

# eda1

November 20, 2024

```
[3]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
import warnings
warnings.filterwarnings('ignore')

data=pd.read_csv('/content/Cardiotocographic.csv')
data
```

```
[3]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	\
0	120.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	73.0	
1	132.000000	0.006380	0.000000	0.006380	0.003190	0.0	0.0	17.0	
2	133.000000	0.003322	0.000000	0.008306	0.003322	0.0	0.0	16.0	
3	134.000000	0.002561	0.000000	0.007742	0.002561	0.0	0.0	16.0	
4	131.948232	0.006515	0.000000	0.008143	0.000000	0.0	0.0	16.0	
...	...	...	...	...	...	...	...	...	...
2121	140.000000	0.000000	0.961268	0.007426	0.000000	0.0	0.0	79.0	
2122	140.000000	0.000775	0.000000	0.006979	0.000000	0.0	0.0	78.0	
2123	140.000000	0.000980	0.000000	0.006863	0.000000	0.0	0.0	79.0	
2124	140.000000	0.000679	0.000000	0.006110	0.000000	0.0	0.0	78.0	
2125	142.000000	0.001616	-0.000188	0.008078	0.000000	0.0	0.0	74.0	

	MSTV	ALTV	MLTV	Width	Tendency	NSP
0	0.5	43.000000	2.4	64.0	0.999926	2.000000
1	2.1	0.000000	10.4	130.0	0.000000	1.000000
2	2.1	0.000000	13.4	130.0	0.000000	1.000000
3	2.4	0.000000	23.0	117.0	1.000000	1.000000
4	2.4	0.000000	19.9	117.0	1.000000	1.000000
...	...	...	...	...	...	...
2121	0.2	25.000000	7.2	40.0	0.000000	2.000000
2122	0.4	22.000000	7.1	66.0	1.000000	2.000000
2123	0.4	20.000000	6.1	67.0	1.000000	1.990464
2124	0.4	27.000000	7.0	66.0	1.000000	2.000000
2125	0.4	35.857183	5.0	42.0	0.000000	1.000000

[2126 rows x 14 columns]

```
[4]: df=data.iloc[:, :10]  
df
```

```
[4]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	\
0	120.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.0	73.0	
1	132.000000	0.006380	0.000000	0.006380	0.003190	0.0	0.0	17.0	
2	133.000000	0.003322	0.000000	0.008306	0.003322	0.0	0.0	16.0	
3	134.000000	0.002561	0.000000	0.007742	0.002561	0.0	0.0	16.0	
4	131.948232	0.006515	0.000000	0.008143	0.000000	0.0	0.0	16.0	
...	...	...	...	...	...	...	...	...	...
2121	140.000000	0.000000	0.961268	0.007426	0.000000	0.0	0.0	79.0	
2122	140.000000	0.000775	0.000000	0.006979	0.000000	0.0	0.0	78.0	
2123	140.000000	0.000980	0.000000	0.006863	0.000000	0.0	0.0	79.0	
2124	140.000000	0.000679	0.000000	0.006110	0.000000	0.0	0.0	78.0	
2125	142.000000	0.001616	-0.000188	0.008078	0.000000	0.0	0.0	74.0	
	MSTV	ALTV							
0	0.5	43.000000							
1	2.1	0.000000							
2	2.1	0.000000							
3	2.4	0.000000							
4	2.4	0.000000							
...	...	...							
2121	0.2	25.000000							
2122	0.4	22.000000							
2123	0.4	20.000000							
2124	0.4	27.000000							
2125	0.4	35.857183							

[2126 rows x 10 columns]

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

```
[5]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	\
0	120.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	73.0	0.5	
1	132.000000	0.006380	0.0	0.006380	0.003190	0.0	0.000000	17.0	2.1	
2	133.000000	0.003322	0.0	0.008306	0.003322	0.0	0.000000	16.0	2.1	
3	134.000000	0.002561	0.0	0.007742	0.002561	0.0	0.000000	16.0	2.4	
4	131.948232	0.006515	0.0	0.008143	0.000000	0.0	0.000000	16.0	2.4	
5	134.000000	0.001116	0.0	0.010493	0.009444	0.0	0.002099	26.0	5.9	
6	134.000000	0.001403	0.0	0.012623	0.008415	0.0	0.002805	29.0	6.3	
7	122.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	83.0	0.5	
8	122.000000	0.038567	0.0	0.001517	0.000000	0.0	0.000000	84.0	0.5	
9	122.000000	0.000000	0.0	0.002967	0.000000	0.0	0.000000	86.0	0.3	

	ALTV
0	43.0
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	6.0
8	5.0
9	6.0

```
[7]: df.tail(10)
```

```
[7]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	\
2116	140.0	0.003968	0.000000	0.003968	0.000000	0.0	0.0	80.0	0.2	
2117	140.0	0.000000	0.000000	0.007812	0.000000	0.0	0.0	79.0	0.3	
2118	140.0	0.000000	0.000000	0.006470	0.000898	0.0	0.0	79.0	0.5	
2119	140.0	0.000000	0.000000	0.006764	0.001127	0.0	0.0	79.0	0.6	
2120	140.0	0.000000	0.000000	0.004975	0.001244	0.0	0.0	77.0	0.7	
2121	140.0	0.000000	0.961268	0.007426	0.000000	0.0	0.0	79.0	0.2	
2122	140.0	0.000775	0.000000	0.006979	0.000000	0.0	0.0	78.0	0.4	
2123	140.0	0.000980	0.000000	0.006863	0.000000	0.0	0.0	79.0	0.4	
2124	140.0	0.000679	0.000000	0.006110	0.000000	0.0	0.0	78.0	0.4	
2125	142.0	0.001616	-0.000188	0.008078	0.000000	0.0	0.0	74.0	0.4	

	ALTV
2116	36.000000
2117	20.000000
2118	26.428179
2119	27.000000
2120	17.000000
2121	25.000000
2122	22.000000
2123	20.000000
2124	27.000000
2125	35.857183

```
[8]: type(df)
```

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

```
[9]: df.dtypes
```

```
[9]: LB      float64
     AC      float64
     FM      float64
```

```

UC      float64
DL      float64
DS      float64
DP      float64
ASTV    float64
MSTV    float64
ALTV    float64
dtype: object

```

```
[10]: df.shape
```

```
[10]: (2126, 10)
```

```
[11]: df.size
```

```
[11]: 21260
```

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

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 10 columns):
#   Column  Non-Null Count  Dtype
---  -
0   LB      2105 non-null    float64
1   AC      2106 non-null    float64
2   FM      2126 non-null    float64
3   UC      2126 non-null    float64
4   DL      2126 non-null    float64
5   DS      2105 non-null    float64
6   DP      2105 non-null    float64
7   ASTV    2126 non-null    float64
8   MSTV    2126 non-null    float64
9   ALTV    2126 non-null    float64
dtypes: float64(10)
memory usage: 166.2 KB

```

```
[13]: df.describe()
```

```

[13]:
count    LB      AC      FM      UC      DL  \
count    2105.000000  2106.000000  2126.000000  2126.000000  2126.000000
mean      133.343598    0.003219    0.009894    0.004391    0.001895
std        11.270154    0.004391    0.067540    0.003340    0.003343
min         51.842487   -0.019284   -0.480634   -0.014925   -0.015393
25%        126.000000    0.000000    0.000000    0.001851    0.000000
50%        133.000000    0.001634    0.000000    0.004484    0.000000
75%        140.000000    0.005650    0.002567    0.006536    0.003289

```

max	214.000000	0.038567	0.961268	0.030002	0.030769
-----	------------	----------	----------	----------	----------

	DS	DP	ASTV	MSTV	ALTV
count	2105.000000	2105.000000	2126.000000	2126.000000	2126.000000
mean	0.000003	0.000175	46.995984	1.364378	10.285964
std	0.000142	0.000840	18.813973	1.173632	21.205041
min	-0.001353	-0.005348	-63.000000	-6.600000	-91.000000
25%	0.000000	0.000000	32.000000	0.700000	0.000000
50%	0.000000	0.000000	49.000000	1.200000	0.000000
75%	0.000000	0.000000	61.000000	1.700000	11.000000
max	0.002706	0.010695	162.000000	13.800000	182.000000

