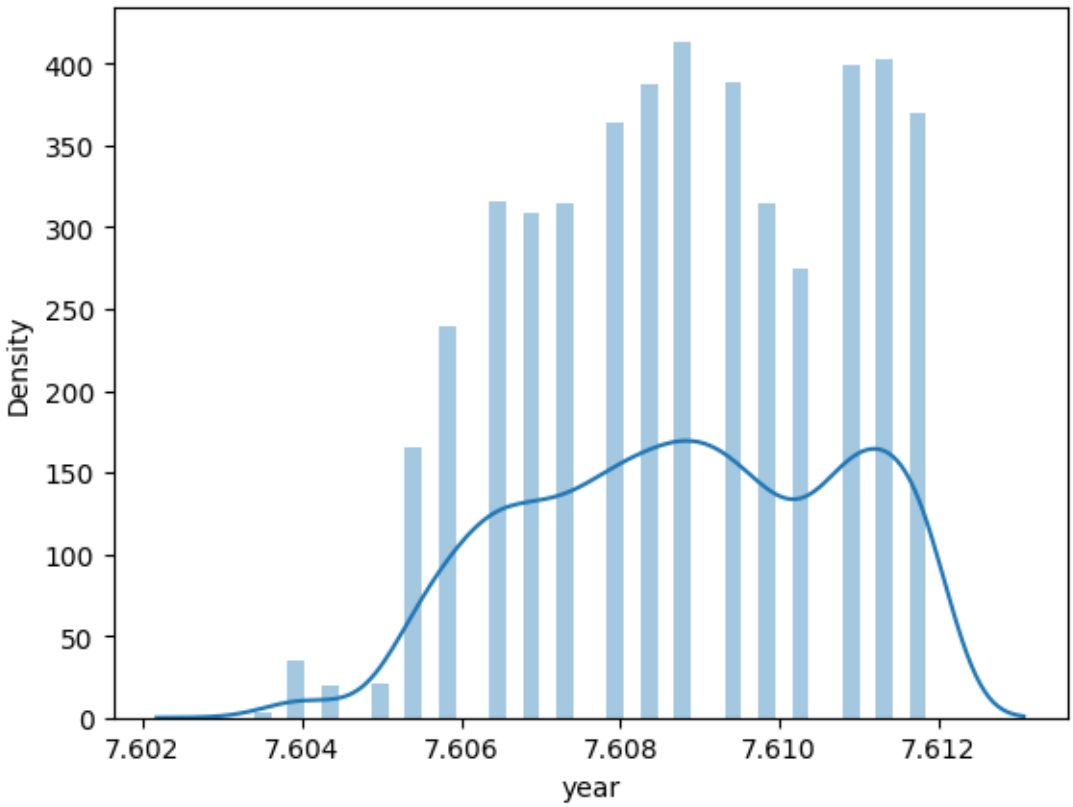
```
-0.18203775462443883
```

[21]:
```
sb.distplot(log, bins=40)
```

```
C:\Users\kumar\AppData\Local\Temp\ipykernel_9060\3503255974.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

```
sb.distplot(log, bins=40)
```

[21]: <Axes: xlabel='year', ylabel='Density'>




[ ]: