Problem Statement

UNICEF has collected the pupil dropout rate at different educational levels (primary, lower secondary and upper secondary) through various sources over the course of 10 years. This dataset contains dropout rates based on several parameters (gender, living conditions and financial conditions) in different regions around the globe. It finally mentions the total pupil dropout rate in these regions.

Objective

To use the given dataset for finding the total dropout rate given other factors. It will help determine the total dropout rate for future predictions as well and will hence help identify the areas for development and progress.

Data Extraction

Importing libraries and Global School dropouts UNICEF dataset

```
%matplotlib
```

```
Using matplotlib backend: Qt5Agg
```

```
from numpy import arange
import numpy
from matplotlib import pyplot as plt
from scipy.stats import norm
import pandas as pd
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.linear_model import LinearRegression
from sklearn import model_selection
```

```
plt.rcParams['figure.figsize'] = [16, 7]
```

All columns represent the dropout rates of pupils based on different parameters

```
columns=['Region','UNICEF Sub-region 1','Development Regions','Female','Male','Rural','Urban','Poorest','Poor','Middle Segment',
         'Rich','Richest','Data Source','Time Period','Educational Level','Total Dropout']
df = pd.read_csv(r"C:/Users/Aadya/Desktop/Global_school_dropouts_UNICEF.csv")
```

```python
df.columns = columns
df.head()
```

	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Data Source	Time Period	Educational Level	Total Dropout
0	SA	SA	Least Developed	47.0	28.0	42.0	19.0	42.0	47.0	46.0	32.0	16.0	DHS 2015	2015.0	Primary	37.0
1	ECA	EECA	More Developed	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	DHS 2017-18	2018.0	Primary	2.0
2	MENA	MENA	Less Developed	2.0	2.0	2.0	2.0	3.0	2.0	2.0	2.0	1.0	MICS 2012-13	2013.0	Primary	2.0
3	ECA	WE	More Developed	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Primary	NaN
4	SSA	ESA	Least Developed	22.0	21.0	35.0	14.0	39.0	33.0	19.0	12.0	5.0	DHS 2015-16	2016.0	Primary	22.0

Removing the first row of the dataset. It will be used later.

```python
#Original dataframe
df.head(2)
```

	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Data Source	Time Period	Educational Level	Total Dropout
0	SA	SA	Least Developed	47.0	28.0	42.0	19.0	42.0	47.0	46.0	32.0	16.0	DHS 2015	2015.0	Primary	37.0
1	ECA	EECA	More Developed	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	DHS 2017-18	2018.0	Primary	2.0

```python
#Removing the first row from the dataframe
row = df.iloc[-1]
#print(row)
row=list(row)
print(row)
```

```
['SSA', 'ESA', 'Less Developed', 54.0, 48.0, 55.0, 43.0, 63.0, 54.0, 52.0, 49.0, 38.0, 'MICS 2019', 2019.0, 'Upper secondary', 51.0]
```

```python
#Removing first row from the dataframe
df = df.drop([0],axis=0)
```

```python
#First row of the new dataframe
df.head(1)
```

	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Data Source	Time Period	Educational Level	Total Dropout
1	ECA	EECA	More Developed	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	DHS 2017-18	2018.0	Primary	2.0

Data Exploration

```
#Information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 605 entries, 1 to 605
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Region               599 non-null    object 
 1   UNICEF Sub-region 1  599 non-null    object 
 2   Development Regions  605 non-null    object 
 3   Female               329 non-null    float64
 4   Male                 329 non-null    float64
 5   Rural                321 non-null    float64
 6   Urban                321 non-null    float64
 7   Poorest              305 non-null    float64
 8   Poor                 305 non-null    float64
 9   Middle Segment       305 non-null    float64
 10  Rich                 305 non-null    float64
 11  Richest              305 non-null    float64
 12  Data Source          329 non-null    object 
 13  Time Period          329 non-null    float64
 14  Educational Level    605 non-null    object 
```

```
#Data types of the dataset columns
df.dtypes
```

```
Region                   object
UNICEF Sub-region 1      object
Development Regions      object
Female                  float64
Male                    float64
Rural                   float64
Urban                   float64
Poorest                 float64
Poor                    float64
Middle Segment          float64
Rich                    float64
Richest                 float64
Data Source              object
Time Period             float64
Educational Level        object
Total Dropout           float64
dtype: object
```

```
#Memory used by each column in the dataset
df.memory_usage()
```

