Handling Missing Values

July 10, 2024

2 methods

-Imputation -Dropping

[1]:
```
#mporting the libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
dataset = pd.read_csv('placement_Dataset.csv')
dataset.head()
```

[2]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	\
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	
1	2	M	79.33	Central	78.33	Others	Science	77.48	
2	3	M	65.00	Central	68.00	Central	Arts	64.00	
3	4	M	56.00	Central	52.00	Central	Science	52.00	
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	

	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

[3]:
```
dataset.shape
```

[3]:
```
(215, 15)
```

[7]:
```
dataset.isnull().sum()
```

[7]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
```

```
degree_p           0
degree_t           0
workex             0
etest_p            0
specialisation     0
mba_p              0
status             0
salary            67
dtype: int64
```

Using imputation method. They are all called central tendencies. -Mean -sum the number and divide by tht total numbers. -Mode -the most occurence numbers in a dataset. -Median -arrange the values in accessending orders and take the midle value.

when to use mean- first find how the data is distributed in the column before filling the missing value. if it is skew were data is distriputed in one area of the chart as shown below you can't use mean values to replace the missing values because it has outliers and it will increase the mean values. In that cases you can use mode or median as a replacement for the missing values. If it is a normal distributed values were values are distributed in all magnitude in such cases you can use mean.

[13]:
```
fig, plot = plt.subplots(figsize=(10,10))
sns.distplot(dataset.salary);
```

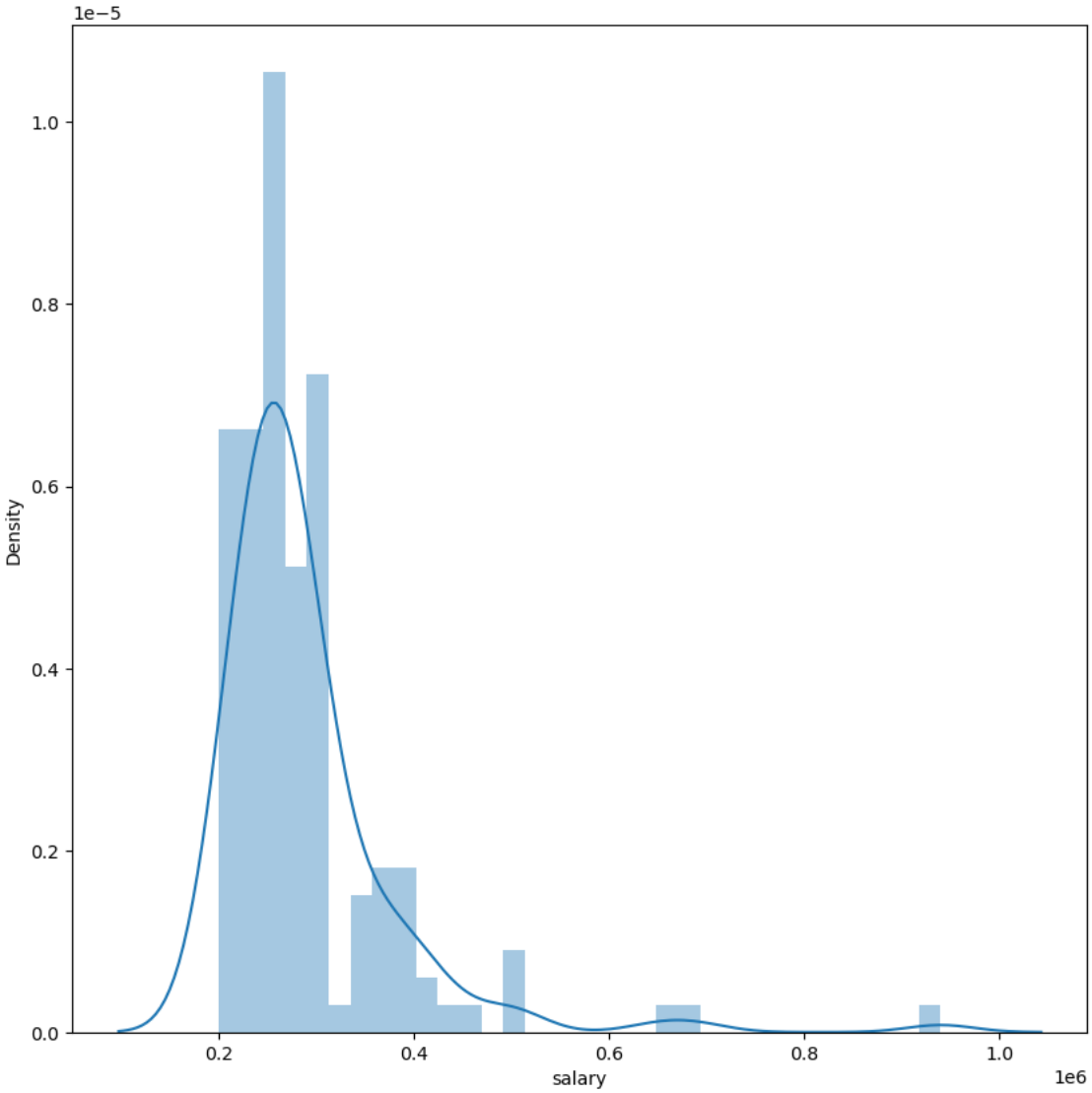
```
C:\Users\DONATUS\AppData\Local\Temp\ipykernel_19212\2041372052.py:2:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset.salary);
```



Replacing the missing values with Median Values

[16]:
```
dataset['salary'].fillna(dataset['salary'].median(), inplace = True)
```

[18]:
```
dataset.isnull().sum()
```

[18]:
```
sl_no           0
gender          0
ssc_p           0
ssc_b           0
hsc_p           0
hsc_b           0
hsc_s           0
```

```
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

Filling missing values with mean see exmple below

dataset['salary'].fillna(dataset['salary'].mean(), inplace = True)

How to drop missing values.

[22]:
```
salary_dataset = pd.read_csv('placement_Dataset.csv')
salary_dataset.shape
```

[22]:
```
(215, 15)
```

[24]:
```
salary_dataset.isnull().sum()
```

[24]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary           67
dtype: int64
```

[25]:
```
salary_dataset=salary_dataset.dropna(how='any')
```

[26]:
```
salary_dataset.isnull().sum()
```

[26]:
```
sl_no             0
gender            0
ssc_p             0
ssc_b             0
hsc_p             0
```

```
hsc_b             0
hsc_s             0
degree_p          0
degree_t          0
workex            0
etest_p           0
specialisation    0
mba_p             0
status            0
salary            0
dtype: int64
```

[27]:
```
salary_dataset.shape
```

[27]:
```
(148, 15)
```

[ ]:
```

```