In [1]:
```
l=[15,24,6,37,50]
```

In [2]:
```
import pandas as pd
```

In [3]:
```
l=pd.DataFrame(l)
print(l.std())
```

```
0    17.472836
dtype: float64
```

In [4]:
```
data=pd.read_csv(r"C:\Users\DELL\Downloads\Feature Scaling Resource16994544271.csv")
```

In [5]:
```
data
```

Out[5]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [6]:
```
data.head()
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [7]:
```
data.isnull().any
```

Out[7]:
```
<bound method NDFrame._add_numeric_operations.<locals>.any of      Pregnancies  Glucose  BloodPressure  SkinThickness
Insulin    BMI  \
0          False    False          False          False    False  False
1          False    False          False          False    False  False
2          False    False          False          False    False  False
3          False    False          False          False    False  False
4          False    False          False          False    False  False
..           ...      ...            ...            ...      ...    ...
763        False    False          False          False    False  False
764        False    False          False          False    False  False
765        False    False          False          False    False  False
766        False    False          False          False    False  False
767        False    False          False          False    False  False

     DiabetesPedigreeFunction    Age  Outcome
0                       False  False    False
1                       False  False    False
2                       False  False    False
3                       False  False    False
4                       False  False    False
..                        ...    ...      ...
763                     False  False    False
764                     False  False    False
765                     False  False    False
766                     False  False    False
767                     False  False    False

[768 rows x 9 columns]>
```

In [8]:
```
data.dtypes
```

Out[8]:
```
Pregnancies                 int64
Glucose                     int64
BloodPressure               int64
SkinThickness               int64
Insulin                     int64
BMI                       float64
DiabetesPedigreeFunction  float64
Age                         int64
Outcome                     int64
dtype: object
```

In [9]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [10]:
```
data.describe()
```

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

In [11]:
```
from sklearn.preprocessing import MinMaxScaler
```

In [49]:
```
scaler1=MinMaxScaler()
```

In [50]:
```
data[["Pregnancies1","Glucose1"]]=scaler1.fit_transform(data[["Pregnancies","Glucose"]])
```

In [13]:
```
scaler1.fit_transform(data[["Pregnancies","Glucose"]])
```

Out[13]:
```
array([[0.35294118, 0.74371859],
       [0.05882353, 0.42713568],
       [0.47058824, 0.91959799],
       ...,
       [0.29411765, 0.6080402 ],
       [0.05882353, 0.63316583],
       [0.05882353, 0.46733668]])
```

In [14]:
```
data["Pregnancies"].min()
```

Out[14]:
```
0
```

In [15]:
```
data["Pregnancies"].max()
```

Out[15]:
```
17
```

In [16]:
```
data[data["Pregnancies"]>15]
```

Out[16]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
159	17	163	72	41	114	40.9	0.817	47	1

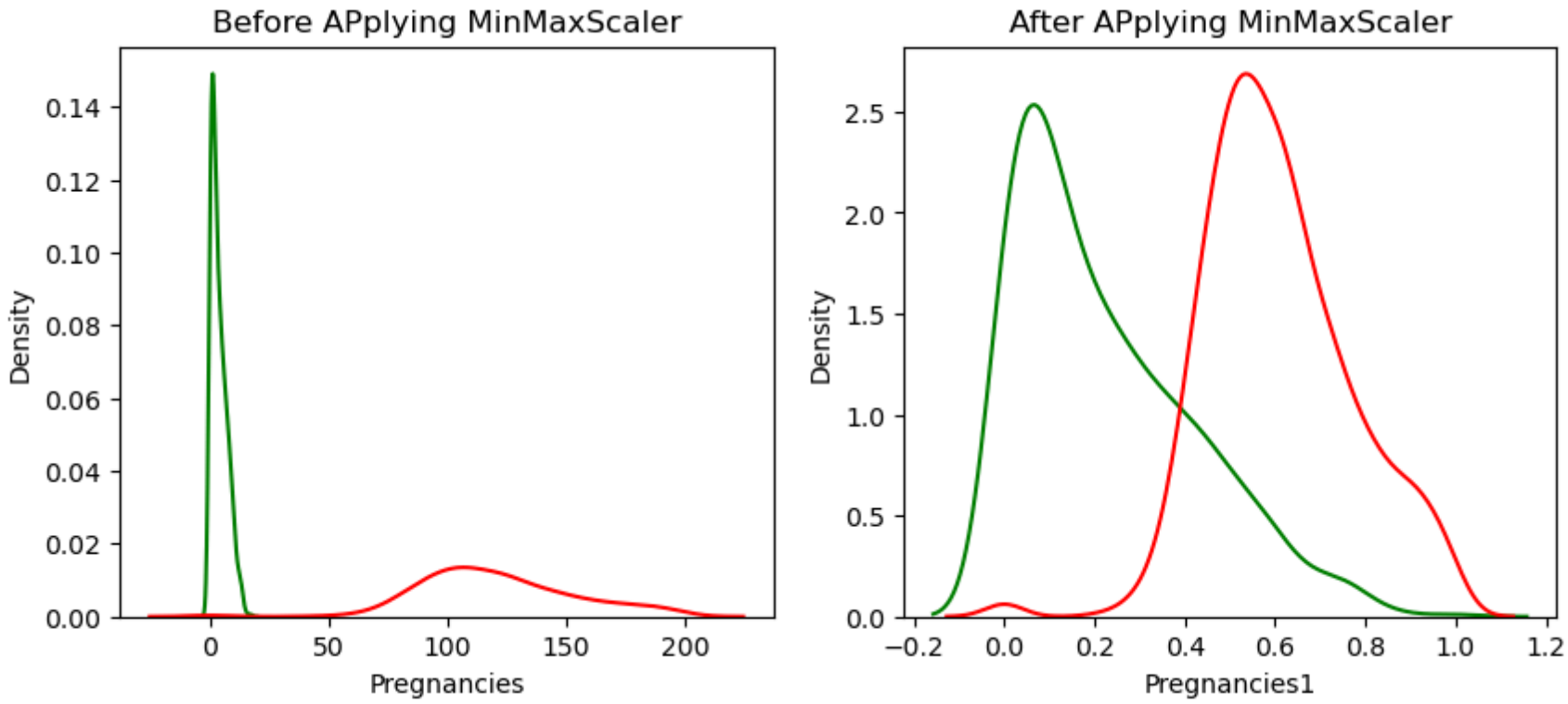
In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [51]:
```
fig,(ax1,ax2)=plt.subplots(ncols=2,figsize=(10,4))

ax1.set_title("Before APplying MinMaxScaler")
sns.kdeplot(data["Pregnancies"],ax=ax1,color="green")
sns.kdeplot(data["Glucose"],ax=ax1,color="red")

ax2.set_title("After APplying MinMaxScaler")
sns.kdeplot(data["Pregnancies1"],ax=ax2,color="green")
sns.kdeplot(data["Glucose1"],ax=ax2,color="red")
```

Out[51]:
```
<Axes: title={'center': 'After APplying MinMaxScaler'}, xlabel='Pregnancies1', ylabel='Density'>
```



In [21]:
```
from sklearn.preprocessing import StandardScaler
```

In [22]:
```
scaler2=StandardScaler()
```

In [26]:
```
data[["BloodPressure1","SkinThickness1"]]=scaler2.fit_transform(data[["BloodPressure","SkinThickness"]])
```

In [27]:
```
data[["BloodPressure","BloodPressure1"]]
```

Out[27]:

	BloodPressure	BloodPressure1
0	72	0.149641
1	66	-0.160546
2	64	-0.263941
3	66	-0.160546
4	40	-1.504687
...	...	...
763	76	0.356432
764	70	0.046245
765	72	0.149641
766	60	-0.470732
767	70	0.046245

768 rows × 2 columns

In [28]:
```
data["BloodPressure"].mean()
```

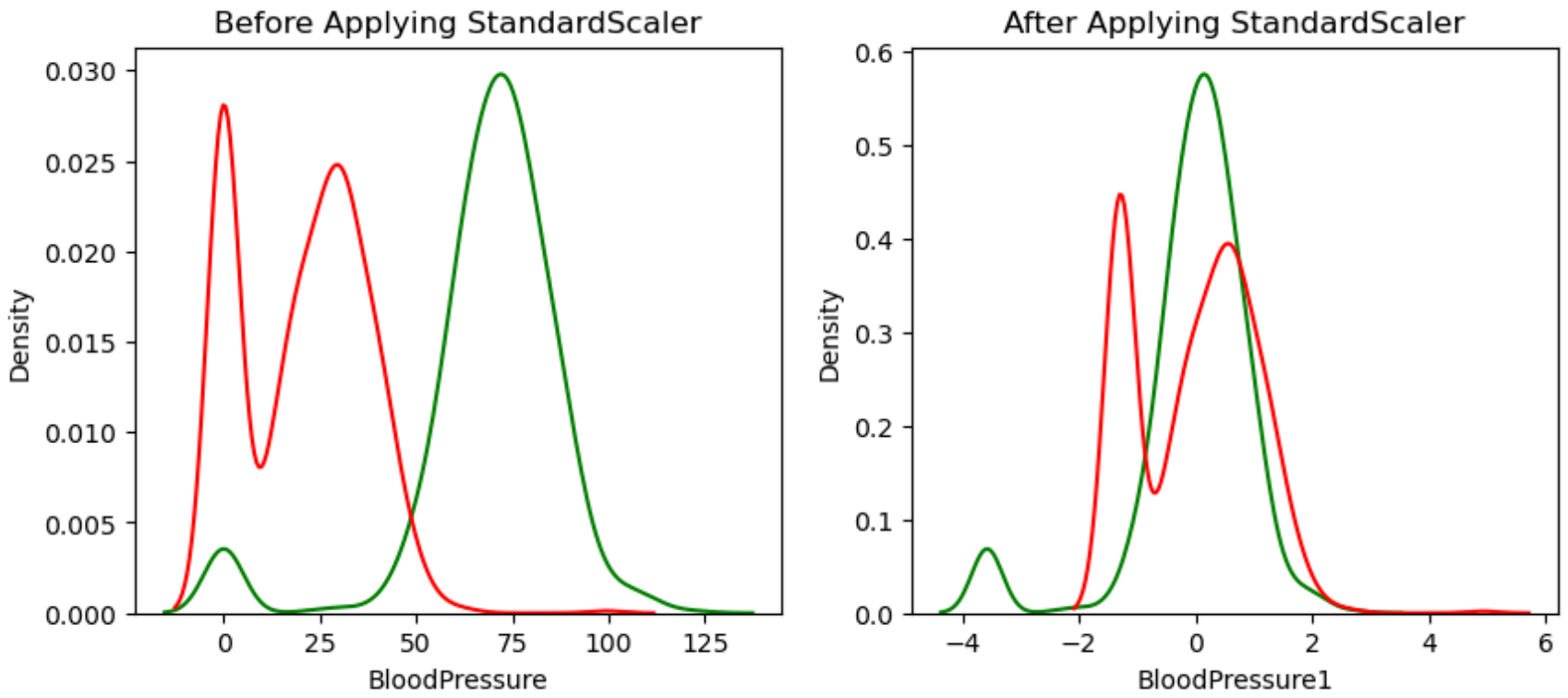
Out[28]:
```
69.10546875
```

In [32]:
```
fig,(ax1,ax2)=plt.subplots(ncols=2,figsize=(10,4))

ax1.set_title("Before Applying StandardScaler")
sns.kdeplot(data["BloodPressure"],ax=ax1,color="green")
sns.kdeplot(data["SkinThickness"],ax=ax1,color="red")
```

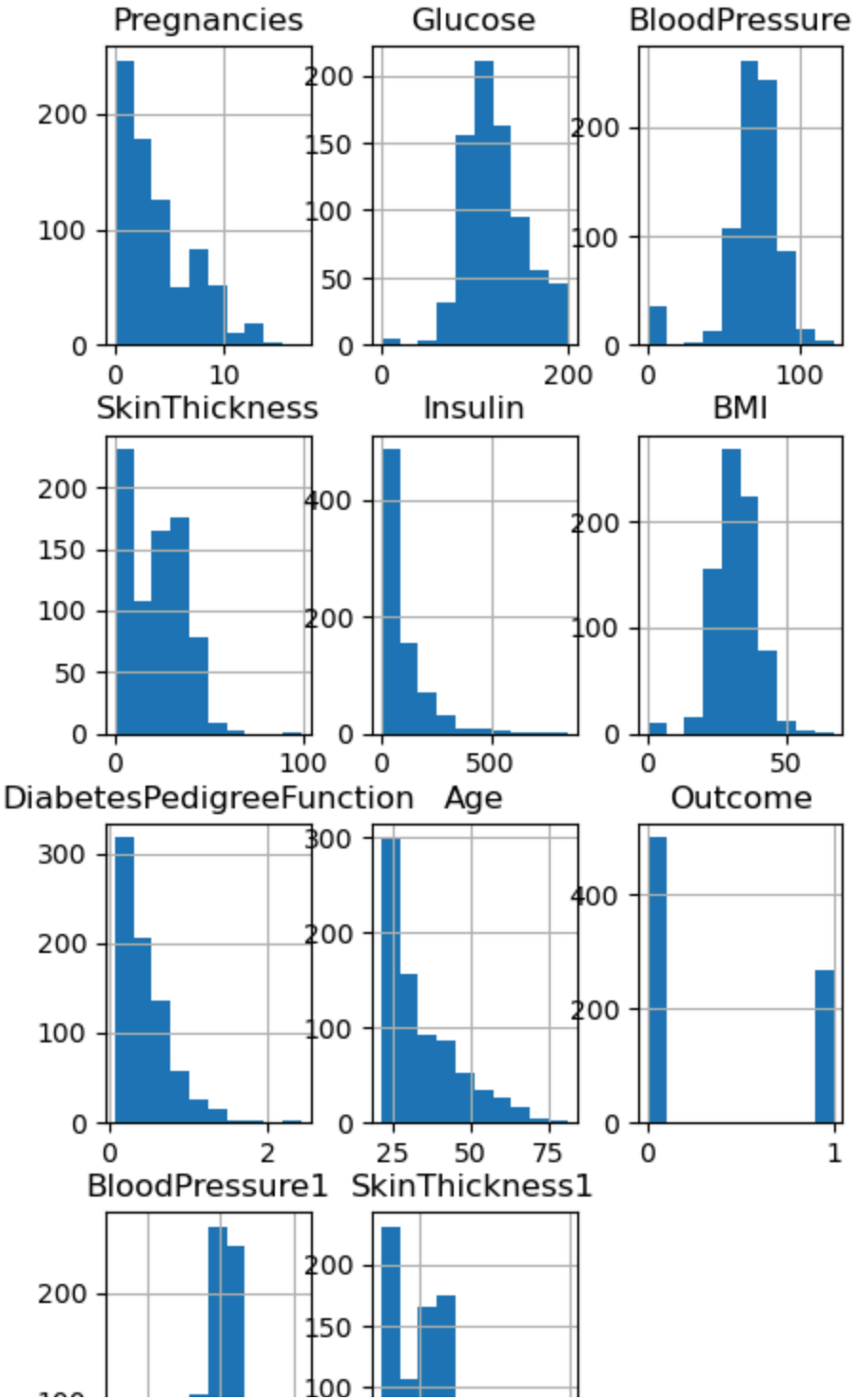
```
ax2.set_title("After Applying StandardScaler")
sns.kdeplot(data["BloodPressure1"],ax=ax2,color="green")
sns.kdeplot(data["SkinThickness1"],ax=ax2,color="red")
```

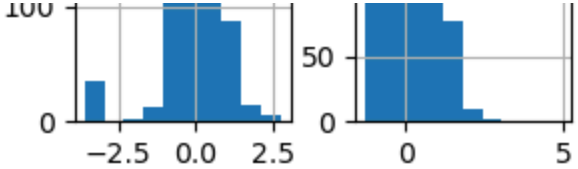
Out[32]:
```
<Axes: title={'center': 'After Applying StandardScaler'}, xlabel='BloodPressure1', ylabel='Density'>
```



In [36]:
```
data.hist(figsize=(5,10))
```

Out[36]:
```
array([[<Axes: title={'center': 'Pregnancies'}>,
        <Axes: title={'center': 'Glucose'}>,
        <Axes: title={'center': 'BloodPressure'}>],
       [<Axes: title={'center': 'SkinThickness'}>,
        <Axes: title={'center': 'Insulin'}>,
        <Axes: title={'center': 'BMI'}>],
       [<Axes: title={'center': 'DiabetesPedigreeFunction'}>,
        <Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'Outcome'}>],
       [<Axes: title={'center': 'BloodPressure1'}>,
        <Axes: title={'center': 'SkinThickness1'}>, <Axes: >]],
      dtype=object)
```


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
BloodPressure1
SkinThickness1



In [37]:
```
from sklearn.preprocessing import RobustScaler
```

In [38]:
```
scaler3=RobustScaler()
```

In [43]:
```
data[["Insulin1","BMI1"]]=scaler3.fit_transform(data[["Insulin","BMI"]])
```

In [44]:
```
data[["Insulin1","BMI1"]]
```

Out[44]:

	Insulin1	BMI1
0	-0.239686	0.172043
1	-0.239686	-0.580645
2	-0.239686	-0.935484
3	0.499018	-0.419355
4	1.080550	1.193548
...	...	...
763	1.174853	0.096774
764	-0.239686	0.516129
765	0.640472	-0.623656
766	-0.239686	-0.204301
767	-0.239686	-0.172043

768 rows × 2 columns

In [46]:
```
data["Insulin"].mean()
```

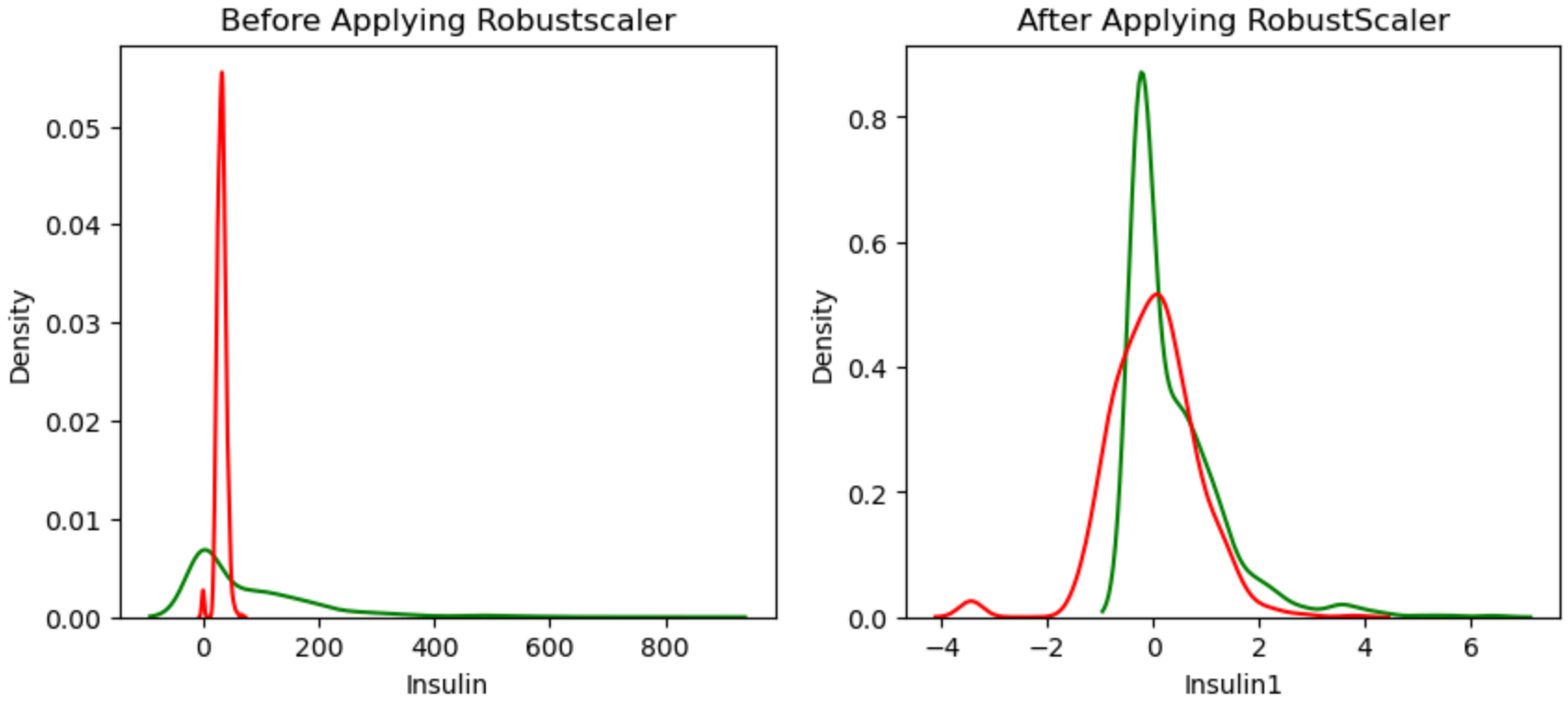
Out[46]:
```
79.79947916666667
```

In [47]:
```
fig,(ax1,ax2)=plt.subplots(ncols=2,figsize=(10,4))

ax1.set_title("Before Applying Robustscaler")
sns.kdeplot(data["Insulin"],ax=ax1,color="green")
sns.kdeplot(data["BMI"],ax=ax1,color="red")

ax2.set_title("After Applying RobustScaler")
sns.kdeplot(data["Insulin1"],ax=ax2,color="green")
sns.kdeplot(data["BMI1"],ax=ax2,color="red")
```

Out[47]:
```
<Axes: title={'center': 'After Applying RobustScaler'}, xlabel='Insulin1', ylabel='Density'>
```



In [ ]: