reg1-1-rework

November 21, 2023

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import mean_absolute_error as mae
from sklearn.preprocessing import OneHotEncoder
%matplotlib inline
```

[2]:
```
df=pd.read_csv(r'C:
↪\Users\risha\Documents\KRMU\AIML_assigment\datasets\Nutrition__Physical_Activity__and_Obesi
↪csv')
```

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

[3]:
```
   YearStart  YearEnd LocationAbbr LocationDesc  \
0       2011     2011           AL      Alabama
1       2011     2011           AL      Alabama
2       2011     2011           AL      Alabama
3       2011     2011           AL      Alabama
4       2011     2011           AL      Alabama

                                  Datasource                    Class  \
0  Behavioral Risk Factor Surveillance System  Obesity / Weight Status
1  Behavioral Risk Factor Surveillance System  Obesity / Weight Status
2  Behavioral Risk Factor Surveillance System  Obesity / Weight Status
3  Behavioral Risk Factor Surveillance System  Obesity / Weight Status
4  Behavioral Risk Factor Surveillance System  Obesity / Weight Status

                     Topic                                          Question  \
0  Obesity / Weight Status  Percent of adults aged 18 years and older who …
```

```
1  Obesity / Weight Status  Percent of adults aged 18 years and older who …
2  Obesity / Weight Status  Percent of adults aged 18 years and older who …
3  Obesity / Weight Status  Percent of adults aged 18 years and older who …
4  Obesity / Weight Status  Percent of adults aged 18 years and older who …

   Data_Value_Unit Data_Value_Type  …  \
0              NaN           Value  …
1              NaN           Value  …
2              NaN           Value  …
3              NaN           Value  …
4              NaN           Value  …

                              GeoLocation  ClassID TopicID QuestionID  \
0  (32.84057112200048, -86.63186076199969)      OWS    OWS1       Q036
1  (32.84057112200048, -86.63186076199969)      OWS    OWS1       Q036
2  (32.84057112200048, -86.63186076199969)      OWS    OWS1       Q036
3  (32.84057112200048, -86.63186076199969)      OWS    OWS1       Q036
4  (32.84057112200048, -86.63186076199969)      OWS    OWS1       Q036

   DataValueTypeID  LocationID  StratificationCategory1  \
0            VALUE           1                    Total
1            VALUE           1                   Gender
2            VALUE           1                   Gender
3            VALUE           1                Education
4            VALUE           1                Education

         Stratification1 StratificationCategoryId1 StratificationID1
0                  Total                      OVR           OVERALL
1                   Male                      GEN              MALE
2                 Female                      GEN            FEMALE
3  Less than high school                      EDU             EDUHS
4   High school graduate                      EDU         EDUHSGRAD

[5 rows x 33 columns]
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53392 entries, 0 to 53391
Data columns (total 33 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   YearStart                   53392 non-null  int64
 1   YearEnd                     53392 non-null  int64
 2   LocationAbbr                53392 non-null  object
 3   LocationDesc                53392 non-null  object
 4   Datasource                  53392 non-null  object
```

```
 5   Class                        53392 non-null  object
 6   Topic                        53392 non-null  object
 7   Question                     53392 non-null  object
 8   Data_Value_Unit              0 non-null      float64
 9   Data_Value_Type              53392 non-null  object
 10  Data_Value                   48346 non-null  float64
 11  Data_Value_Alt               48346 non-null  float64
 12  Data_Value_Footnote_Symbol   5046 non-null   object
 13  Data_Value_Footnote          5046 non-null   object
 14  Low_Confidence_Limit         48346 non-null  float64
 15  High_Confidence_Limit        48346 non-null  float64
 16  Sample_Size                  48346 non-null  float64
 17  Total                        1907 non-null   object
 18  Age(years)                   11438 non-null  object
 19  Education                    7628 non-null   object
 20  Gender                       3814 non-null   object
 21  Income                       13349 non-null  object
 22  Race/Ethnicity               15256 non-null  object
 23  GeoLocation                  52384 non-null  object
 24  ClassID                      53392 non-null  object
 25  TopicID                      53392 non-null  object
 26  QuestionID                   53392 non-null  object
 27  DataValueTypeID              53392 non-null  object
 28  LocationID                   53392 non-null  int64
 29  StratificationCategory1      53392 non-null  object
 30  Stratification1              53392 non-null  object
 31  StratificationCategoryId1    53392 non-null  object
 32  StratificationID1            53392 non-null  object
dtypes: float64(6), int64(3), object(24)
memory usage: 13.4+ MB
```

[5]:
```
df.shape
```

[5]:
```
(53392, 33)
```

[6]:
```
df.columns
```

[6]:
```
Index(['YearStart', 'YearEnd', 'LocationAbbr', 'LocationDesc', 'Datasource',
       'Class', 'Topic', 'Question', 'Data_Value_Unit', 'Data_Value_Type',
       'Data_Value', 'Data_Value_Alt', 'Data_Value_Footnote_Symbol',
       'Data_Value_Footnote', 'Low_Confidence_Limit', 'High_Confidence_Limit',
       'Sample_Size', 'Total', 'Age(years)', 'Education', 'Gender', 'Income',
       'Race/Ethnicity', 'GeoLocation', 'ClassID', 'TopicID', 'QuestionID',
       'DataValueTypeID', 'LocationID', 'StratificationCategory1',
       'Stratification1', 'StratificationCategoryId1', 'StratificationID1'],
      dtype='object')
```

[7]:
```
df.isna().sum()
```

[7]:
```
YearStart                          0
YearEnd                            0
LocationAbbr                       0
LocationDesc                       0
Datasource                         0
Class                              0
Topic                              0
Question                           0
Data_Value_Unit                53392
Data_Value_Type                    0
Data_Value                      5046
Data_Value_Alt                  5046
Data_Value_Footnote_Symbol     48346
Data_Value_Footnote            48346
Low_Confidence_Limit            5046
High_Confidence_Limit           5046
Sample_Size                     5046
Total                          51485
Age(years)                     41954
Education                      45764
Gender                         49578
Income                         40043
Race/Ethnicity                 38136
GeoLocation                     1008
ClassID                            0
TopicID                            0
QuestionID                         0
DataValueTypeID                    0
LocationID                         0
StratificationCategory1            0
Stratification1                    0
StratificationCategoryId1          0
StratificationID1                  0
dtype: int64
```

[8]:
```
df=df.drop(['Data_Value',␣
 ↪'Income','Data_Value_Unit','Data_Value_Footnote_Symbol','Total','Data_Value_Footnote'],␣
 ↪axis=1)
```

[9]:
```
df.columns
```

[9]:
```
Index(['YearStart', 'YearEnd', 'LocationAbbr', 'LocationDesc', 'Datasource',
       'Class', 'Topic', 'Question', 'Data_Value_Type', 'Data_Value_Alt',
       'Low_Confidence_Limit', 'High_Confidence_Limit', 'Sample_Size',
       'Age(years)', 'Education', 'Gender', 'Race/Ethnicity', 'GeoLocation',
```

```
       'ClassID', 'TopicID', 'QuestionID', 'DataValueTypeID', 'LocationID',
       'StratificationCategory1', 'Stratification1',
       'StratificationCategoryId1', 'StratificationID1'],
      dtype='object')
```

[10]:
```
df.Data_Value_Type.value_counts()
```

[10]:
```
Data_Value_Type
Value    53392
Name: count, dtype: int64
```

[11]:
```
df=df.drop(['Data_Value_Type','Topic', 'Question'], axis=1)
```

[12]:
```
df.isna().sum()
```

[12]:
```
YearStart                       0
YearEnd                         0
LocationAbbr                    0
LocationDesc                    0
Datasource                      0
Class                           0
Data_Value_Alt               5046
Low_Confidence_Limit         5046
High_Confidence_Limit        5046
Sample_Size                  5046
Age(years)                  41954
Education                   45764
Gender                      49578
Race/Ethnicity              38136
GeoLocation                  1008
ClassID                         0
TopicID                         0
QuestionID                      0
DataValueTypeID                 0
LocationID                      0
StratificationCategory1         0
Stratification1                 0
StratificationCategoryId1       0
StratificationID1               0
dtype: int64
```

[13]:
```
df.Gender.value_counts()
```

[13]:
```
Gender
Male      1907
Female    1907
Name: count, dtype: int64
```

[14]:
```
df['Race/Ethnicity'].value_counts()
```

[14]:
```
Race/Ethnicity
Non-Hispanic White                1907
Non-Hispanic Black                1907
Hispanic                          1907
Asian                             1907
Hawaiian/Pacific Islander         1907
American Indian/Alaska Native     1907
2 or more races                   1907
Other                             1907
Name: count, dtype: int64
```

[15]:
```
pd.unique(df.YearStart)
```

[15]:
```
array([2011, 2012, 2014, 2013, 2015, 2016], dtype=int64)
```

[16]:
```
pd.unique(df.YearEnd)
```

[16]:
```
array([2011, 2012, 2014, 2013, 2015, 2016], dtype=int64)
```

[17]:
```
df[['YearStart','YearEnd']]
```

[17]:

	YearStart	YearEnd
0	2011	2011
1	2011	2011
2	2011	2011
3	2011	2011
4	2011	2011
…	…	…
53387	2016	2016
53388	2016	2016
53389	2016	2016
53390	2016	2016
53391	2016	2016

[53392 rows x 2 columns]

[18]:
```
df=df.dropna(subset=['Education'])
```

[19]:
```
print(df.duplicated().sum())
```

```
0
```

[20]:
```
def remove_outliers(df):
    for col in df.select_dtypes(include=np.number).columns:
        q1 = df[col].quantile(0.25)
```

```
        q3 = df[col].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - (1.5 * iqr)
        upper_bound = q3 + (1.5 * iqr)
        df.drop(df[(df[col] < lower_bound) | (df[col] > upper_bound)].index,␣
↪inplace=True)
```

[21]:
```
remove_outliers(df)
df
```

[21]:

	YearStart	YearEnd	LocationAbbr	LocationDesc	\
3	2011	2011	AL	Alabama	
4	2011	2011	AL	Alabama	
5	2011	2011	AL	Alabama	
6	2011	2011	AL	Alabama	
31	2011	2011	AL	Alabama	
…	…	…	…	…	
53343	2016	2016	PR	Puerto Rico	
53367	2016	2016	VI	Virgin Islands	
53368	2016	2016	VI	Virgin Islands	
53369	2016	2016	VI	Virgin Islands	
53370	2016	2016	VI	Virgin Islands	

	Datasource	Class	\
3	Behavioral Risk Factor Surveillance System	Obesity / Weight Status	
4	Behavioral Risk Factor Surveillance System	Obesity / Weight Status	
5	Behavioral Risk Factor Surveillance System	Obesity / Weight Status	
6	Behavioral Risk Factor Surveillance System	Obesity / Weight Status	
31	Behavioral Risk Factor Surveillance System	Obesity / Weight Status	
…	…	…	
53343	Behavioral Risk Factor Surveillance System	Physical Activity	
53367	Behavioral Risk Factor Surveillance System	Physical Activity	
53368	Behavioral Risk Factor Surveillance System	Physical Activity	
53369	Behavioral Risk Factor Surveillance System	Physical Activity	
53370	Behavioral Risk Factor Surveillance System	Physical Activity	

	Data_Value_Alt	Low_Confidence_Limit	High_Confidence_Limit	\
3	33.6	29.9	37.6	
4	32.8	30.2	35.6	
5	33.8	31.0	36.8	
6	26.4	23.7	29.3	
31	33.2	29.2	37.5	
…	…	…	…	
53343	33.9	31.1	36.8	
53367	29.2	21.2	38.6	
53368	26.9	21.1	33.5	
53369	31.2	22.2	41.8	

```
53370           16.4                  11.7                  22.5

       Sample_Size  …                              GeoLocation ClassID  \
3           1153.0  …  (32.84057112200048, -86.63186076199969)     OWS
4           2402.0  …  (32.84057112200048, -86.63186076199969)     OWS
5           1925.0  …  (32.84057112200048, -86.63186076199969)     OWS
6           1812.0  …  (32.84057112200048, -86.63186076199969)     OWS
31          1153.0  …  (32.84057112200048, -86.63186076199969)     OWS
…            …  …                                          …     …
53343       1735.0  …                (18.220833, -66.590149)      PA
53367        238.0  …                (18.335765, -64.896335)      PA
53368        394.0  …                (18.335765, -64.896335)      PA
53369        230.0  …                (18.335765, -64.896335)      PA
53370        378.0  …                (18.335765, -64.896335)      PA

      TopicID QuestionID DataValueTypeID LocationID StratificationCategory1  \
3        OWS1       Q036           VALUE          1                Education
4        OWS1       Q036           VALUE          1                Education
5        OWS1       Q036           VALUE          1                Education
6        OWS1       Q036           VALUE          1                Education
31       OWS1       Q037           VALUE          1                Education
…       …        …              …       …                     …
53343     PA1       Q047           VALUE         72                Education
53367     PA1       Q047           VALUE         78                Education
53368     PA1       Q047           VALUE         78                Education
53369     PA1       Q047           VALUE         78                Education
53370     PA1       Q047           VALUE         78                Education

                     Stratification1 StratificationCategoryId1  \
3                Less than high school                      EDU
4                 High school graduate                      EDU
5      Some college or technical school                      EDU
6                    College graduate                      EDU
31               Less than high school                      EDU
…                                  …                      …
53343                College graduate                      EDU
53367            Less than high school                      EDU
53368             High school graduate                      EDU
53369  Some college or technical school                      EDU
53370                College graduate                      EDU

       StratificationID1
3                  EDUHS
4              EDUHSGRAD
5               EDUCOTEC
6              EDUCOGRAD
31                 EDUHS
```

```
…                    …
53343           EDUCOGRAD
53367                EDUHS
53368           EDUHSGRAD
53369            EDUCOTEC
53370           EDUCOGRAD

[6940 rows x 24 columns]
```

[22]:
```
def standardize_text(df):
    for col in df.select_dtypes(include=['object']).columns:
        df[col] = df[col].str.lower()
```

[23]:
```
standardize_text(df)
```

[24]:
```
df.isna().sum()
```

[24]:
```
YearStart                     0
YearEnd                       0
LocationAbbr                  0
LocationDesc                  0
Datasource                    0
Class                         0
Data_Value_Alt                0
Low_Confidence_Limit          0
High_Confidence_Limit         0
Sample_Size                   0
Age(years)                 6940
Education                     0
Gender                     6940
Race/Ethnicity             6940
GeoLocation                   0
ClassID                       0
TopicID                       0
QuestionID                    0
DataValueTypeID               0
LocationID                    0
StratificationCategory1       0
Stratification1               0
StratificationCategoryId1     0
StratificationID1             0
dtype: int64
```

[25]:
```
df['Data_Value_Alt']= df['Data_Value_Alt'].fillna(df['Data_Value_Alt'].mean())
```

[26]:
```
df['Low_Confidence_Limit']=df['Low_Confidence_Limit'].
 ↪fillna(df['Low_Confidence_Limit'].median())
```

[27]:
```
df['High_Confidence_Limit']=df['High_Confidence_Limit'].
 ↪fillna(df['High_Confidence_Limit'].median())
```

[28]:
```
df['Sample_Size']=df['Sample_Size'].fillna(df['Sample_Size'].median())
```

[29]:
```
num_col= df.select_dtypes(include=['int','float']).columns
num= df[num_col]
```

[30]:
```
num
```

[30]:
```
       YearStart  YearEnd  Data_Value_Alt  Low_Confidence_Limit  \
3           2011     2011            33.6                  29.9
4           2011     2011            32.8                  30.2
5           2011     2011            33.8                  31.0
6           2011     2011            26.4                  23.7
31          2011     2011            33.2                  29.2
…            …        …               …                     …
53343       2016     2016            33.9                  31.1
53367       2016     2016            29.2                  21.2
53368       2016     2016            26.9                  21.1
53369       2016     2016            31.2                  22.2
53370       2016     2016            16.4                  11.7

       High_Confidence_Limit  Sample_Size  LocationID
3                       37.6       1153.0           1
4                       35.6       2402.0           1
5                       36.8       1925.0           1
6                       29.3       1812.0           1
31                      37.5       1153.0           1
…                        …          …               …
53343                   36.8       1735.0          72
53367                   38.6        238.0          78
53368                   33.5        394.0          78
53369                   41.8        230.0          78
53370                   22.5        378.0          78

[6940 rows x 7 columns]
```

[31]:
```
num.isna().sum()
```

[31]:
```
YearStart               0
YearEnd                 0
Data_Value_Alt          0
Low_Confidence_Limit    0
High_Confidence_Limit   0
Sample_Size             0
LocationID              0
```

```
dtype: int64
```

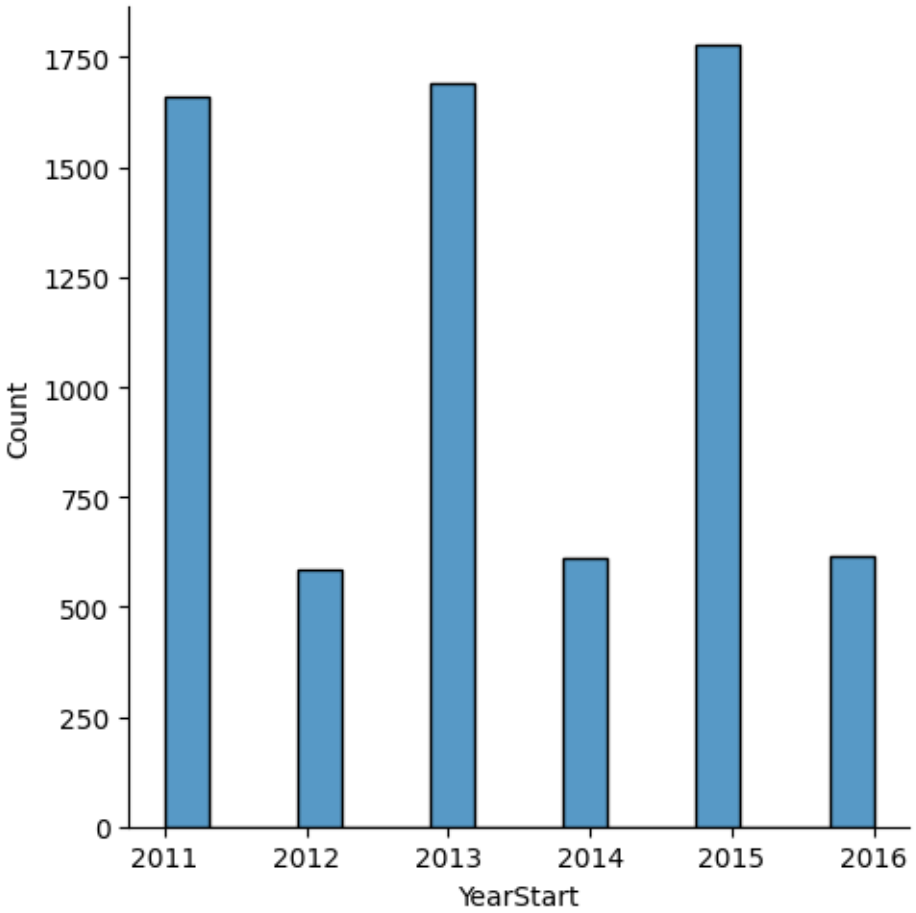
[32]:
```
cat_col= df.select_dtypes(include=['object']).columns
cat= df[cat_col]
```

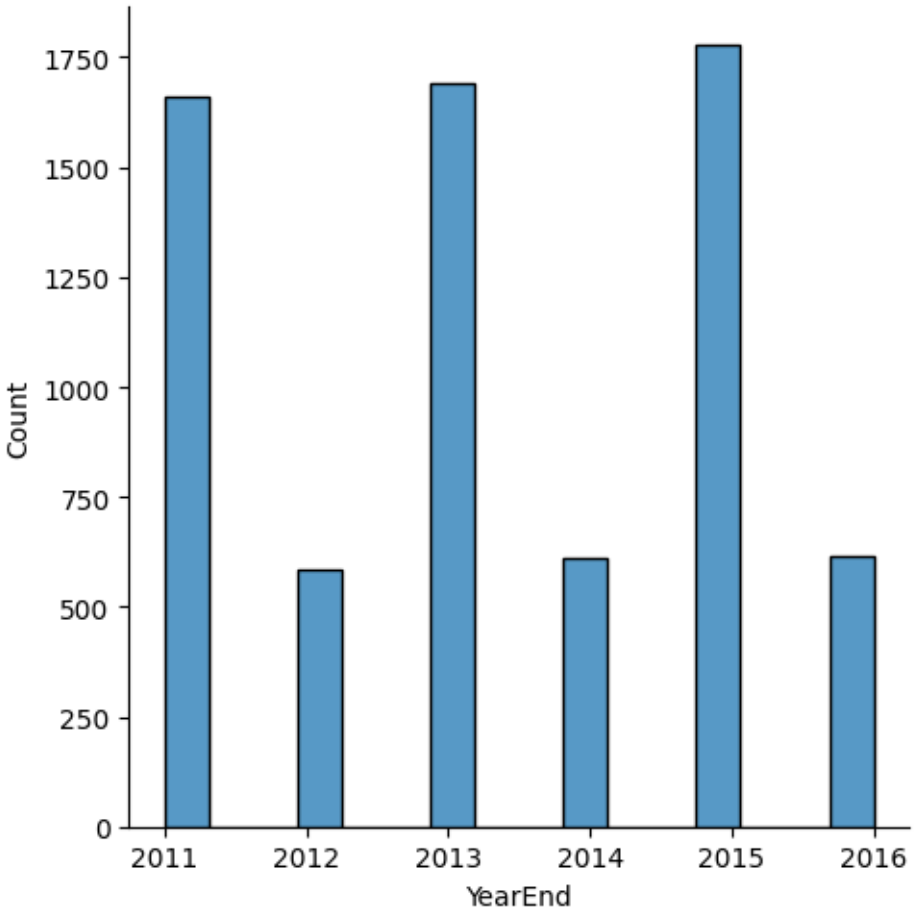
[33]:
```
cat.columns
```

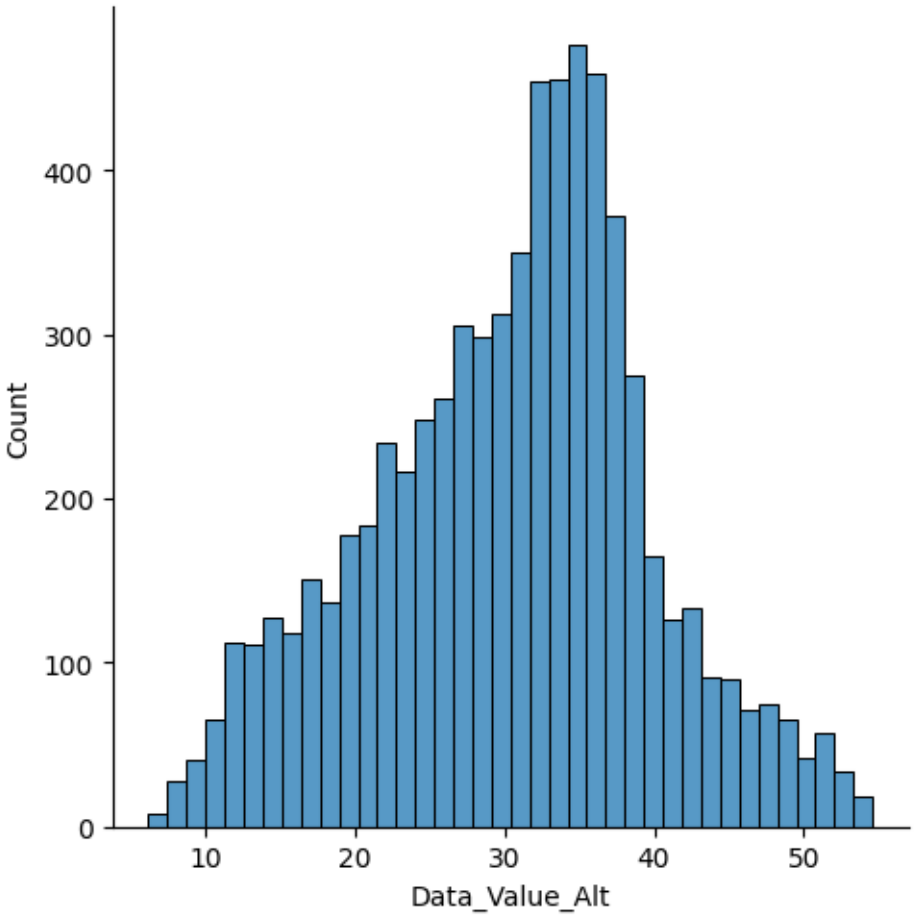
[33]:
```
Index(['LocationAbbr', 'LocationDesc', 'Datasource', 'Class', 'Age(years)',
       'Education', 'Gender', 'Race/Ethnicity', 'GeoLocation', 'ClassID',
       'TopicID', 'QuestionID', 'DataValueTypeID', 'StratificationCategory1',
       'Stratification1', 'StratificationCategoryId1', 'StratificationID1'],
      dtype='object')
```

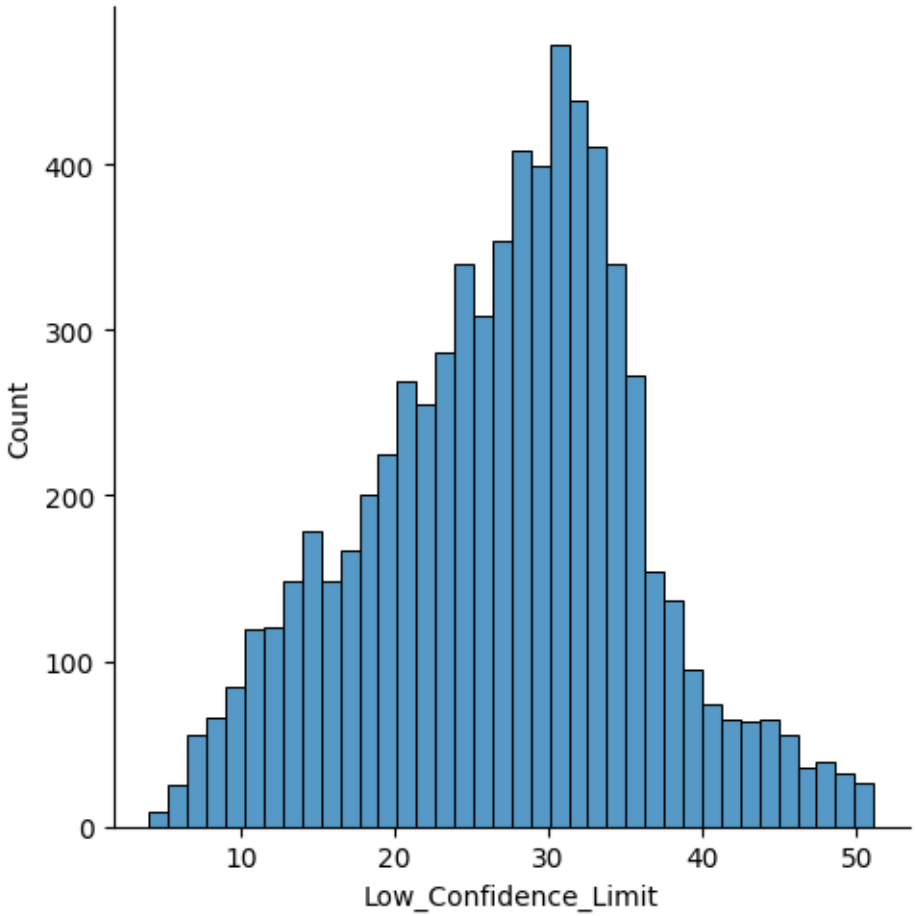
[34]:
```
ed_map = {'less than high school': 0, 'high school graduate':1, 'some college␣
 ↪or technical school':2,'college graduate':3}
ed_map
df['Education'] = df['Education'].map(ed_map)
```

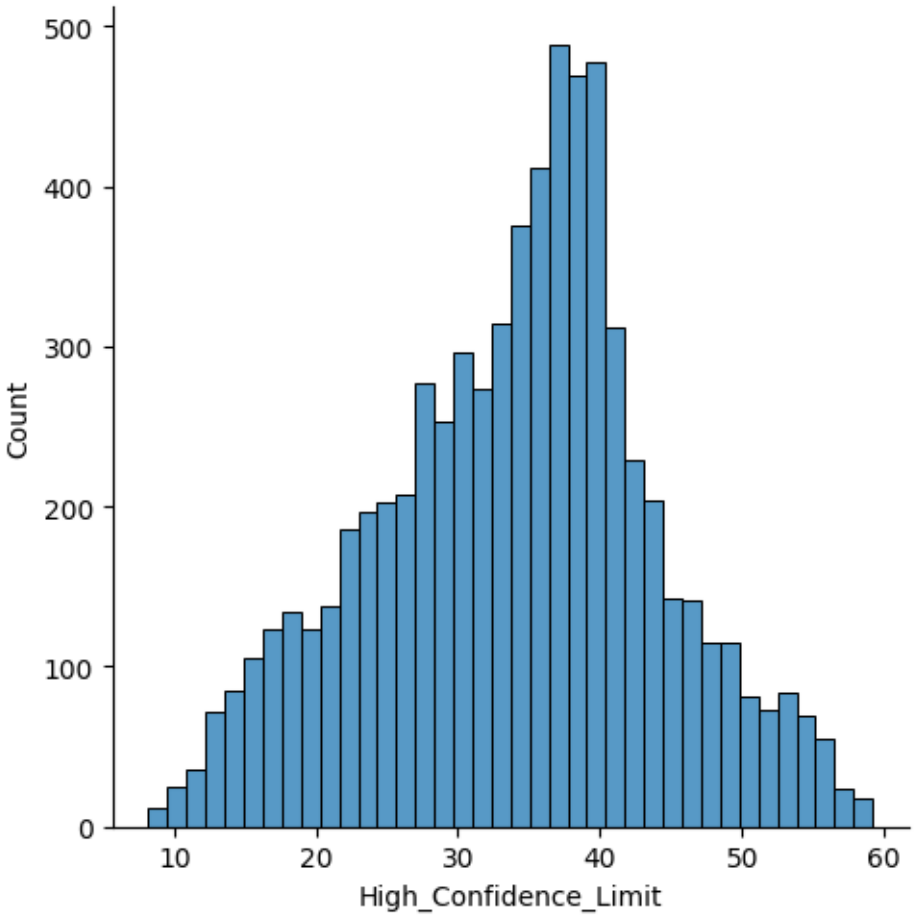
[35]:
```
for col in num.columns:
    sns.displot(num[col])
    plt.show()
```

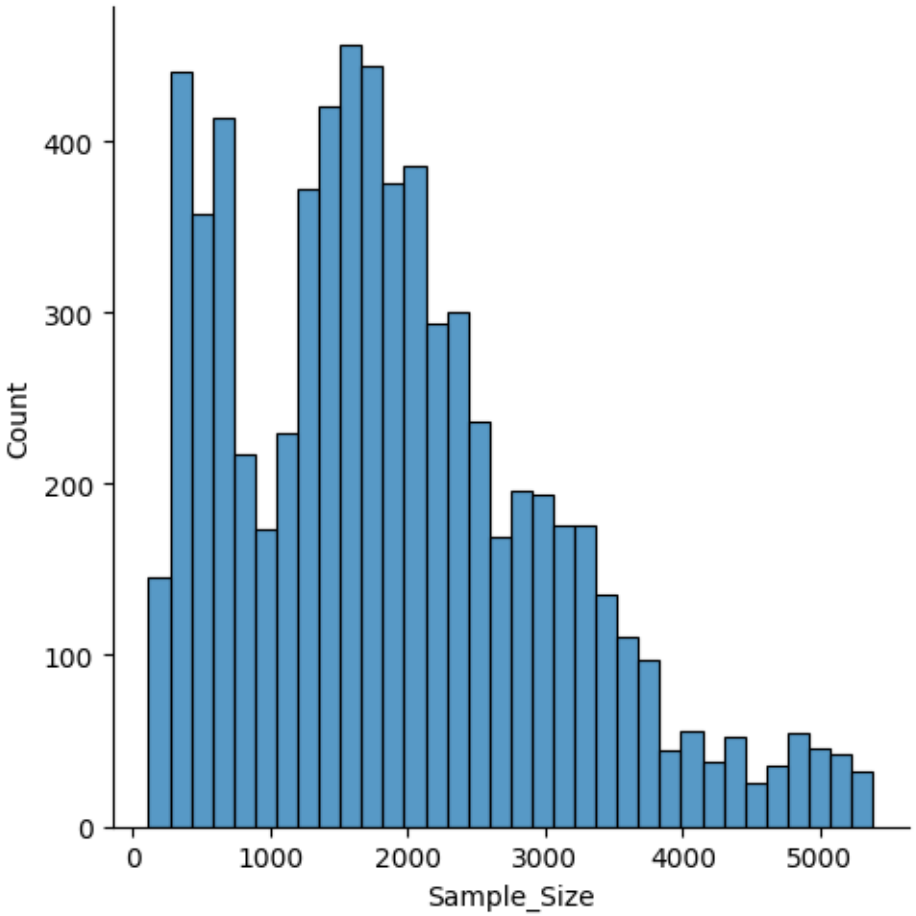

1750
1500
1250
1000
750
500
250
0
Count
2011
2012
2013
2014
2015
2016
YearStart

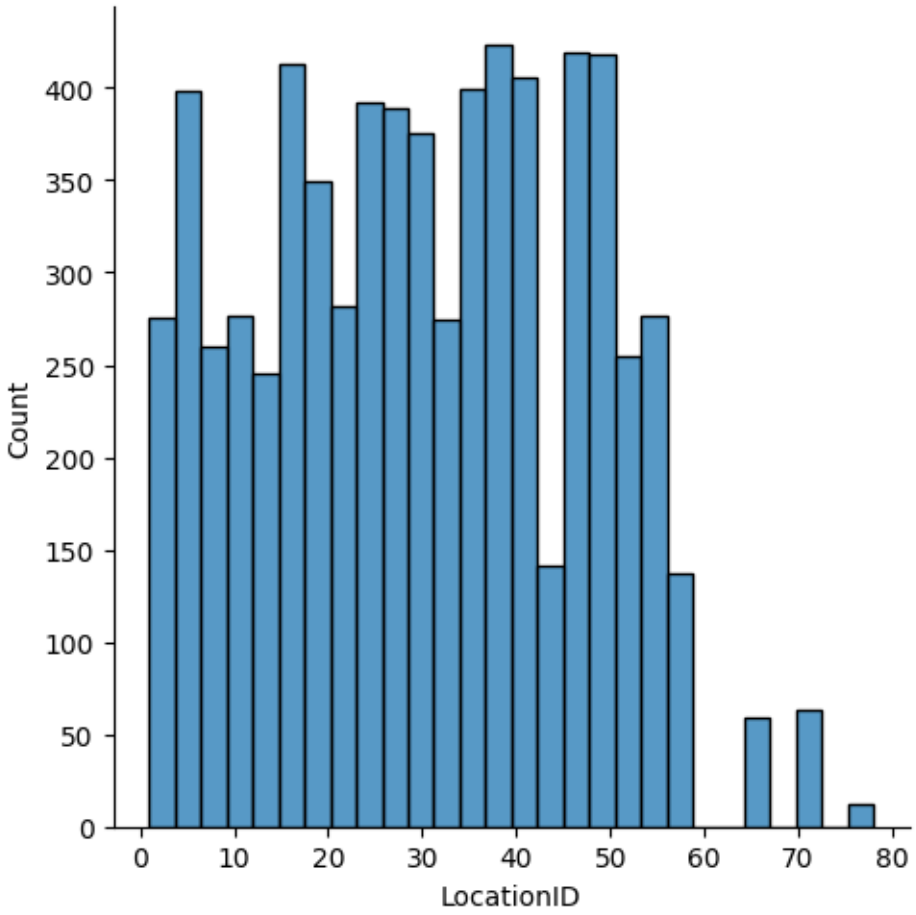

1750
1500
1250
1000
750
500
250
0
Count
2011
2012
2013
2014
2015
2016
YearEnd


Count
400
300
200
100
0
10
20
30
40
50
Data_Value_Alt


Count
0
100
200
300
400
10
20
30
40
50
Low_Confidence_Limit


500
400
300
200
100
0
Count
10
20
30
40
50
60
High_Confidence_Limit

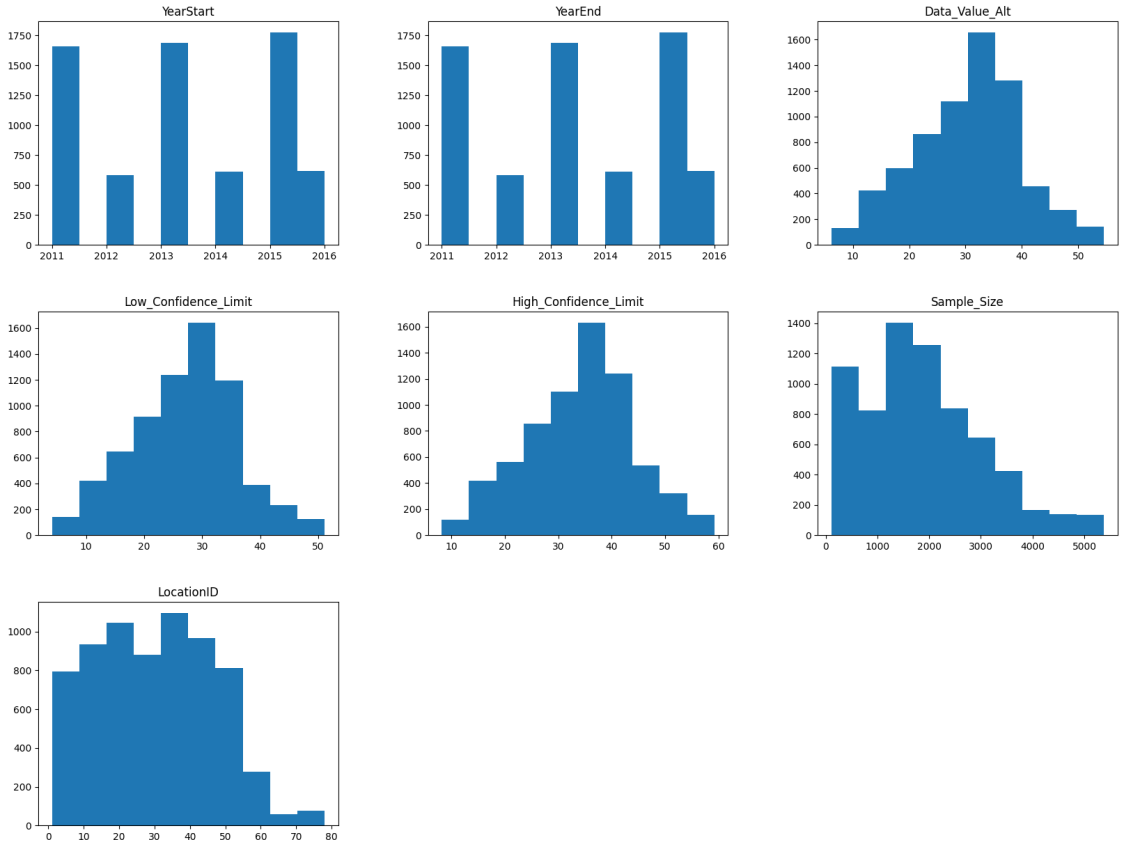

Count
400
300
200
100
0
0
1000
2000
3000
4000
5000
Sample_Size



[36]:
```
num.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6940 entries, 3 to 53370
Data columns (total 7 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   YearStart              6940 non-null   int64
 1   YearEnd                6940 non-null   int64
 2   Data_Value_Alt         6940 non-null   float64
 3   Low_Confidence_Limit   6940 non-null   float64
 4   High_Confidence_Limit  6940 non-null   float64
 5   Sample_Size            6940 non-null   float64
 6   LocationID             6940 non-null   int64
dtypes: float64(4), int64(3)
memory usage: 433.8 KB
```

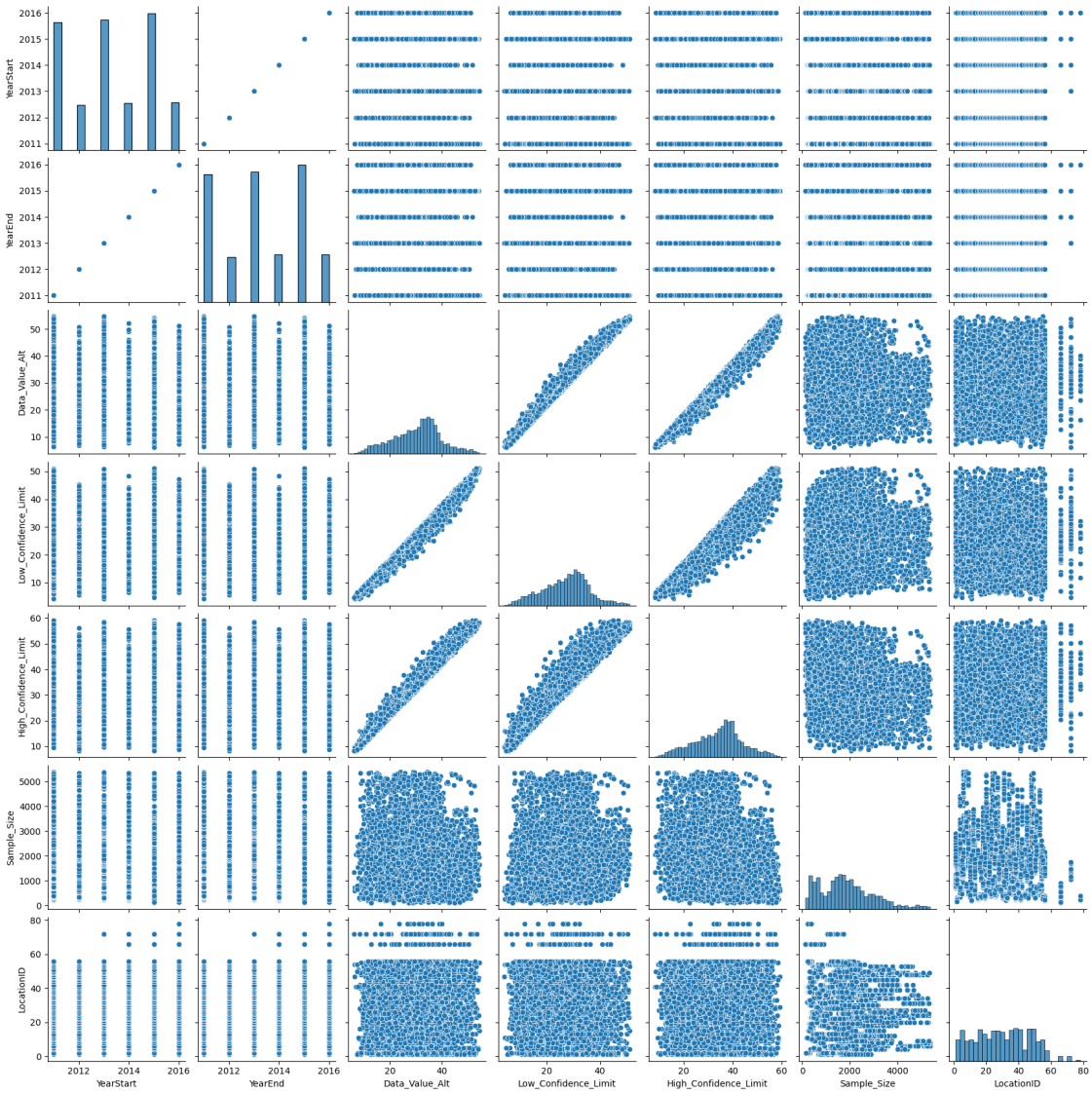
[37]:
```
num.hist(figsize=(20,15), grid=False)
plt.show()
```



0.0.1 Plot relationships between Numerical variables

[38]:
```
sns.pairplot(num)
```

[38]:
```
<seaborn.axisgrid.PairGrid at 0x1ee25c192b0>
```



[39]:
```
num.corr()
```

[39]:

	YearStart	YearEnd	Data_Value_Alt	\
YearStart	1.000000	1.000000	0.019509	
YearEnd	1.000000	1.000000	0.019509	
Data_Value_Alt	0.019509	0.019509	1.000000	
Low_Confidence_Limit	0.009062	0.009062	0.986612	
High_Confidence_Limit	0.029935	0.029935	0.984898	
Sample_Size	-0.078040	-0.078040	-0.124962	
LocationID	0.046434	0.046434	-0.000214	

	Low_Confidence_Limit	High_Confidence_Limit	\
YearStart	0.009062	0.029935	

```
YearEnd                          0.009062              0.029935
Data_Value_Alt                   0.986612              0.984898
Low_Confidence_Limit             1.000000              0.943704
High_Confidence_Limit            0.943704              1.000000
Sample_Size                      0.002910             -0.256214
LocationID                       0.004402             -0.005112

                        Sample_Size  LocationID
YearStart                 -0.078040    0.046434
YearEnd                   -0.078040    0.046434
Data_Value_Alt            -0.124962   -0.000214
Low_Confidence_Limit       0.002910    0.004402
High_Confidence_Limit     -0.256214   -0.005112
Sample_Size                1.000000    0.013820
LocationID                 0.013820    1.000000
```

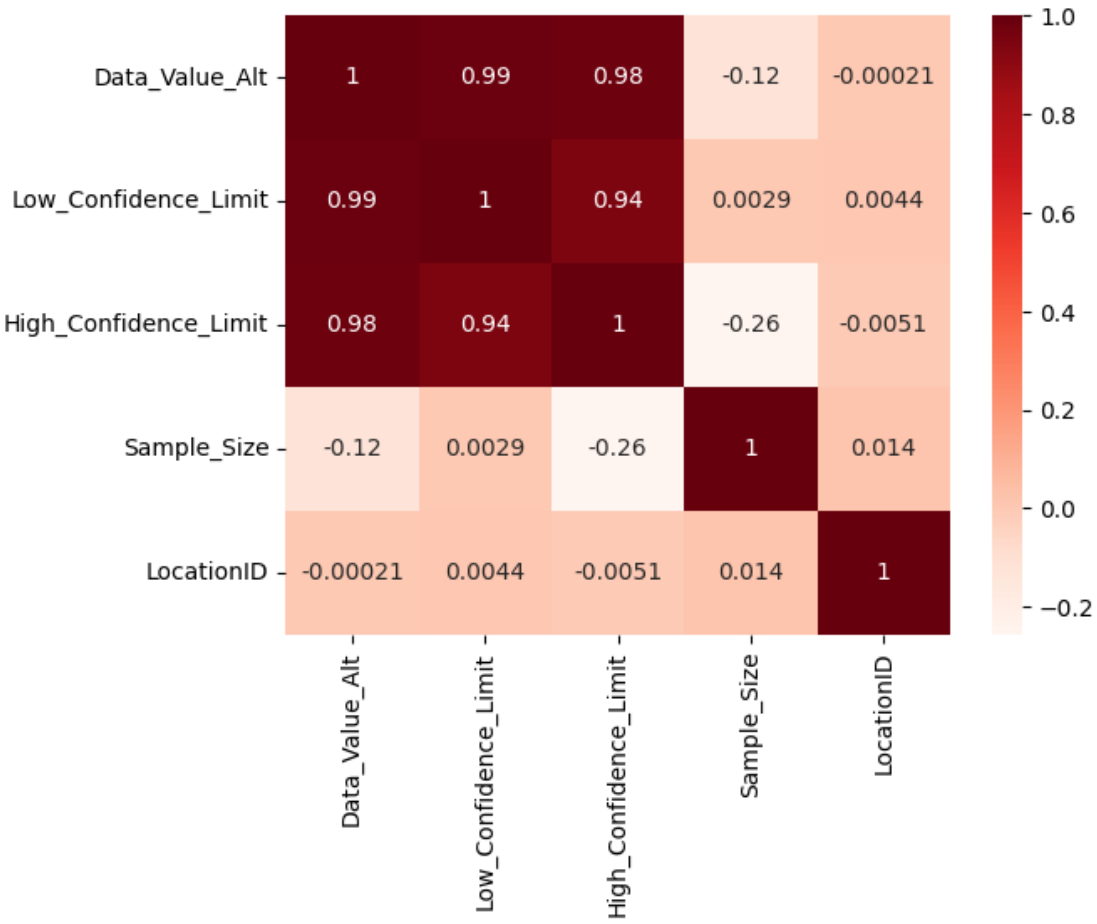
[40]:
```
num.columns
```

[40]:
```
Index(['YearStart', 'YearEnd', 'Data_Value_Alt', 'Low_Confidence_Limit',
       'High_Confidence_Limit', 'Sample_Size', 'LocationID'],
      dtype='object')
```

[41]:
```
n= num[['Data_Value_Alt', 'Low_Confidence_Limit',
       'High_Confidence_Limit', 'Sample_Size', 'LocationID']]
```

[42]:
```
corr = n.corr()
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns,
        annot=True, cmap='Reds')
```

[42]:
```
<Axes: >
```



[43]:
```
X = df[['YearStart','Education', 'LocationID']]
y = num['Data_Value_Alt']
```

[44]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
X_train = pd.DataFrame(X_train)
X_test = pd.DataFrame(X_test)
X_train.head(3)
```

[44]:

	YearStart	Education	LocationID
14967	2011	2	26
46600	2015	1	19
24003	2013	0	38

[45]:
```
std_scaler=StandardScaler().fit(X_train)

X_train_scaled=std_scaler.transform(X_train)
```

[46]:
```
X_test_scaled=std_scaler.transform(X_test)
```

[47]:
```
print(X_train_scaled)
print("--------")
print(X_test_scaled)
```

```
[[-1.38033768  0.50342707 -0.2404683 ]
 [ 1.01134505 -0.39570848 -0.66584102]
 [-0.18449632 -1.29484403  0.48874207]
 …
 [ 1.01134505  0.50342707  1.15718492]
 [ 0.41342437 -1.29484403  1.46102257]
 [-0.18449632  0.50342707  1.21795245]]
--------
[[-0.18449632  1.40256262 -1.03044621]
 [-0.782417   -0.39570848  0.30643948]
 [ 0.41342437 -0.39570848  2.55483814]
 …
 [-1.38033768  0.50342707 -1.21274881]
 [-1.38033768  0.50342707  0.36720701]
 [ 1.01134505 -1.29484403  0.30643948]]
```

[48]:
```
X_train_const_scaled = sm.add_constant(X_train_scaled)

model = sm.OLS(y_train, X_train_const_scaled).fit()

predictions_train = model.predict(X_train_const_scaled)

X_test_const_scaled = sm.add_constant(X_test_scaled)

predictions_test = model.predict(X_test_const_scaled)

print_model = model.summary()
print(print_model)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Data_Value_Alt   R-squared:                       0.039
Model:                            OLS   Adj. R-squared:                  0.038
Method:                 Least Squares   F-statistic:                     65.43
Date:                Tue, 21 Nov 2023   Prob (F-statistic):           1.82e-41
Time:                        14:53:25   Log-Likelihood:                -17674.
No. Observations:                4858   AIC:                         3.536e+04
Df Residuals:                    4854   BIC:                         3.538e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         30.4204      0.132    230.356      0.000      30.162      30.679
```

```
x1             0.2043      0.132      1.546      0.122      -0.055       0.463
x2            -1.8412      0.132    -13.940      0.000      -2.100      -1.582
x3            -0.0300      0.132     -0.227      0.820      -0.289       0.229
==============================================================================
Omnibus:                       16.017   Durbin-Watson:                   2.048
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               15.586
Skew:                          -0.119   Prob(JB):                     0.000413
Kurtosis:                       2.856   Cond. No.                         1.04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[49]:
```
predictions_test
```

[49]:
```
array([27.831358  , 30.97994839, 31.15666981, …, 29.24800755,
       29.20053708, 33.00180947])
```

[50]:
```
model.rsquared_adj
```

[50]:
```
0.03827150254827483
```

0.0.2 model fitting

[51]:
```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

[52]:
```
model.coef_
```

[52]:
```
array([ 0.12213778, -1.65544775, -0.00182579])
```

[53]:
```
model.score(X_test_scaled,y_test)
```

```
c:\Users\risha\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py:465: UserWarning: X does not have valid feature names,
but LinearRegression was fitted with feature names
  warnings.warn(
```

[53]:
```
-709.8894397972846
```

[54]:
```
y_test
```

[54]:
```
7316     15.2
21849    28.2
27055    38.1
```

```
32600    43.1
37176    21.4
         …
21204    36.3
34761    36.2
4780     37.8
22355    23.6
36923    42.9
Name: Data_Value_Alt, Length: 2082, dtype: float64
```

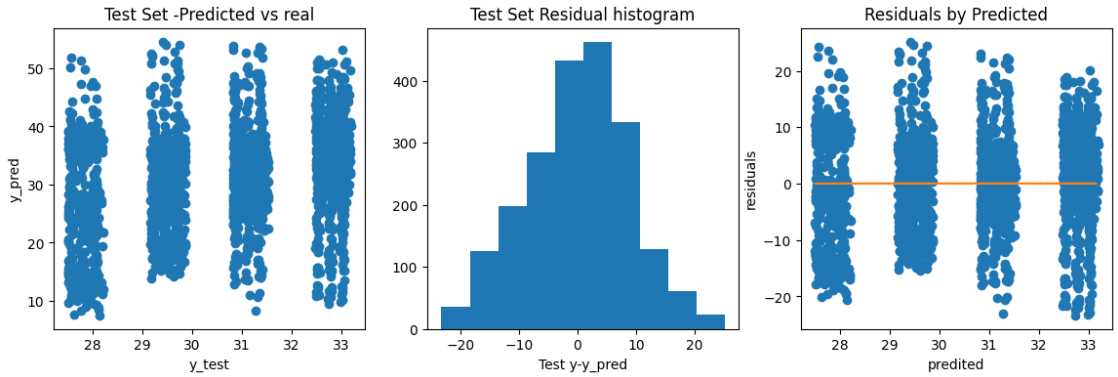
[55]:
```
y_pred
```

[55]:
```
array([27.831358  , 30.97994839, 31.15666981, …, 29.24800755,
       29.20053708, 33.00180947])
```

[56]:
```
result=pd.DataFrame({"y_test":y_test,"y_pred":y_pred})
```

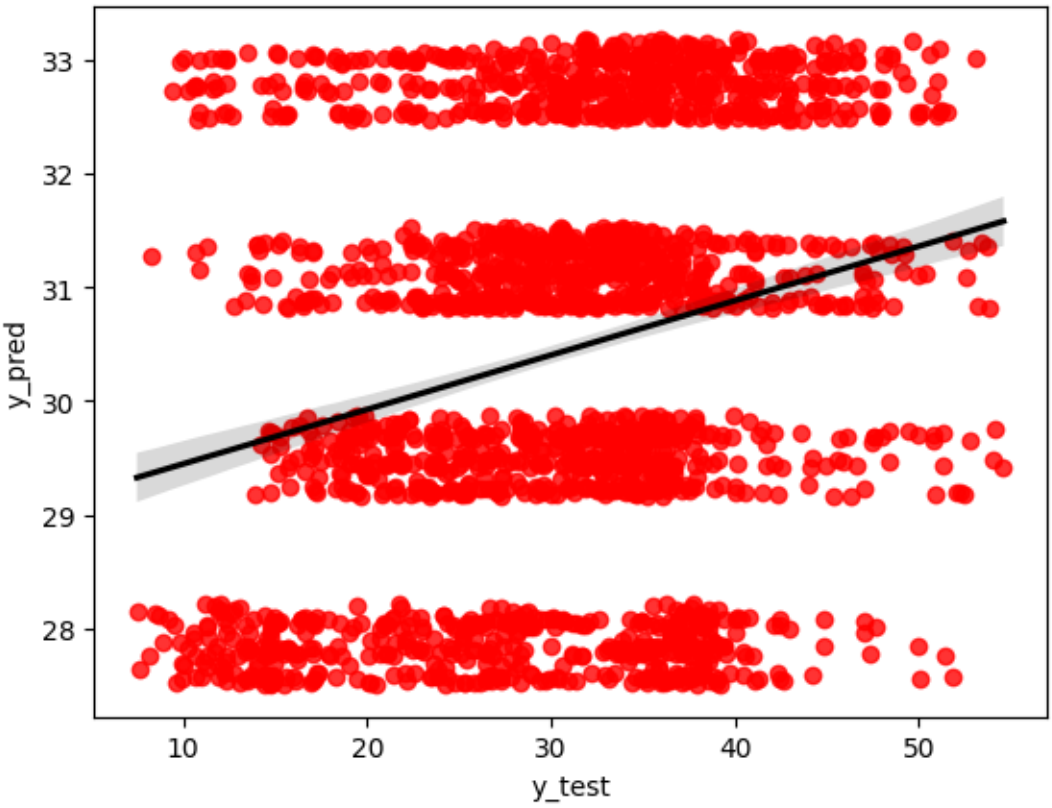
[57]:
```
fig, ax = plt.subplots(1,3,figsize=(14,4))
ax[0].plot(y_pred, y_test, 'o')
ax[0].set_xlabel("y_test")
ax[0].set_ylabel("y_pred")
ax[0].set_title("Test Set -Predicted vs real")

ax[1].hist(y_test - y_pred)
ax[1].set_xlabel("Test y-y_pred")
ax[1].set_title("Test Set Residual histogram")

ax[2].plot(y_pred,y_test - y_pred,"o")
ax[2].set_xlabel("predited")
ax[2].set_ylabel("residuals")
ax[2].set_title("Residuals by Predicted")
ax[2].plot(y_pred,np.zeros(len(y_pred)),linestyle='dashed')
```

[57]:
```
[<matplotlib.lines.Line2D at 0x1ee2724da30>]
```



[58]:
```
sns.regplot(x='y_test',y='y_pred', data=result, scatter_kws={"color": "red"},␣
↪line_kws={"color": "black"})
```

[58]: <Axes: xlabel='y_test', ylabel='y_pred'>



0.0.3 Evaluating the model using mean squared error and R-squared

[59]:
```
mse = mse(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
adj_r2= 1 - (1-r2)*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1)
print("Mean Squared Error:", mse)
print("R-squared:", r2)
print("Adjusted R-squared:", adj_r2)
```

```
Mean Squared Error: 78.83817466174246
R-squared: 0.056188307102694623
Adjusted R-squared: 0.05482573006771296
```

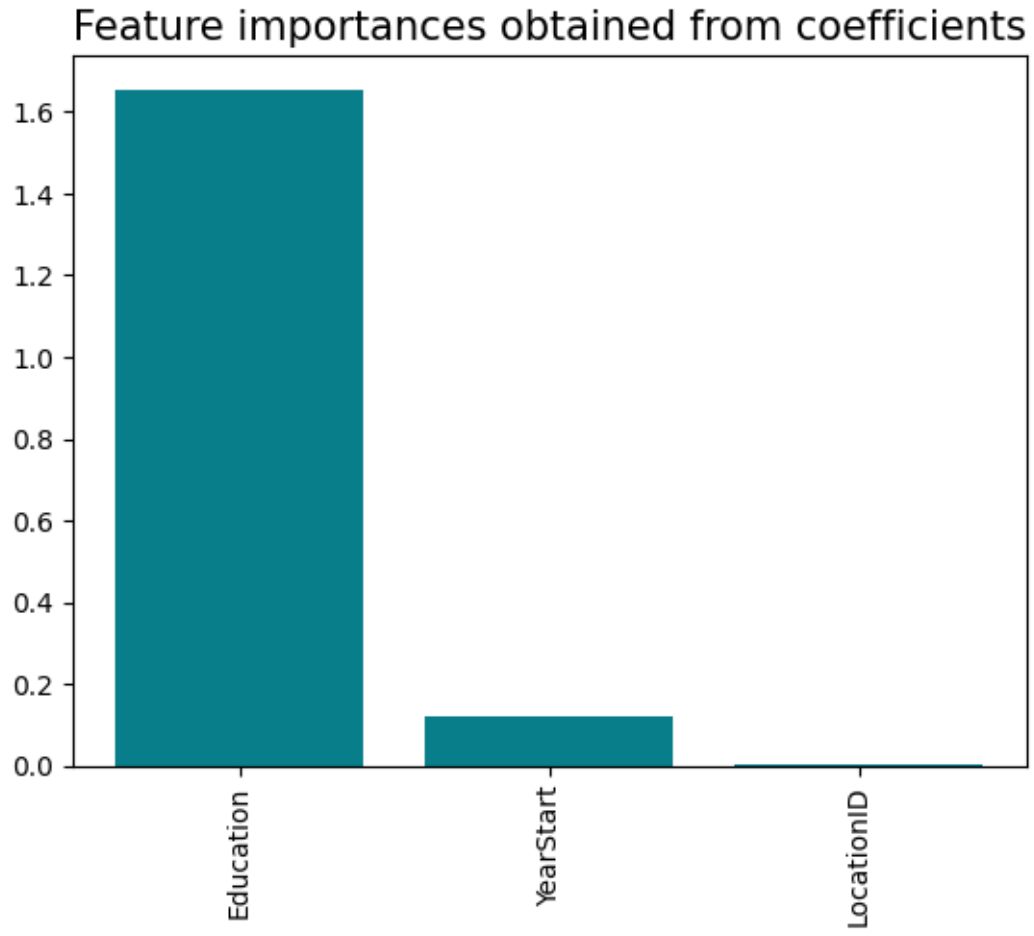
[60]:
```
features_importances = pd.DataFrame(data={
    'Attribute': X_train.columns,
    'Importance': abs(model.coef_)
})
features_importances = features_importances.sort_values(by='Importance',␣
 ↪ascending=False)
```

[61]:
```
features_importances
```

[61]:

	Attribute	Importance
1	Education	1.655448
0	YearStart	0.122138
2	LocationID	0.001826

[62]:
```
plt.bar(x=features_importances['Attribute'],␣
 ↪height=features_importances['Importance'], color='#087E8B')
plt.title('Feature importances obtained from coefficients', size=15)
plt.xticks(rotation='vertical')
plt.show()
```



[64]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot( data=df,x='Sample_Size', y='Data_Value_Alt')
plt.title('Relationship between Sample Size and Data Value (Obesity Index)')
plt.xlabel('Sample Size')
plt.ylabel('Data Value (Obesity Index)')
plt.show()
```

[ ]: