In [2]:
```
# import the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import warnings
warnings.filterwarnings("ignore")
pd.set_option("display.max_columns",None)
```

In [3]:
```
# Loading the Data Frame
df = pd.read_csv("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_De
```

In [4]:
```
df.sample(2)
```

Out[4]:

	Id	Diagnosis	Radius_Mean	Texture_Mean	Perimeter_Mean	Area_Mean	Smoothness_I
448	911685	B	11.49	14.59	73.99	404.9	0.1
123	867387	B	15.71	13.93	102.00	761.7	0.0

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 535 entries, 0 to 534
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Id                       535 non-null    int64  
 1   Diagnosis                535 non-null    object 
 2   Radius_Mean              535 non-null    float64
 3   Texture_Mean             535 non-null    float64
 4   Perimeter_Mean           535 non-null    float64
 5   Area_Mean                535 non-null    float64
 6   Smoothness_Mean          535 non-null    float64
 7   Compactness_Mean         535 non-null    float64
 8   Concavity_Mean           535 non-null    float64
 9   Concave Points_Mean      535 non-null    float64
 10  Symmetry_Mean            535 non-null    float64
 11  Fractal_Dimension_Mean   535 non-null    float64
 12  Radius_Se                535 non-null    float64
 13  Texture_Se               535 non-null    float64
 14  Perimeter_Se             535 non-null    float64
 15  Area_Se                  535 non-null    float64
 16  Smoothness_Se            535 non-null    float64
 17  Compactness_Se           535 non-null    float64
 18  Concavity_Se             535 non-null    float64
 19  Concave Points_Se        535 non-null    float64
 20  Symmetry_Se              535 non-null    float64
 21  Fractal_Dimension_Se     535 non-null    float64
 22  Radius_Worst             535 non-null    float64
 23  Texture_Worst            535 non-null    float64
 24  Perimeter_Worst          535 non-null    float64
 25  Area_Worst               535 non-null    float64
 26  Smoothness_Worst         535 non-null    float64
 27  Compactness_Worst        535 non-null    float64
 28  Concavity_Worst          535 non-null    float64
 29  Concave Points_Worst     535 non-null    float64
 30  Symmetry_Worst           535 non-null    float64
 31  Fractal_Dimension_Worst  535 non-null    float64
dtypes: float64(30), int64(1), object(1)
memory usage: 133.9+ KB
```

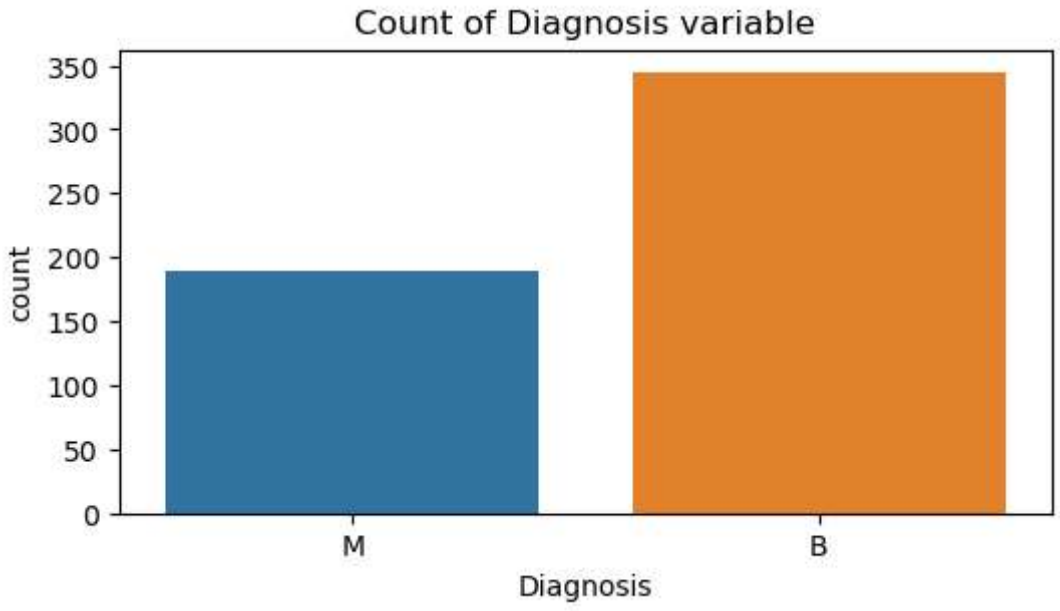
In [6]:
```
df.describe(
)
```

Out[6]:

	Id	Radius_Mean	Texture_Mean	Perimeter_Mean	Area_Mean	Smoothness_Me
count	5.350000e+02	535.000000	535.000000	535.000000	535.000000	535.0000
mean	2.800529e+07	14.009338	19.153065	91.027121	639.084673	0.0955
std	1.172244e+08	3.268515	4.160548	22.439502	313.983225	0.0132
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.0526
25%	8.690515e+05	11.725000	16.070000	75.265000	421.950000	0.0858
50%	9.060240e+05	13.300000	18.770000	86.100000	546.300000	0.0949
75%	8.812860e+06	15.635000	21.680000	102.750000	759.950000	0.1044
max	9.113205e+08	25.220000	33.810000	171.500000	1878.000000	0.1398

Univariate Analysis

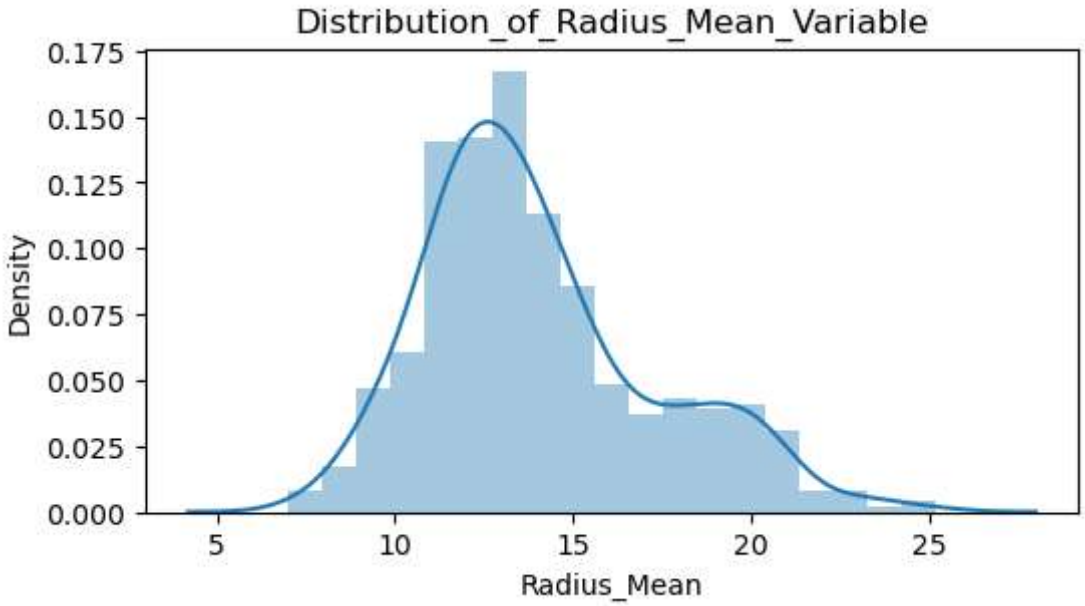
In [7]:
```
# univariate analysis on diagnosis_variable
plt.figure(figsize=(6,3))
sns.countplot(data = df, x = "Diagnosis")
plt.title("Count of Diagnosis variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above plot:

1. in our dataset we have more of benign groups than those of malignants.

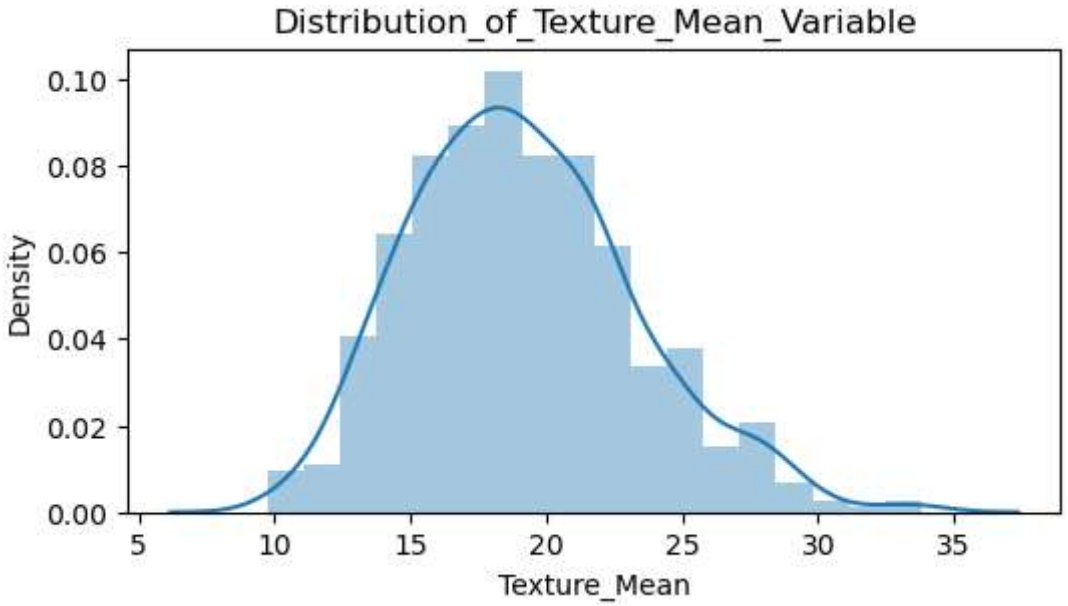
In [8]:
```
# Univariate analysis on radius mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Radius_Mean)
plt.title("Distribution_of_Radius_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above distplot:

1. slightly more records have a "Radius RMean" value less than that of the mean.
2. most of the records have a "Radius_Mean" value between 10 and 20

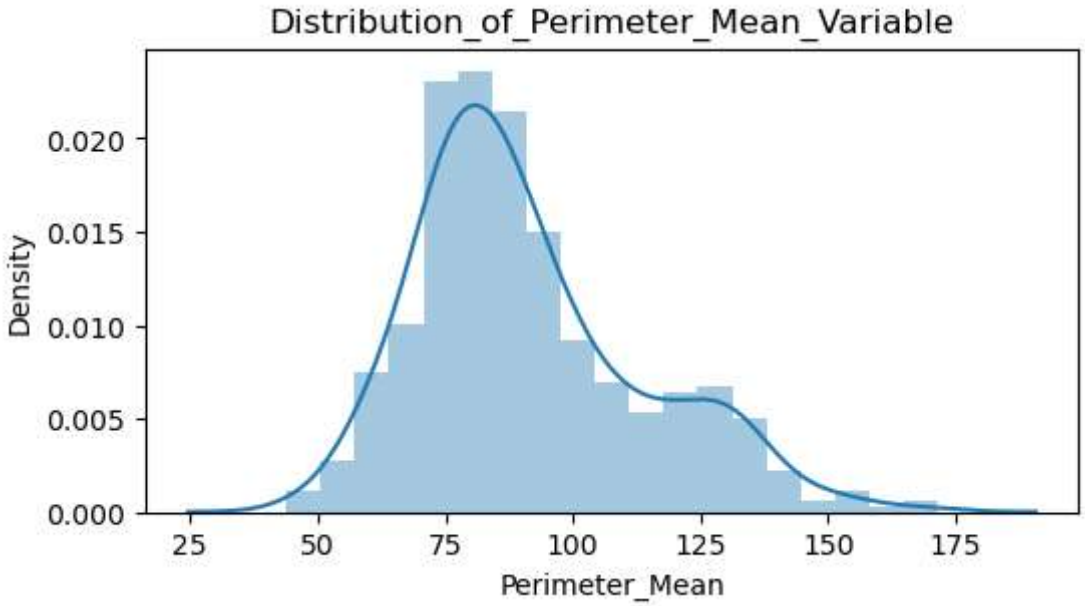
In [9]:
```
# Univariate analysis on Texture mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Texture_Mean)
plt.title("Distribution_of_Texture_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the texture mean variable distribution plot above:

1. The distrubution plot shows a bell-shaped curve, which is characteristic of a normal distribution.
2. The distribution appears to be fairly symmetric around the center.
3. The data points are spread between approximately 5 and 35, with the highest density around the center.

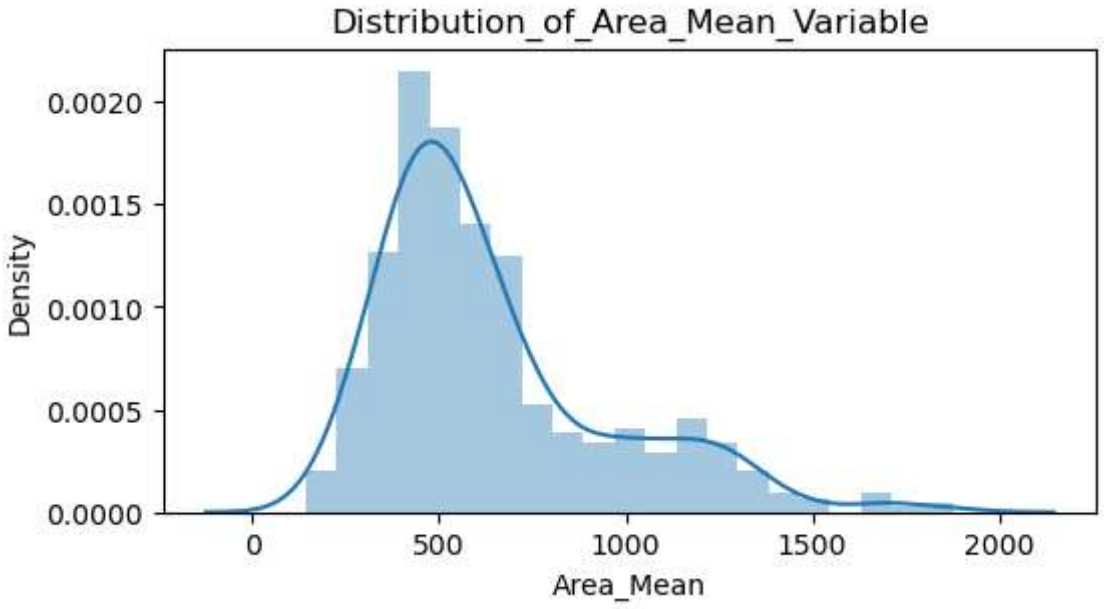
In [10]:
```
# Univariate analysis on Perimeter mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Perimeter_Mean)
plt.title("Distribution_of_Perimeter_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above distribution plot of Perimeter mean variable

1. The histogram shows a somewhat bell-shaped curve but with some deviations.
2. The distribution appears to be slightly skewed to the right (positive skewness), as the tail on the right side is longer than the left.
3. The data points are spread between approximately 25 and 175, with the highest density around the center.

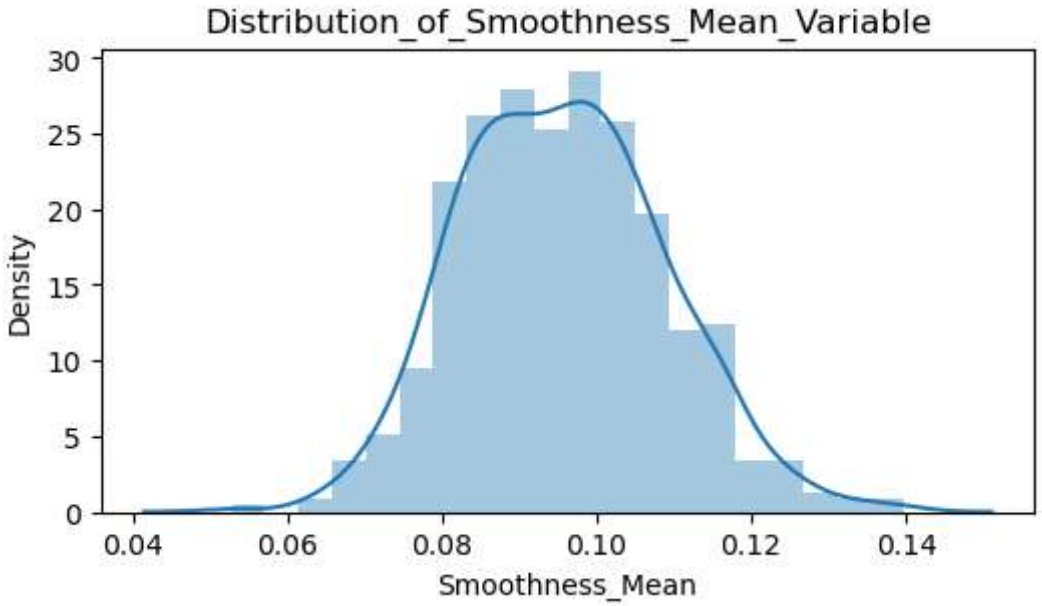
In [11]:
```
# Univariate analysis on Area mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Area_Mean)
plt.title("Distribution_of_Area_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above Distribution Plot of Area_Mean variable

1. The histogram shows a right-skewed distribution rather than a bell-shaped curve.
2. The distribution is not symmetric; it has a longer tail on the right side.
3. The data points are spread between approximately 0 and 2000, with the highest density around lower values.

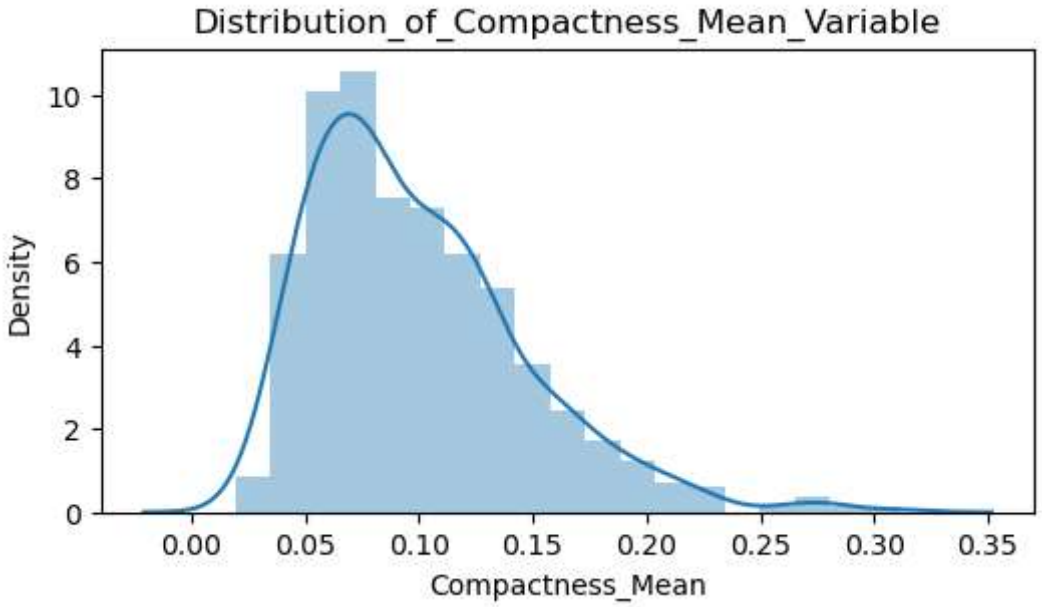
In [12]:
```
# Univariate analysis on Smoothness mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Smoothness_Mean)
plt.title("Distribution_of_Smoothness_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above smoothness mean distribution plot

1. The distribution appears to be approximately normal, centered around a mean smoothness value of approximately 0.10.
2. The smoothness values range roughly from 0.04 to 0.14, indicating the spread or variability in the data.

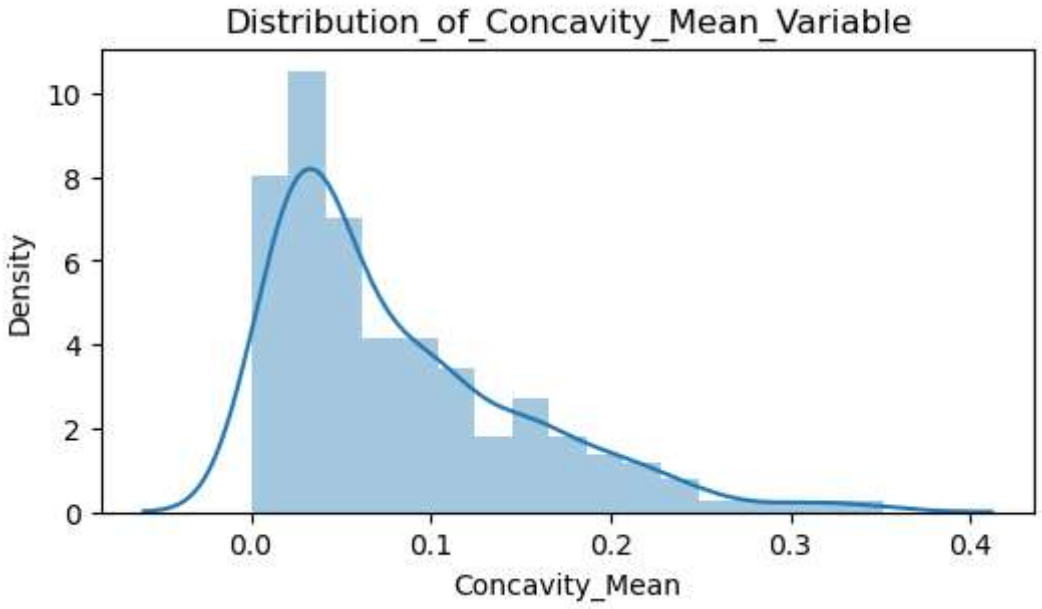
In [13]:
```
# Univariate analysis on Compactness mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Compactness_Mean)
plt.title("Distribution_of_Compactness_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the distribution Compactness distribution plot above:

1. The distribution has a peak around the 0.05 to 0.07 range, suggesting that the most frequent values for compactness mean are in this range.
2. The compactness values range from close to 0 to about 0.35, indicating a wide variability in the data.
3. The distribution is right-skewed (positively skewed), with a longer tail extending to the right. This indicates that while most values are clustered around the lower end, there are a few higher values stretching the distribution to the right.

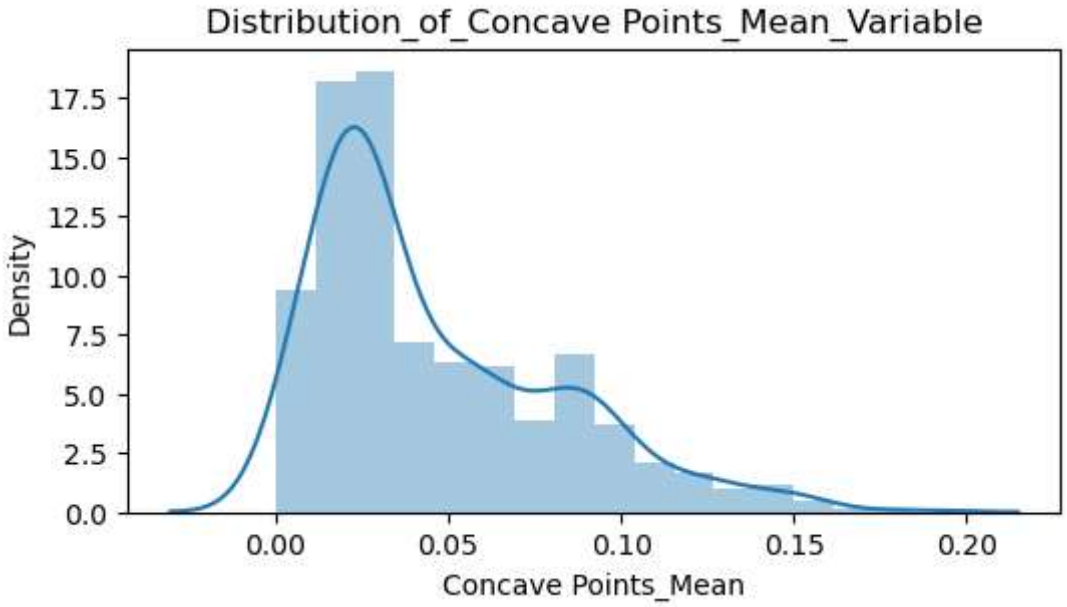
In [14]:
```
# Univariate analysis on Concavity mean variable
plt.figure(figsize=(6,3))
sns.distplot(df.Concavity_Mean)
plt.title("Distribution_of_Concavity_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



As we can see from the above concavity mean distribution plot:

1. The distribution has a peak around 0.02 to 0.04, suggesting that the most frequent values for concavity mean are in this range.
2. The concavity values range from close to 0 to about 0.4, indicating a wide variability in the data.
3. The distribution is right-skewed (positively skewed), with a longer tail extending to the right. This indicates that while most values are clustered around the lower end, there are a few higher values stretching the distribution to the right.

In [16]:
```
# Univariate analysis on Concave Points_Mean variable
plt.figure(figsize=(6,3))
sns.distplot(df["Concave Points_Mean"])
plt.title("Distribution_of_Concave Points_Mean_Variable")
plt.savefig("C:\\Users\\yozil\\Desktop\\My projects\\3.0 Breast_Cancer_Detecti
plt.show()
```



In [8]:
```
df.columns
```

Out[8]:
```
Index(['Id', 'Diagnosis', 'Radius_Mean', 'Texture_Mean', 'Perimeter_Mean',
       'Area_Mean', 'Smoothness_Mean', 'Compactness_Mean', 'Concavity_Mean',
       'Concave Points_Mean', 'Symmetry_Mean', 'Fractal_Dimension_Mean',
       'Radius_Se', 'Texture_Se', 'Perimeter_Se', 'Area_Se', 'Smoothness_Se',
       'Compactness_Se', 'Concavity_Se', 'Concave Points_Se', 'Symmetry_Se',
       'Fractal_Dimension_Se', 'Radius_Worst', 'Texture_Worst',
       'Perimeter_Worst', 'Area_Worst', 'Smoothness_Worst',
       'Compactness_Worst', 'Concavity_Worst', 'Concave Points_Worst',
       'Symmetry_Worst', 'Fractal_Dimension_Worst'],
      dtype='object')
```

In [ ]: