```
# cca----> complete case analysis

# df----> missing data ----> filter missing data---->
# new----> filtered_columns_in_which_we_have_missing_data

# new_df---> missing_data_drop

# architecture ----> histogram--->
# past column in which we have missing data.
# updated column in which we have no missing data

# if past data architecture is overlap to new data architecture
# it means we can drop missing data
# if past data architecture is not overlap to new data architecture
# it means we can not drop missing data .we will fill missing data
```

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

```
df=pd.read_csv("/content/dsjob - dsjob1.csv")
df.head(2)
```

	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level	major_di
0	32403	city_41	0.827	Male	Has relevent experience	Full time course	Graduate	
1	9858	city_103	0.920	Female	Has relevent experience	no_enrollment	Graduate	

```
df.isnull().sum()*100
```

	0
enrollee_id	0
city	0
city_development_index	0
gender	50800
relevent_experience	0
enrolled_university	3100
education_level	5200
major_discipline	31200
experience	500
company_size	62200
company_type	63400
last_new_job	4000
training_hours	0

dtype: int64

```
cols= [var for var in df.columns if df[var].isnull().mean()< 0.05 and df[var].isnull().mean()>0]
cols
```

```
['enrolled_university', 'education_level', 'experience', 'last_new_job']
```

```
df[cols].sample(5)
```

	enrolled_university	education_level	experience	last_new_job
1295	no_enrollment	Graduate	11	1
213	no_enrollment	High School	10	never
1911	Part time course	Graduate	15	>4
2001	Full time course	Graduate	2	2
1100	no_enrollment	Graduate	10	1

```
df['education_level'].value_counts()
```

education_level	count
Graduate	1269
Masters	496
High School	222
Phd	54
Primary School	36

dtype: int64

```
len(df[cols].dropna())/ len(df)
```

```
0.9478628464067638
```

```
new_df= df[cols].dropna()
df.shape, new_df.shape
```

```
((2129, 13), (2018, 4))
```

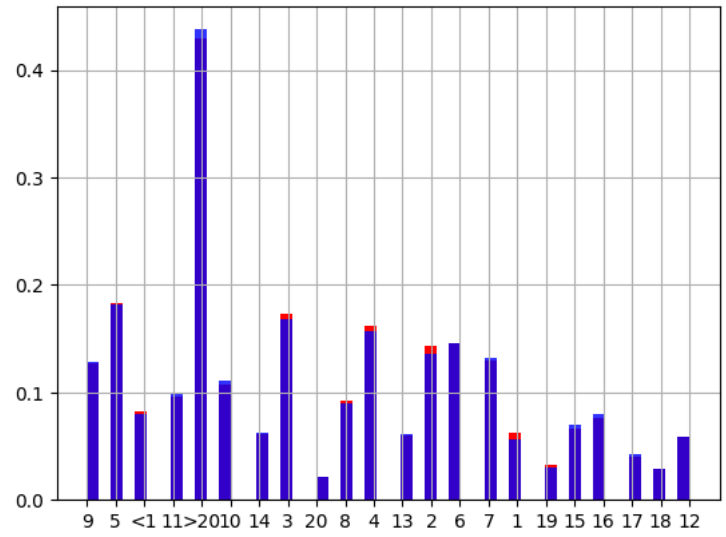
```
import matplotlib.pyplot as plt
```

```
fig= plt.figure()
ax=fig.add_subplot(111)

# original data
df['experience'].hist(bins=50,ax=ax,density=True,color='red')

# data after cca, the argument alpha makes the color transparent , so we can
# see the overlap of the 2 distributions
new_df['experience'].hist(bins=50,ax=ax,density=True,color='blue',alpha=0.8)
```

```
<Axes: >
```




```
df=pd.read_csv('/content/covid_toy - covid_toy.csv')
df.head(2)
```

	age	gender	fever	cough	city	has_covid
0	60	Male	103.0	Mild	Kolkata	No
1	27	Male	100.0	Mild	Delhi	Yes

```
df.isnull().sum()*100
```

	0
age	0
gender	0
fever	1000
cough	0
city	0
has_covid	0

dtype: int64

```
cols= [var for var in df.columns if df[var].isnull().mean()< 20 and df[var].isnull().mean()>0]
cols
```

```
['fever']
```

```
df[cols].sample(5)
```

	fever
32	101.0
89	103.0
87	101.0
43	99.0
41	NaN

```
df['fever'].value_counts()
```

fever	count
101.0	17
98.0	17
104.0	14
100.0	13
99.0	10
102.0	10
103.0	9

dtype: int64

```
len(df[cols].dropna())/ len(df)
```

```
0.9
```

```
new_df= df[cols].dropna()
df.shape, new_df.shape
```

```
((100, 6), (90, 1))
```

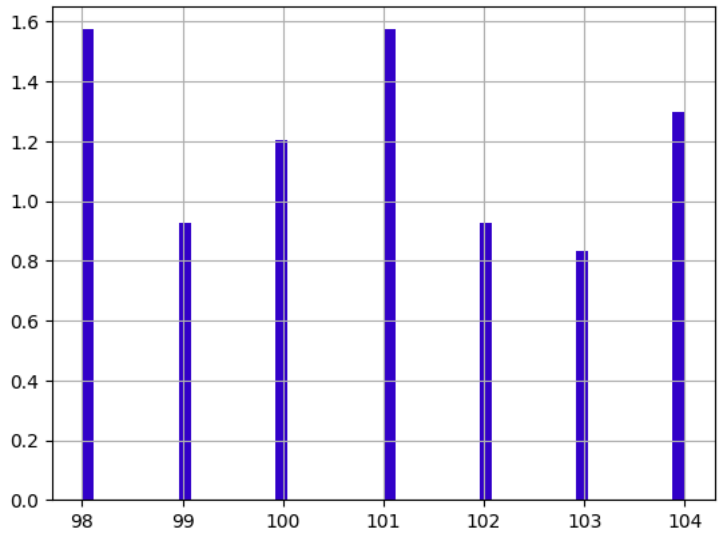
```
import matplotlib.pyplot as plt
```

```
fig= plt.figure()
ax=fig.add_subplot(111)

# original data
df['fever'].hist(bins=50,ax=ax,density=True, color='red')

# data after cca, the argument alpha makes the color transparent , so we can
# see the overlap of the 2 distributions
new_df['fever'].hist(bins=50,ax=ax,density=True, color='blue',alpha=0.8)
```

```
<Axes: >
```




```
new_df['fever'].isnull().sum()
```

```
np.int64(0)
```

```
df=pd.read_csv('/content/titanic - titanic.csv')
df.head(2)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S

```
df.isnull().sum()*100
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	8600
SibSp	0
Parch	0
Ticket	0
Fare	100
Cabin	32700
Embarked	0

dtype: int64

```
cols= [var for var in df.columns if df[var].isnull().mean()<25 and df[var].isnull().mean()>0]
cols
```

```
['Age', 'Fare', 'Cabin']
```

```
df[cols].sample(5)
```

	Age	Fare	Cabin
88	NaN	7.750	NaN
14	47.0	61.175	E31
52	20.0	23.000	NaN
120	12.0	15.750	NaN
231	21.0	26.550	NaN

```
df['Age'].value_counts()
```

Age	count
21.0	17
24.0	17
22.0	16
30.0	15
18.0	13
...	...
44.0	1
5.0	1
51.0	1
3.0	1
38.5	1

79 rows × 1 columns

dtype: int64

```
len(df[cols].dropna())/ len(df)
```

```
0.20813397129186603
```

```
new_df= df[cols].dropna()
df.shape, new_df.shape
```

```
((418, 12), (87, 3))
```

```
import matplotlib.pyplot as plt
```

```
fig= plt.figure()
ax=fig.add_subplot(111)

# original data
df['Age'].hist(bins=50,ax=ax,density=True, color='red')

# data after cca, the argument alpha makes the color transparent , so we can
# see the overlap of the 2 distributions
new_df['Age'].hist(bins=50,ax=ax,density=True, color='green',alpha=0.8)
```

```
<Axes: >
```