```
[14]: df.isnull()
```

```
[14]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	MSTV	ALTV
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
2121	False	False	False	False	False	False	False	False	False	False
2122	False	False	False	False	False	False	False	False	False	False
2123	False	False	False	False	False	False	False	False	False	False
2124	False	False	False	False	False	False	False	False	False	False
2125	False	False	False	False	False	False	False	False	False	False

[2126 rows x 10 columns]

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

```
[15]: LB      21
      AC      20
      FM       0
      UC       0
      DL       0
      DS      21
      DP      21
      ASTV     0
      MSTV     0
      ALTV     0
      dtype: int64
```

```
[16]: df['LB']=df['LB'].fillna(df['LB'].mean())
      df['AC']=df['AC'].fillna(df['AC'].mean())
      df['DS']=df['DS'].fillna(df['DS'].mean())
      df['DP']=df['DP'].fillna(df['DP'].mean())
```

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

```
[17]: LB      0
      AC      0
      FM      0
      UC      0
      DL      0
      DS      0
      DP      0
      ASTV    0
      MSTV    0
      ALTV    0
      dtype: int64
```

```
[18]: df.duplicated().sum()
```

```
[18]: 6
```

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

```
[19]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV	\
68	140.0	0.007278	0.000000	0.004367	0.000000	0.0	0.000000	34.0	
306	145.0	0.000000	0.020249	0.000000	0.000000	0.0	0.000000	77.0	
791	123.0	0.002660	0.003989	0.000000	0.000000	0.0	0.000000	50.0	
1115	122.0	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	19.0	
1381	121.0	0.000000	0.000000	0.003478	0.005217	0.0	0.001739	24.0	
1458	148.0	0.004717	0.000000	0.002358	0.000000	0.0	0.000000	40.0	

	MSTV	ALTV
68	1.2	0.0
306	0.2	45.0
791	0.9	4.0
1115	1.9	0.0
1381	1.5	0.0
1458	0.9	0.0

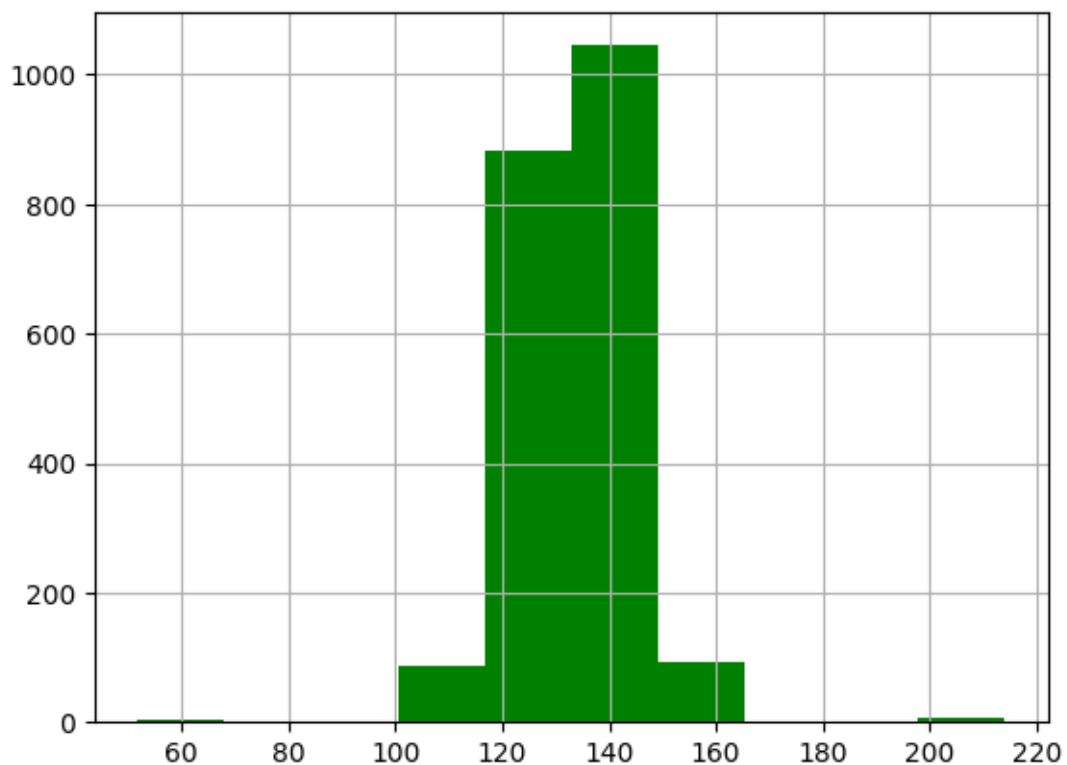
```
[20]: df2=df.drop_duplicates()
```

```
[21]: df2.shape
```

```
[21]: (2120, 10)
```

```
[22]: df2['LB'].hist(bins=10,color='green')
```

```
[22]: <Axes: >
```



```
[23]: df2.skew()
```

```
[23]: LB      0.322978
      AC      2.025886
      FM      6.743374
      UC      0.973240
      DL      2.008601
      DS      8.448745
      DP      6.460347
      ASTV    0.053887
      MSTV    4.139738
      ALTV    2.982116
      dtype: float64
```

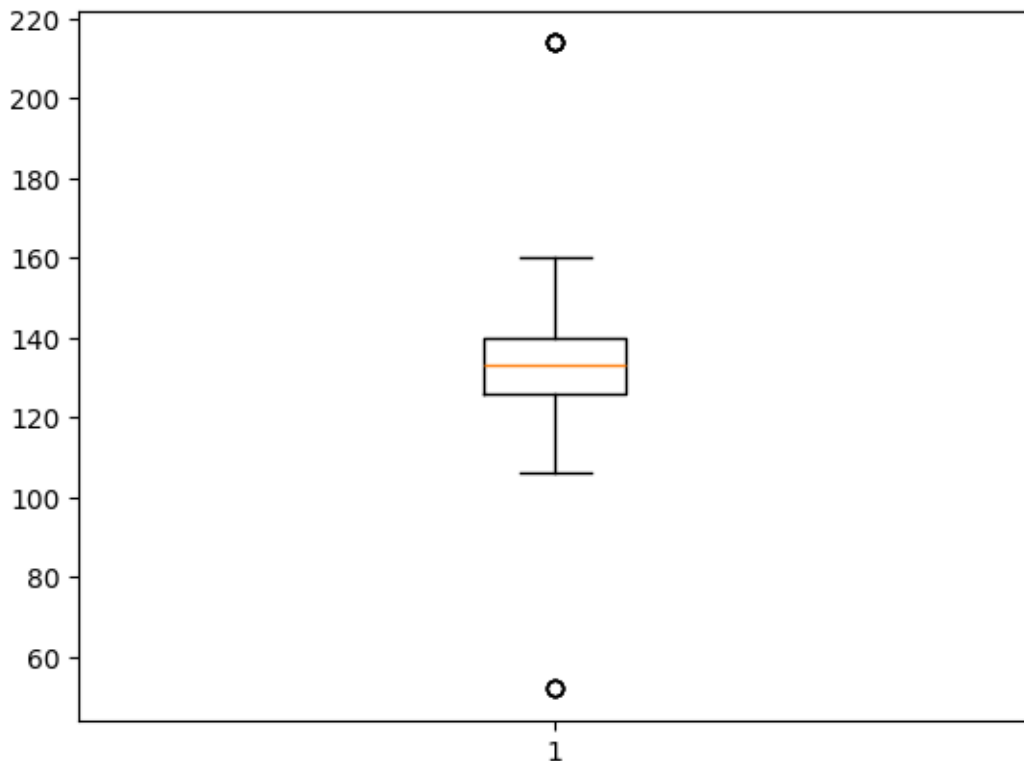
```
[24]: df2.var()
```

```
[24]: LB      1.257477e+02
      AC      1.912785e-05
      FM      4.574273e-03
      UC      1.116059e-05
      DL      1.119147e-05
      DS      1.989047e-08
```

```
DP      6.989970e-07
ASTV    3.538162e+02
MSTV    1.380312e+00
ALTV    4.501399e+02
dtype: float64
```

```
[25]: plt.boxplot(df2.LB)
```

```
[25]: {'whiskers': [<matplotlib.lines.Line2D at 0x7bc3b5b25ed0>,
<matplotlib.lines.Line2D at 0x7bc3b6d18730>],
'caps': [<matplotlib.lines.Line2D at 0x7bc3b5b24af0>,
<matplotlib.lines.Line2D at 0x7bc3b5b27c40>],
'boxes': [<matplotlib.lines.Line2D at 0x7bc3b5b26530>],
'medians': [<matplotlib.lines.Line2D at 0x7bc3b5b255d0>],
'fliers': [<matplotlib.lines.Line2D at 0x7bc3b5b25330>],
'means': []}
```



```
[26]: df2['LB'].describe()
```

```
[26]: count    2120.000000
mean      133.344099
std       11.213729
```



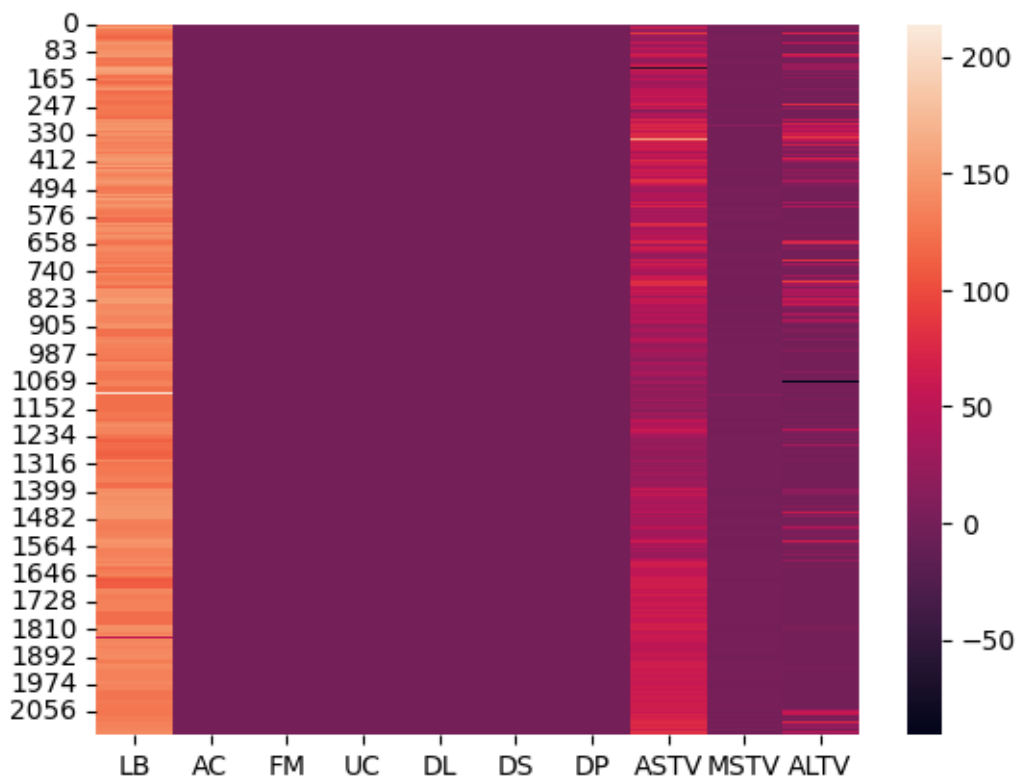
```

min      51.842487
25%     126.000000
50%     133.000000
75%     140.000000
max      214.000000
Name: LB, dtype: float64

```

```
[27]: sns.heatmap(df2)
```

```
[27]: <Axes: >
```



```
[28]: df2[['LB', 'FM']].corr()
```

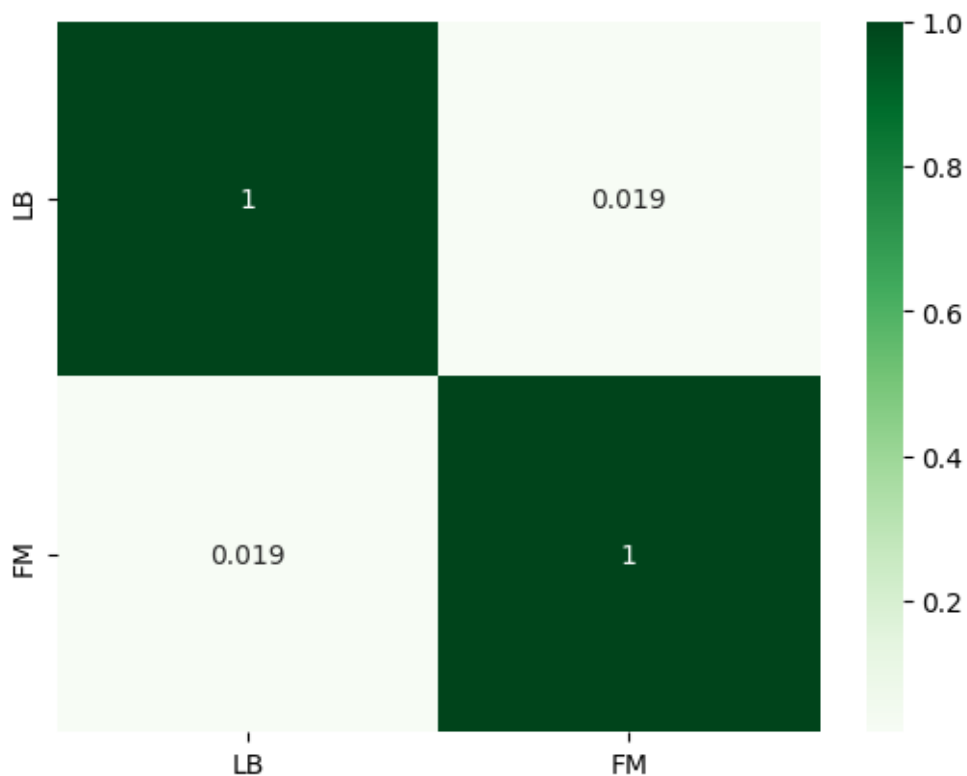
```

[28]:      LB      FM
LB  1.000000  0.018676
FM  0.018676  1.000000

```

```
[29]: sns.heatmap(df2[['LB', 'FM']].corr(), annot=True, cmap='Greens')
```

```
[29]: <Axes: >
```



```
[30]: std_scaler=StandardScaler()
```

```
[31]: df2['LB']=std_scaler.fit_transform(df2[['LB']])
df2['ASTV']=std_scaler.fit_transform(df2[['ASTV']])
df2['MSTV']=std_scaler.fit_transform(df2[['MSTV']])
df2['ALTV']=std_scaler.fit_transform(df2[['ALTV']])
df2['Ds']=std_scaler.fit_transform(df2[['DS']])
```

```
[32]: df2
```

```
[32]:
```

	LB	AC	FM	UC	DL	DS	DP	ASTV \
0	-1.190259	0.000000	0.000000	0.000000	0.000000	0.0	0.0	1.381830
1	-0.119890	0.006380	0.000000	0.006380	0.003190	0.0	0.0	-1.596011
2	-0.030693	0.003322	0.000000	0.008306	0.003322	0.0	0.0	-1.649187
3	0.058505	0.002561	0.000000	0.007742	0.002561	0.0	0.0	-1.649187
4	-0.124508	0.006515	0.000000	0.008143	0.000000	0.0	0.0	-1.649187
...	...	...	...	...	...	...	...	...
2121	0.593689	0.000000	0.961268	0.007426	0.000000	0.0	0.0	1.700885
2122	0.593689	0.000775	0.000000	0.006979	0.000000	0.0	0.0	1.647709
2123	0.593689	0.000980	0.000000	0.006863	0.000000	0.0	0.0	1.700885
2124	0.593689	0.000679	0.000000	0.006110	0.000000	0.0	0.0	1.647709
2125	0.772084	0.001616	-0.000188	0.008078	0.000000	0.0	0.0	1.435006

	MSTV	ALTV	Ds
0	-0.736535	1.541996	-0.017845
1	0.625643	-0.485207	-0.017845
2	0.625643	-0.485207	-0.017845
3	0.881051	-0.485207	-0.017845
4	0.881051	-0.485207	-0.017845
...	...	...	...
2121	-0.991944	0.693399	-0.017845
2122	-0.821671	0.551967	-0.017845
2123	-0.821671	0.457678	-0.017845
2124	-0.821671	0.787688	-0.017845
2125	-0.821671	1.205253	-0.017845

[2120 rows x 11 columns]

```
[33]: df2.mean()
```

```
[33]: LB      -9.384527e-17
AC       3.221176e-03
FM       9.911054e-03
UC       4.398229e-03
DL       1.898248e-03
DS       2.516084e-06
DP       1.748266e-04
ASTV     6.703233e-17
MSTV     8.043880e-17
ALTV     0.000000e+00
Ds       2.513713e-18
dtype: float64
```

```
[34]: df2.median()
```

```
[34]: LB      -0.030693
AC       0.001668
FM       0.000000
UC       0.004505
DL       0.000000
DS       0.000000
DP       0.000000
ASTV     0.105612
MSTV    -0.140582
ALTV    -0.485207
Ds      -0.017845
dtype: float64
```

```
[35]: df2.std()
```

```
[35]: LB      1.000236
      AC      0.004374
      FM      0.067633
      UC      0.003341
      DL      0.003345
      DS      0.000141
      DP      0.000836
      ASTV    1.000236
      MSTV    1.000236
      ALTV    1.000236
      Ds      1.000236
      dtype: float64
```

```
[36]: df2.var()
```

```
[36]: LB      1.000472e+00
      AC      1.912785e-05
      FM      4.574273e-03
      UC      1.116059e-05
      DL      1.119147e-05
      DS      1.989047e-08
      DP      6.989970e-07
      ASTV    1.000472e+00
      MSTV    1.000472e+00
      ALTV    1.000472e+00
      Ds      1.000472e+00
      dtype: float64
```

```
[38]: #Performed EDA on the given dataset 'cardiotocographic.csv' by using Jupyter
      ↪ notebook with the help of manipulation libraries and visualization
      ↪ libraries, done the basic analysis on data, handled the null values with
      ↪ mean, removed the duplicates in the data, by visualization of histogram found
      ↪ some extreme values are there with the help of boxplot visualized the
      ↪ outliers in the data. with the help of heatmap found the correlation between
      ↪ the pair variables, and done statistical analysis, and used the standard
      ↪ scaler method standardized the data in the same scale. with standard data
      ↪ will expect good performance from the model.
```

```
[ ]:
```