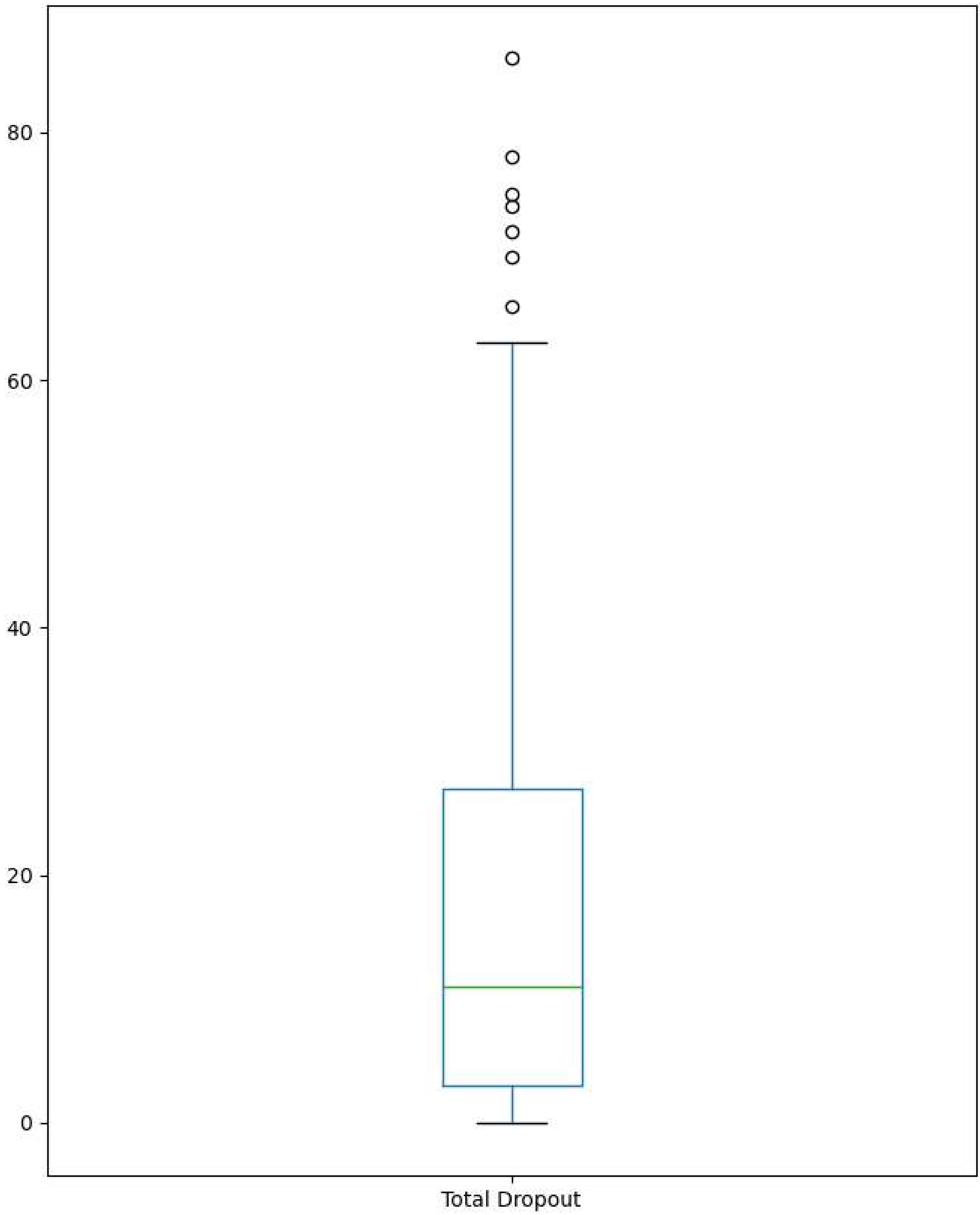
```
Index                   4840
Region                  4840
UNICEF Sub-region 1     4840
Development Regions     4840
Female                  4840
Male                    4840
Rural                   4840
Urban                   4840
Poorest                 4840
Poor                    4840
Middle Segment          4840
Rich                    4840
Richest                 4840
Data Source             4840
Time Period             4840
Educational Level       4840
Total Dropout           4840
dtype: int64
```

```
#Total memory used by the dataset
df.memory_usage().sum()
```

```
82280
```

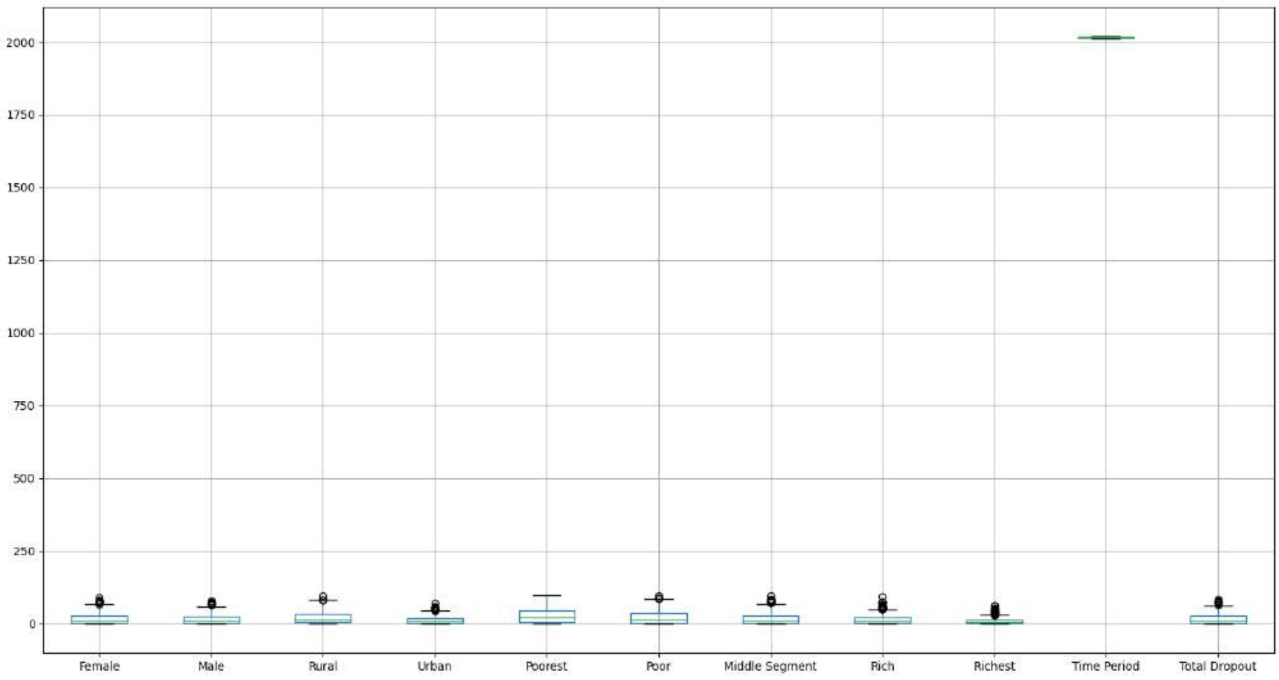
Data Visualization

```
#Boxplot
df['Total Dropout'].plot.box(figsize=(8, 15));
```

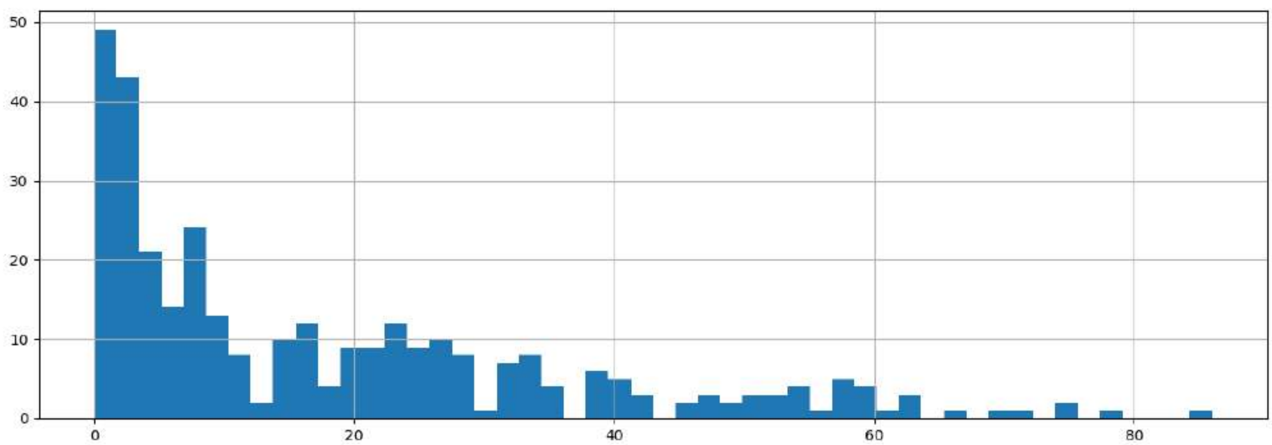

80
60
40
20
0
Total Dropout

```
#Boxplot of all the columns with numerical data
df.boxplot(figsize=(20,20))
```

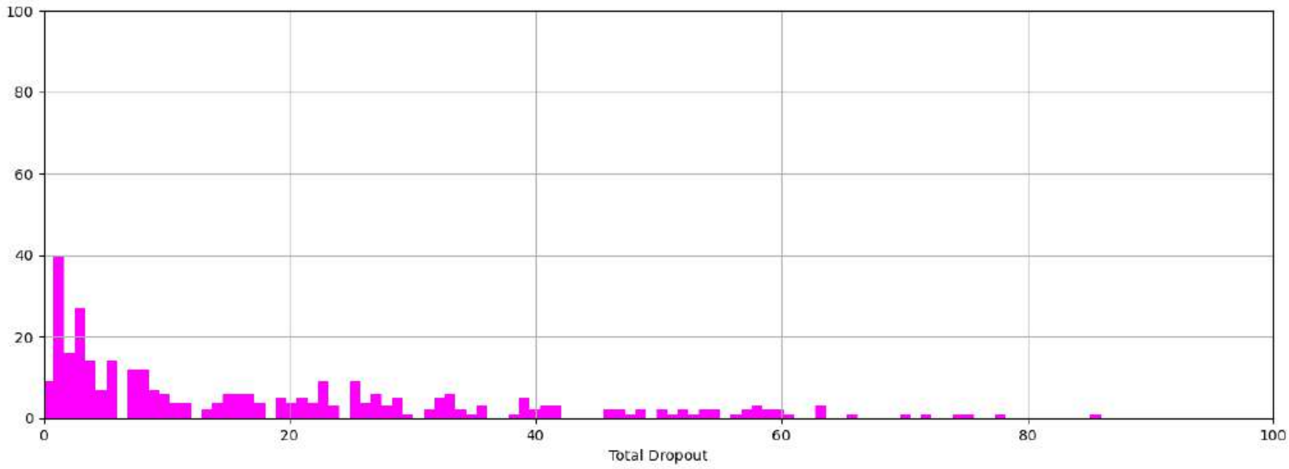
```
<AxesSubplot:>
```


2000
1750
1500
1250
1000
750
500
250
0
Female
Male
Rural
Urban
Poorest
Poor
Middle Segment
Rich
Richest
Time Period
Total Dropout

```
#Histogram
df['Total Dropout'].hist(bins=50, figsize=(15, 5));
```

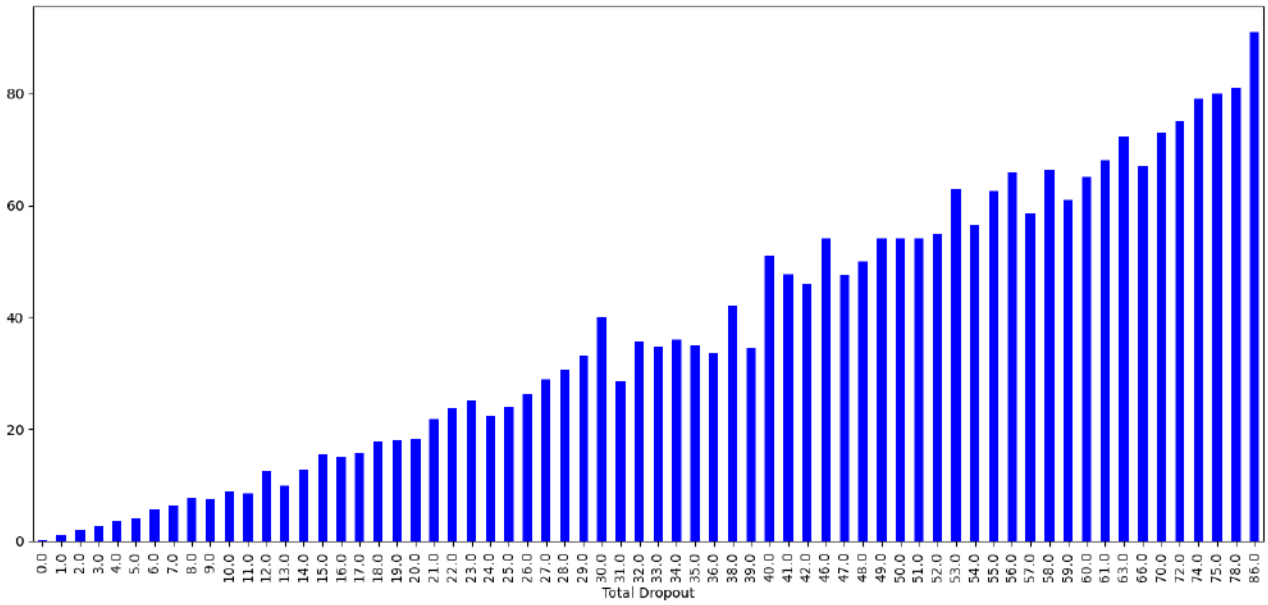

50
40
30
20
10
0
0
20
40
60
80

```
#Histogram with specifications
ax = df['Total Dropout'].hist(bins=100, color='magenta', figsize=(15, 5))
ax.set_xlabel('Total Dropout')
ax.set_xlim(0,100) #limiting display range to 0-100 for the x-axis
ax.set_ylim(0,100); #limiting display range to 0-100 for the y-axis
```



100
80
60
40
20
0
0
20
40
60
80
100
Total Dropout

```
#Barplot with Female dropout as dependent variable
df_avg_female = df.groupby('Total Dropout')['Female'].mean()
df_avg_female[:].plot.bar(color='blue');
```



0
20
40
60
80
0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0 21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0 31.0 32.0 33.0 34.0 35.0 36.0 38.0 39.0 40.0 41.0 42.0 46.0 47.0 48.0 49.0 50.0 51.0 52.0 53.0 54.0 55.0 56.0 57.0 58.0 59.0 60.0 61.0 63.0 66.0 70.0 72.0 74.0 75.0 78.0 86.0
Total Dropout

Data Cleaning

```
#Check if there are missing values in the dataset
df.isnull().sum().sum()
```

```
3460
```

```
#Removing null values from the dataset
df=df.dropna()
```

```
#Recheck if there are missing values in the dataset
df.isnull().sum().sum()
```

```
0
```

```
#Check if there are duplicate rows in the dataset
df.duplicated().sum()
```

```
0
```

```
#No. of rows in the dataset after cleaning
print(len(df.axes[0]))
```

```
302
```

Label encoding columns having non-integer values

```
df['Region'].replace({'SA':0,'ECA':1,'MENA':2,'SSA':3,'LAC':4,'EAP':5,'NA':6},inplace=True)
```

```
df['UNICEF Sub-region 1'].replace({'SA':0,'EECA':1,'MENA':2,'WE':3,'ESA':4,'LAC':5,'EAP':6,'WCA':7,'NA':8},inplace=True)
```

```
df['Development Regions'].replace({'Not Classified':0,'Least Developed':1,'Less Developed':2,'More Developed':3},inplace=True)
```

```
df['Data Source'].replace({'DHS 2015':0,'DHS 2017-18':1,'MICS 2012-13':2,'DHS 2015-16':3,'MICS 2011-12':4,'MICS 2019':5,
                           'MICS 2012':6,'MICS 2015-16':7,'MICS 2010':8,'Botswana Demographic Survey 2017':9,'PNAD 2019':10,
                           'DHS 2010':11,'DHS 2016-17':12,'DHS 2014':13,'DHS 2018':14,'DHS 2014-15':15,'CASEN 2017':16,
                           'National Population Sample Survey 2015':17,'DHS 2012':18,'MICS 2014-15':19,'MICS 2018':20,
                           'MICS 2016':21,'MICS 2017':22,'MICS 2017-18':23,'MICS 2014':24,'GEPH-ENEMDU-DICIEMBRE - 2019':25,
                           'DHS 2016-17':26,'DHS 2013':27,'MICS 2018-19':28,'Encuesta Permanente de Hogares 2018':29,
                           'National Socio-Economic Survey 2018':30,'MICS 2011':31,'MICS 2015':32,'ENIGH 2018':33,'DHS 2011':34,
                           'Population and Housing Census 2011':35,'MICS 2013':36,'DHS 2017':37,'DHS 2019':38,'DHS 2016':39,
                            'DHS style 2013':40,'Institute of Educational Sciences and UNICEF 2015-2016':41,'DHS 2015':42,
                           'Encuesta Permanente de Hogares 2017':43},
                           inplace=True)
```

```
df['Educational Level'].replace({'Primary':0,'Lower secondary':1,'Upper secondary':2},inplace=True)
```

```
df.head()
```

	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Data Source	Time Period	Educational Level	Total Dropout
1	1	1	3	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	1	2018.0	0	2.0
2	2	2	2	2.0	2.0	2.0	2.0	3.0	2.0	2.0	2.0	1.0	2	2013.0	0	2.0
4	3	4	1	22.0	21.0	35.0	14.0	39.0	33.0	19.0	12.0	5.0	3	2016.0	0	22.0
7	4	5	2	1.0	2.0	0.0	0.0	1.0	0.0	2.0	2.0	1.0	4	2012.0	0	1.0
8	1	1	2	4.0	5.0	6.0	3.0	6.0	7.0	6.0	3.0	2.0	3	2016.0	0	5.0

```
#Statistics for all the dataset columns
df.describe()
```

	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	
count	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.
mean	2.821192	4.185430	1.675497	19.930464	18.370861	22.874172	13.473510	29.771523	23.135762	19.139073	14.682119	9.357616	19.
std	1.393321	2.321083	0.615953	20.900932	17.409134	22.069448	13.822473	25.967091	23.295832	20.807787	17.465817	12.121941	11.
min	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.
25%	2.000000	2.000000	1.000000	3.000000	4.000000	4.000000	3.000000	6.000000	3.250000	3.000000	2.000000	1.000000	7.
50%	3.000000	5.000000	2.000000	12.500000	14.000000	16.000000	9.000000	24.000000	15.000000	11.000000	7.000000	4.000000	20.
75%	4.000000	6.000000	2.000000	30.750000	27.750000	35.750000	20.000000	46.750000	36.000000	29.000000	21.000000	13.000000	27.
max	5.000000	7.000000	3.000000	91.000000	79.000000	95.000000	70.000000	99.000000	97.000000	97.000000	92.000000	63.000000	43.

```python
#Variance
df.var()
```

```
Region                    1.941343
UNICEF Sub-region 1       5.387428
Development Regions       0.379398
Female                  436.848969
Male                    303.077952
Rural                   487.060527
Urban                   191.060758
Poorest                 674.289818
Poor                    542.695793
Middle Segment          432.963983
Rich                    305.054762
Richest                 146.941453
Data Source             137.770929
Time Period               6.668731
Educational Level         0.667763
Total Dropout           358.461739
dtype: float64
```

```python
#Skewness
df.skew()
```

```
Region                 -0.441176
UNICEF Sub-region 1    -0.435417
Development Regions     0.336217
Female                  1.195539
Male                    1.109291
Rural                   0.965717
Urban                   1.398729
Poorest                 0.708961
Poor                    1.026683
Middle Segment          1.304636
Rich                    1.620148
Richest                 1.966928
Data Source             0.138877
Time Period            -0.399372
Educational Level       0.006109
Total Dropout           1.139648
dtype: float64
```

```
#Kurtosis
df.kurtosis()
```

```
Region                 -0.506458
UNICEF Sub-region 1    -1.092629
Development Regions    -0.652230
Female                  0.583931
Male                    0.623715
Rural                   0.033932
Urban                   1.610341
Poorest                -0.575796
Poor                    0.146258
Middle Segment          1.026929
Rich                    2.248416
Richest                 3.859236
Data Source            -0.955609
Time Period            -0.942130
Educational Level      -1.502421
Total Dropout           0.596511
dtype: float64
```

Data Selection

```
#Column-wise correlation in the dataset
df.corr()
```

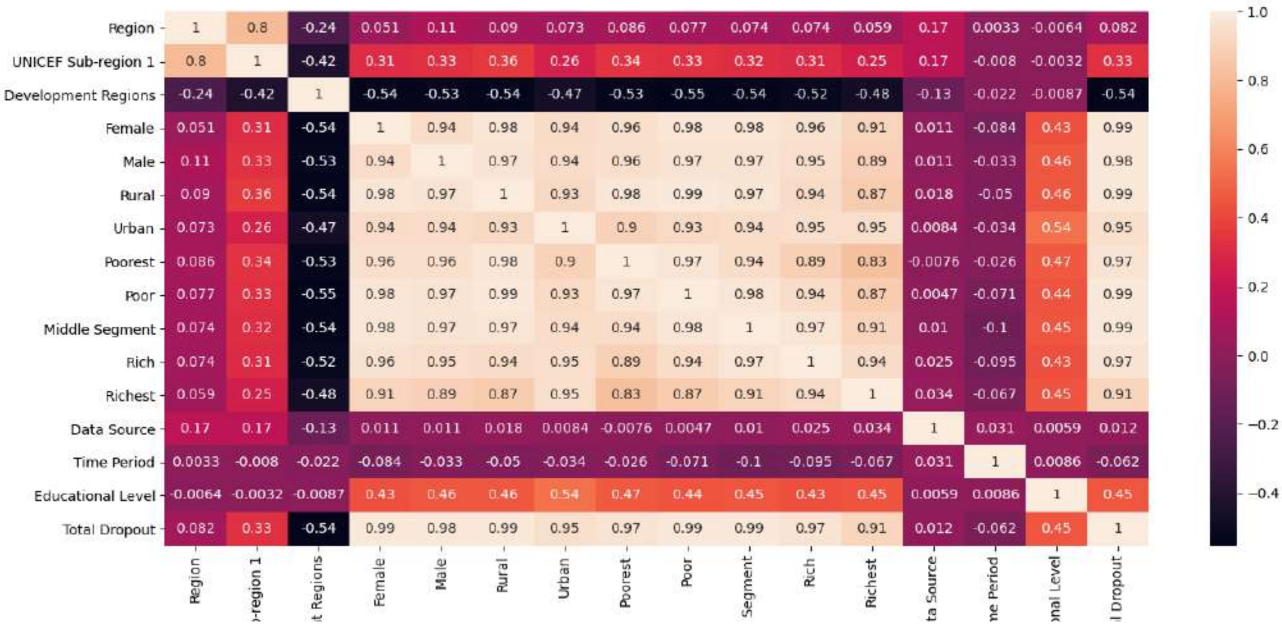
	Region	UNICEF Sub-region 1	Development Regions	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Data Source
Region	1.000000	0.803352	-0.242034	0.051023	0.113273	0.089913	0.073412	0.086192	0.076595	0.073512	0.073834	0.059465	0.169983
UNICEF Sub-region 1	0.803352	1.000000	-0.420205	0.314189	0.328479	0.359825	0.264417	0.338213	0.331933	0.320295	0.307955	0.252330	0.170712
Development Regions	-0.242034	-0.420205	1.000000	-0.540846	-0.530614	-0.537997	-0.473950	-0.530579	-0.553056	-0.538227	-0.521943	-0.476969	-0.125495
Female	0.051023	0.314189	-0.540846	1.000000	0.942020	0.978000	0.938331	0.955428	0.979814	0.980997	0.959708	0.907020	0.011002
Male	0.113273	0.328479	-0.530614	0.942020	1.000000	0.967262	0.941299	0.955686	0.968438	0.965511	0.945611	0.891424	0.011085
Rural	0.089913	0.359825	-0.537997	0.978000	0.967262	1.000000	0.925329	0.979225	0.988447	0.973776	0.938247	0.870049	0.017650
Urban	0.073412	0.264417	-0.473950	0.938331	0.941299	0.925329	1.000000	0.900580	0.927209	0.942868	0.945571	0.949493	0.008391
Poorest	0.086192	0.338213	-0.530579	0.955428	0.955686	0.979225	0.900580	1.000000	0.973056	0.939404	0.888807	0.826003	-0.007606
Poor	0.076595	0.331933	-0.553056	0.979814	0.968438	0.988447	0.927209	0.973056	1.000000	0.977809	0.939863	0.871761	0.004670

```python
#Import seaborn library
import seaborn as sns
```

Pearson correlation

```python
sns.heatmap(df.corr('pearson'),annot=True)
```

```
<AxesSubplot:>
```

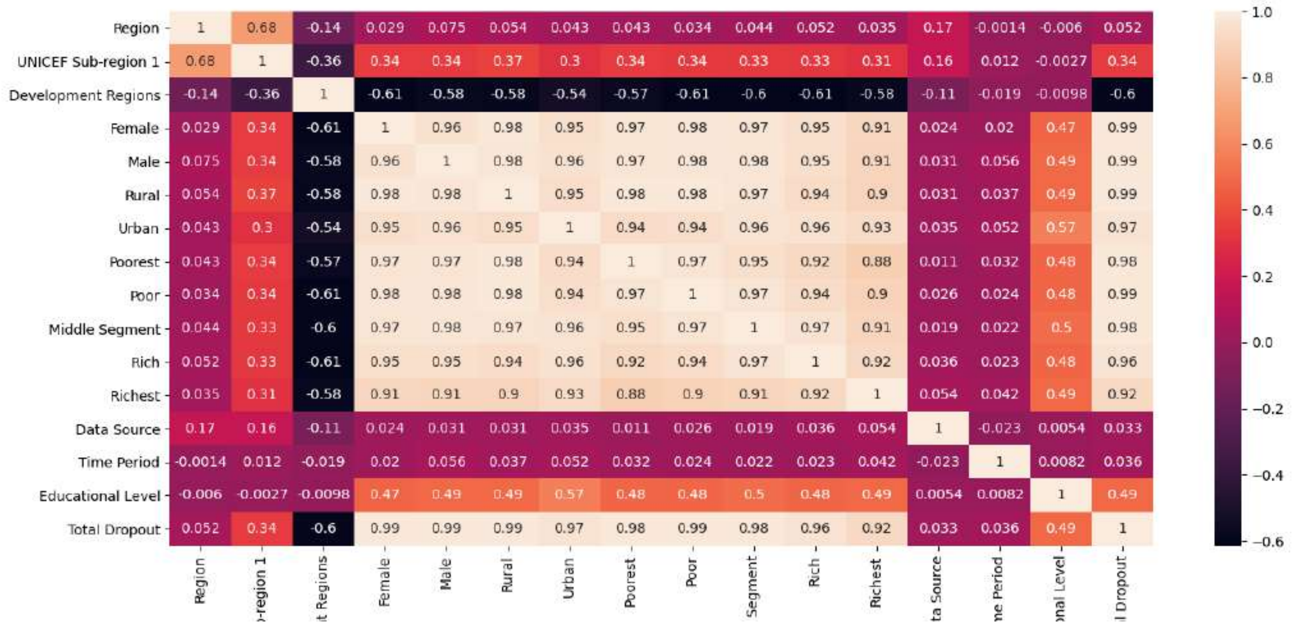


Region
UNICEF Sub-region 1
Development Regions
Female
Male
Rural
Urban
Poorest
Poor
Middle Segment
Rich
Richest
Data Source
Time Period
Educational Level
Total Dropout

Spearman correlation

```
sns.heatmap(df.corr('spearman'),annot=True)
```

```
<AxesSubplot:>
```



Region
UNICEF Sub-region 1
Development Regions
Female
Male
Rural
Urban
Poorest
Poor
Middle Segment
Rich
Richest
Data Source
Time Period
Educational Level
Total Dropout

Kendall correlation

```
sns.heatmap(df.corr('kendall'),annot=True)
```

```
<AxesSubplot:>
```


Region
UNICEF Sub-region 1
Development Regions
Female
Male
Rural
Urban
Poorest
Poor
Middle Segment
Rich
Richest
Data Source
Time Period
Educational Level
Total Dropout
Region
)-region 1
t Regions
Female
Male
Rural
Urban
Poorest
Poor
Segment
Rich
Richest
ta Source
ne Period
onal Level
l Dropout
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Pearson, Spearman and Kendall correlations give similar results.

Target column: Total Dropout

Columns: Region, UNICEF Sub-region 1, Development Regions, Data Source, Time Period, Educational Level are not well correlated with the target column Total Dropout.

Thus, they are dropped.

```
df=df.drop(['Region','UNICEF Sub-region 1','Development Regions','Data Source','Time Period','Educational Level'],axis=1)
```

```
df.replace('', numpy.nan, inplace=True)
```

```
df.dropna(inplace=True)
```

```
df.head()
```

	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Total Dropout
1	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	2.0
2	2.0	2.0	2.0	2.0	3.0	2.0	2.0	2.0	1.0	2.0
4	22.0	21.0	35.0	14.0	39.0	33.0	19.0	12.0	5.0	22.0
7	1.0	2.0	0.0	0.0	1.0	0.0	2.0	2.0	1.0	1.0
8	4.0	5.0	6.0	3.0	6.0	7.0	6.0	3.0	2.0	5.0

Data Splitting and Model Building (Multiple Linear regression)

```
X = df.iloc[:, :-1]
y = df['Total Dropout']
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()
regressor.fit(X, y)
```

```
LinearRegression()
```

```
y_pred = regressor.predict(X_test)
```

```
print(y_pred)
```

```
[ 5.65085809e+01  1.98566797e+01  2.38594371e+01  1.88014224e+01
  2.12666582e+01  8.95956449e+00  8.08405528e+00  3.56249450e+00
  2.59385948e+01  6.00315192e+01  1.91092677e+01  2.88559579e+01
  4.18685440e+01  4.97072943e-01  5.99296303e+00  2.26575444e+01
  1.50598537e+00  2.99933724e+00  5.92219507e+00  5.54749170e+00
  1.02135053e+00  3.96013144e+01  2.86388029e+01  1.49798576e+01
  1.42718560e+00  5.37319997e+01  1.71023379e+01  2.60585343e+01
  7.99080806e+00  4.72578008e+01  2.26446909e+01  2.95501978e+01
  6.99973227e+00  2.00626865e+00  4.10242784e+01  4.68147663e+01
  1.45444644e+01  9.86201644e-01  7.48808149e+00  2.67812329e+01
  6.45792045e+00  2.29550465e+01  1.42672425e+00  5.87622554e+01
  5.45160389e+00  5.45622213e+00  2.54353063e+01  1.20294562e+01
  1.96278998e+01  4.50051474e-01  7.22371953e-02  4.49021620e+00
  7.07705014e+00  4.91106840e+01  3.26903196e+01  1.60171239e+01
 -9.17124608e-04  5.10355482e+01  3.00040295e+00  1.92659010e-02
  9.96400025e-01]
```

```python
print(y_test)
```

```
447    56.0
302    20.0
459    24.0
403    19.0
105    22.0
       ...
293     0.0
605    51.0
55      3.0
218     0.0
182     1.0
Name: Total Dropout, Length: 61, dtype: float64
```

```python
#Checking the model accuracy
regressor.score(X_test, y_test)
```

```
0.9996471691780692
```

```python
#Co-efficients of the Logistic regression equation
regressor.coef_
```

```
array([0.43032422, 0.43661876, 0.02184386, 0.01119668, 0.01934443,
       0.02018303, 0.02411191, 0.01434984, 0.02349589])
```

```python
#y-intercept of the Logistic regression equation
regressor.intercept_
```

```
-0.0009171246082964046
```

Multiple Linear regression equation

y -> target variable i.e. Total Dropout

a -> y-intercept of Total Dropout

b0 -> co-efficient of Female

b1 -> co-efficient of Male

b2 -> co-efficient of Rural

b3 -> co-efficient of Urban

b4 -> co-efficient of Poorest

b5 -> co-efficient of Poor

b6 -> co-efficient of Middle Segment

b7 -> co-efficient of Rich

b8 -> co-efficient of Richest

General equation: y = a + b0x0 + b1x1 + ... + bnxn

Actual equation:

Total Dropout = -0.00092 + 0.4303(Female) + 0.4366(Male) + 0.0218(Rural) + 0.0112(Urban) + 0.0193(Poorest) + 0.0202(Poor) + 0.0241(Middle Segment) + 0.0143(Rich) + 0.0235(Richest)

Model evaluation through k-fold cross validation and evaluation metrics

```
#K-fold cross-validation
#Linear Regression
X = df.iloc[:,:-1]
y = df.iloc[:,-1]
k = 5
kf = model_selection.KFold(n_splits=k, random_state=None)
model = LinearRegression()
result = cross_val_score(model , X, y, cv = kf)
print("Avg accuracy: {}".format(result.mean()))
```

```
Avg accuracy: 0.9994454579559378
```

```
# Root mean square error
import sklearn
sklearn.metrics.mean_squared_error(y_test,y_pred)
```

```
0.10637379914763806
```

```
#R2 score
import sklearn
sklearn.metrics.r2_score(y_test,y_pred)
```

```
0.9996471691780692
```

Cross-checking the model with one of the dataset samples

```
df.head(1)
```

	Female	Male	Rural	Urban	Poorest	Poor	Middle Segment	Rich	Richest	Total Dropout
1	2.0	3.0	4.0	1.0	4.0	3.0	2.0	2.0	1.0	2.0

```
print("Total dropout rate: "+str(int(regressor.predict([[47,28,42,19,42,47,46,32,16]])))+"%")
```

```
Total dropout rate: 37%
```

Using the model to predict the total dropout rate for unseen data

```
#Using the "row" saved earlier
#It is a data sample never seen by the model before

actual_total_dropout_rate = row[-1]
print("Actual total dropout rate: "+str(actual_total_dropout_rate)+"%")

row.pop()
row=row[3:-3]
print(row)

#Predicting using the model
predicted_total_dropout_rate = regressor.predict([row])[0]
print("Predicted total dropout rate: "+str(round(predicted_total_dropout_rate,0))+"%")
```

```
Actual total dropout rate: 51.0%
[54.0, 48.0, 55.0, 43.0, 63.0, 54.0, 52.0, 49.0, 38.0]
Predicted total dropout rate: 51.0%
```

The results show that the most important predictors for the Total Dropout rate of pupils in different regions globally are

Female dropout rate

Male dropout rate

Rural dropout rate

Urban dropout rate

Poorest dropout rate

Poor dropout rate

Middle Segment dropout rate

Rich dropout rate

Richest dropout rate

While the other factors

Region

UNICEF Sub-region 1

Development Regions

Data Source

Time Period

Educational Level

do impact the Total dropout rate yet they are not very significant

Since the model has an extremely high accuracy rate of more than 99%, it can be used to further predict the dropout rate in different regions given the important predictors in the future.

Thus, it can be used to identify the key areas for development and eventually aid in increasing the pupil retention rate globally in Primary, Lower Secondary and Upper Secondary education.