Assignment No 2

```
import pandas as pd
import numpy as np
import io

from google.colab import files
uploaded=files.upload()
```

```
<IPython.core.display.HTML object>

Saving StudentsPerformance.csv to StudentsPerformance (1).csv
```

```
df=pd.read_csv(io.BytesIO(uploaded['StudentsPerformance.csv']))

print(df)
```

```
     gender race/ethnicity parental level of education         lunch  \
0       NaN        group B           bachelor's degree      standard
1    female        group C                some college      standard
2    female        group B             master's degree      standard
3      male        group A           associate's degree  free/reduced
4      male        group C                some college      standard
..      ...            ...                         ...           ...
995  female        group E             master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    test preparation course  math score  reading score  writing score
0                      none       120.0             72             74
1                 completed       150.0             90             88
2                      none         NaN             95             93
3                      none         NaN             57             44
4                      none         NaN             78             75
..                      ...         ...            ...            ...
995               completed        88.0             99             95
996                    none        62.0             55             55
997               completed        59.0             71             65
998               completed        68.0             78             77
999                    none        77.0             86             86

[1000 rows x 8 columns]
```

```
df.head()
```

```
   gender race/ethnicity parental level of education         lunch  \
0     NaN        group B           bachelor's degree      standard
1  female        group C                some college      standard
```

```
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none       120.0             72             74
1               completed       150.0             90             88
2                    none         NaN             95             93
3                    none         NaN             57             44
4                    none         NaN             78             75
```

```
print(df.isnull().sum().sum())
```

```
6
```

```
df.tail()
```

```
     gender race/ethnicity parental level of education         lunch  \
995  female        group E             master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    test preparation course  math score  reading score  writing score
995               completed        88.0             99             95
996                    none        62.0             55             55
997               completed        59.0             71             65
998               completed        68.0             78             77
999                    none        77.0             86             86
```

```
print(df.isnull().sum())
```

```
gender                         3
race/ethnicity                 0
parental level of education    0
lunch                          0
test preparation course        0
math score                     3
reading score                  0
writing score                  0
dtype: int64
```

```
print(df['gender'].isnull().sum())
```

```
3
```

```
from pandas.core.groupby.generic import NamedAgg
df['gender']=df['gender'].fillna(0)
```

```
new_df=df.dropna(axis=0)
```

```
print(new_df)
```

```
     gender race/ethnicity parental level of education         lunch  \
0         0        group B           bachelor's degree      standard
1    female        group C                some college      standard
5    female        group B           associate's degree      standard
6    female        group B                some college      standard
7         0        group B                some college  free/reduced
..      ...            ...                         ...           ...
995  female        group E             master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    test preparation course  math score  reading score  writing score
0                      none       120.0             72             74
1                 completed       150.0             90             88
5                      none        71.0             83             78
6                 completed        88.0             95             92
7                      none        40.0             43             39
..                      ...         ...            ...            ...
995               completed        88.0             99             95
996                    none        62.0             55             55
997               completed        59.0             71             65
998               completed        68.0             78             77
999                    none        77.0             86             86

[997 rows x 8 columns]
```

```
print(new_df['math score'])
```

```
0      120.0
1      150.0
5       71.0
6       88.0
7       40.0
       ...
995     88.0
996     62.0
997     59.0
998     68.0
999     77.0
Name: math score, Length: 997, dtype: float64
```

```
mean_value=new_df['reading score'].mean()
```

```
new_df['reading score'].fillna(value=mean_value,inplace=True)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/generic.py:6392:
SettingWithCopyWarning:
```

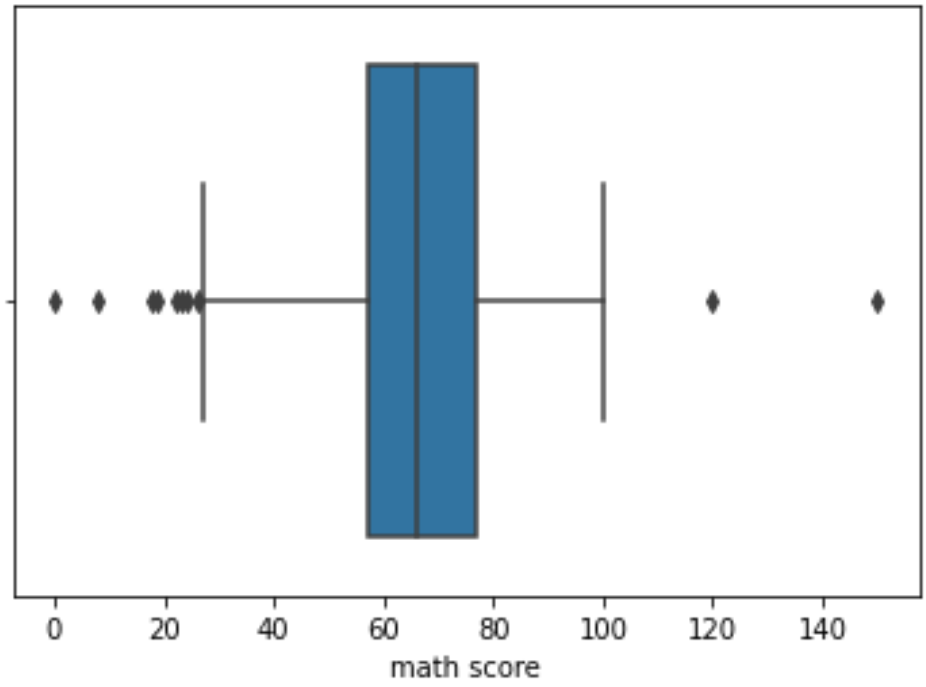
```
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return self._update_inplace(result)
```

```
print(new_df)
```

```
     gender race/ethnicity parental level of education         lunch  \
0         0        group B           bachelor's degree      standard
1    female        group C                some college      standard
5    female        group B           associate's degree      standard
6    female        group B                some college      standard
7         0        group B                some college  free/reduced
..      ...            ...                         ...           ...
995  female        group E             master's degree      standard
996    male        group C                 high school  free/reduced
997  female        group C                 high school  free/reduced
998  female        group D                some college      standard
999  female        group D                some college  free/reduced

    test preparation course  math score  reading score  writing score
0                      none       120.0             72             74
1                 completed       150.0             90             88
5                      none        71.0             83             78
6                 completed        88.0             95             92
7                      none        40.0             43             39
..                      ...         ...            ...            ...
995               completed        88.0             99             95
996                    none        62.0             55             55
997               completed        59.0             71             65
998               completed        68.0             78             77
999                    none        77.0             86             86

[997 rows x 8 columns]
```

```
import seaborn as sns
sns.boxplot(df['math score'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning

<matplotlib.axes._subplots.AxesSubplot at 0x7f6af42d3ed0>
```




```
print(np.where(df['math score']>100))
```

```
(array([0, 1]),)
```

```
df.gender.str.isdigit()
```

```
0        NaN
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Name: gender, Length: 1000, dtype: object
```

```
#Mean - missed value
df['math score']=df['math score'].replace(np.NaN,df['math score'].mean())
print(df['math score'])
```

```
0      120.000000
1      150.000000
2       66.203611
3       66.203611
4       66.203611
```

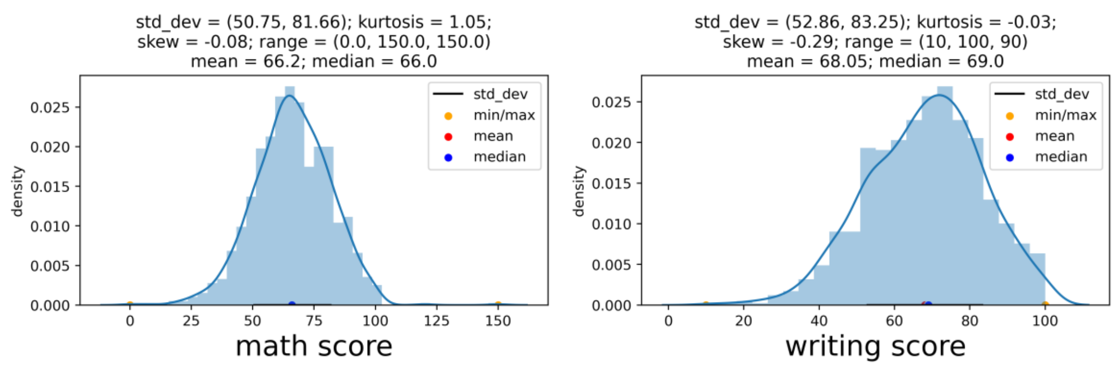
```
          ...
995     88.000000
996     62.000000
997     59.000000
998     68.000000
999     77.000000
Name: math score, Length: 1000, dtype: float64
```

```
import matplotlib.pyplot as plt
import statsmodels.api as sm

def UVA_numeric(data):
    var_group = data.columns
    size = len(var_group)
    plt.figure(figsize = (7*size,3), dpi = 400)

    #looping for each variable
    for j,i in enumerate(data.iloc[:,[-3,-1]]):


        # calculating descriptives of variable
        mini = data[i].min()
        maxi = data[i].max()
        ran = data[i].max()-data[i].min()
        mean = data[i].mean()
        median = data[i].median()
        st_dev = data[i].std()
        skew = data[i].skew()
        kurt = data[i].kurtosis()

        # calculating points of standard deviation
        points = mean-st_dev, mean+st_dev

        #Plotting the variable with every information
        plt.subplot(1,size,j+1)
        sns.distplot(data[i],hist=True, kde=True)

        sns.lineplot(points, [0,0], color = 'black', label = "std_dev")
        sns.scatterplot([mini,maxi], [0,0], color = 'orange', label =
"min/max")
        sns.scatterplot([mean], [0], color = 'red', label = "mean")
        sns.scatterplot([median], [0], color = 'blue', label = "median")
        plt.xlabel('{}'.format(i), fontsize = 20)
        plt.ylabel('density')
        plt.title('std_dev = {}; kurtosis = {};\nskew = {}; range = {}\nmean
= {}; median = {}'.format((round(points[0],2),round(points[1],2)),

round(kurt,2),

round(skew,2),
```

```
(round(mini,2),round(maxi,2),round(ran,2)),

round(mean,2),

round(median,2)))


UVA_numeric(df)
```

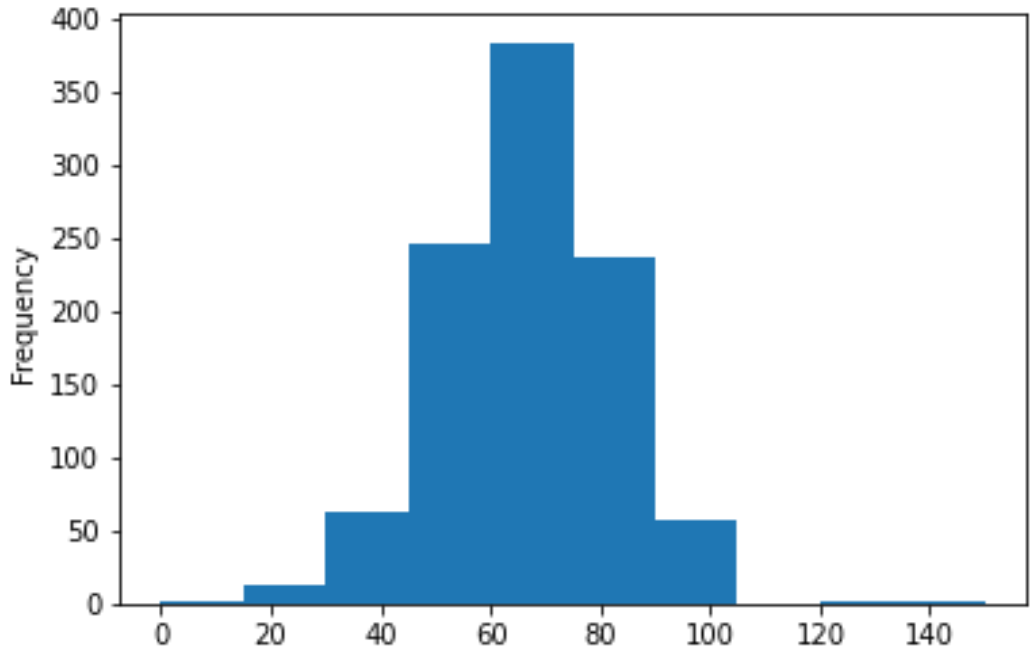
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or
```

```
misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variables as keyword args: x, y. From
version 0.12, the only valid positional argument will be `data`, and passing
other arguments without an explicit keyword will result in an error or
misinterpretation.
  FutureWarning
```




```
import matplotlib.pyplot as plt
df['math score'].plot(kind = 'hist')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6af467b250>
```




```
df['math_score'] = np.log10(df['math score'])
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/arraylike.py:364:
RuntimeWarning: divide by zero encountered in log10
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

```
scatterplot=new_df.plot.scatter(x='math score',y='writing score')
```