```
import pandas as pd
import seaborn as sns
```

```
df=pd.read_csv('insurance.csv')
```

```
df
```

	age	weight	height	income_lpa	smoker	city	occupation	insurance_premium_category
0	67	119.8	1.56	2.92000	False	Jaipur	retired	High
1	36	101.1	1.83	34.28000	False	Chennai	freelancer	Low
2	39	56.8	1.64	36.64000	False	Indore	freelancer	Low
3	22	109.4	1.55	3.34000	True	Mumbai	student	Medium
4	69	62.2	1.60	3.94000	True	Indore	retired	High
...	...	...	...	...	...	...	...	...
95	36	52.8	1.57	19.64000	False	Indore	business_owner	Low
96	26	113.8	1.54	34.01000	False	Delhi	private_job	Low
97	52	60.8	1.80	44.86000	False	Hyderabad	freelancer	Low
98	27	101.1	1.82	28.30000	False	Kolkata	business_owner	Low
99	40	70.0	1.59	28.16664	True	Bangalore	government_job	Low

100 rows × 8 columns

Next steps: Generate code with df | New interactive sheet

```
df.head()
```

	age	weight	height	income_lpa	smoker	city	occupation	insurance_premium_category
0	67	119.8	1.56	2.92	False	Jaipur	retired	High
1	36	101.1	1.83	34.28	False	Chennai	freelancer	Low
2	39	56.8	1.64	36.64	False	Indore	freelancer	Low
3	22	109.4	1.55	3.34	True	Mumbai	student	Medium
4	69	62.2	1.60	3.94	True	Indore	retired	High

Next steps: Generate code with df | New interactive sheet

```
df.tail()
```

	age	weight	height	income_lpa	smoker	city	occupation	insurance_premium_category
95	36	52.8	1.57	19.64000	False	Indore	business_owner	Low
96	26	113.8	1.54	34.01000	False	Delhi	private_job	Low
97	52	60.8	1.80	44.86000	False	Hyderabad	freelancer	Low
98	27	101.1	1.82	28.30000	False	Kolkata	business_owner	Low
99	40	70.0	1.59	28.16664	True	Bangalore	government_job	Low

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        100 non-null    int64
 1   weight                     100 non-null    float64
 2   height                     100 non-null    float64
 3   income_lpa                 100 non-null    float64
 4   smoker                     100 non-null    bool
```

```
 5   city                        100 non-null    object
 6   occupation                  100 non-null    object
 7   insurance_premium_category  100 non-null    object
dtypes: bool(1), float64(3), int64(1), object(3)
memory usage: 5.7+ KB
```

```
df.describe()
```

	age	weight	height	income_lpa
count	100.000000	100.000000	100.000000	100.000000
mean	47.180000	83.894000	1.713200	18.400600
std	16.649312	21.020278	0.110205	16.067465
min	18.000000	51.100000	1.500000	0.530000
25%	34.750000	63.650000	1.610000	2.897500
50%	47.000000	82.300000	1.730000	14.122583
75%	61.000000	101.300000	1.810000	30.162500
max	75.000000	119.800000	1.900000	50.000000

```
df.shape
```

```
(100, 8)
```

```
df.size
```

```
800
```

```
df["age"].mean()
```

```
np.float64(47.18)
```

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

	0
age	0
weight	0
height	0
income_lpa	0
smoker	0
city	0
occupation	0
insurance_premium_category	0

dtype: int64

```
df.nunique()
```

	0
age	54
weight	94
height	38
income_lpa	89
smoker	2
city	15
occupation	7
insurance_premium_category	3

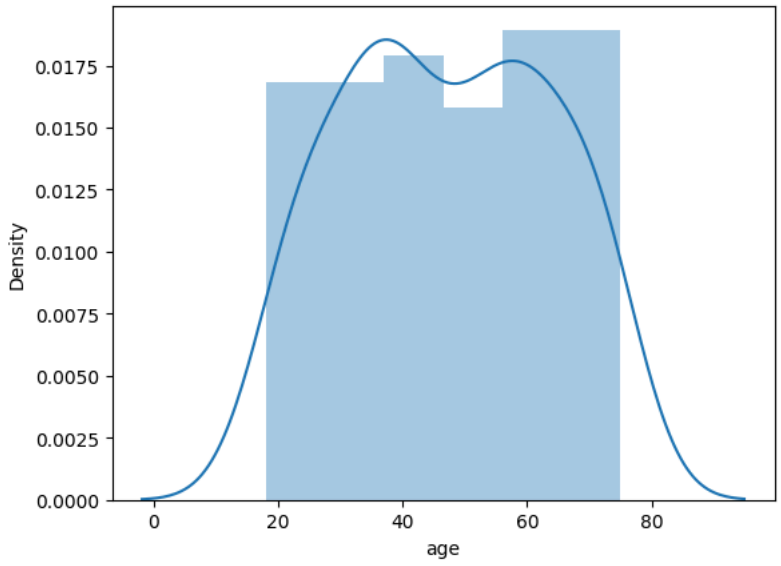
dtype: int64

```
df["age"].value_counts()
```

age	count
69	4
61	4
59	4
35	4
39	3
52	3
24	3
23	3
72	3
38	3
47	3
58	3
36	3
41	3
53	2
67	2

```
sns.distplot(df["age"])
```

```
/tmp/ipython-input-316555093.py:1: 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(df["age"])
<Axes: xlabel='age', ylabel='Density'>
```

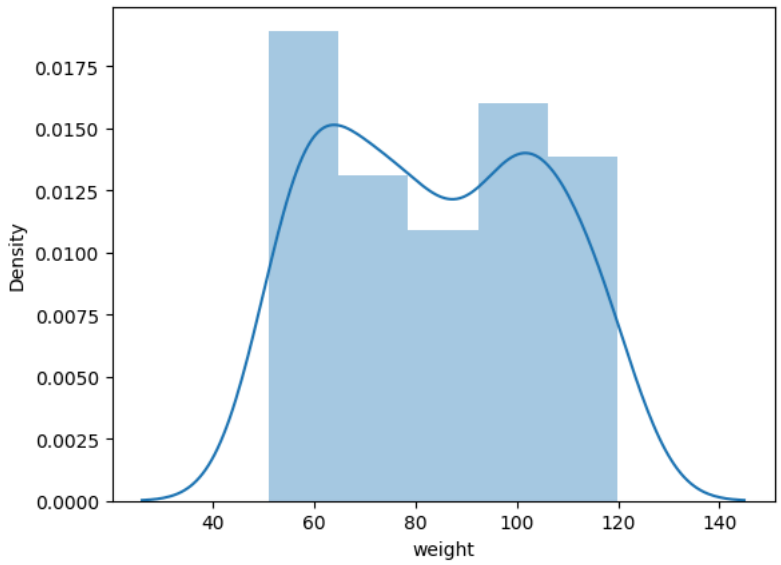



```
sns.distplot(df["weight"])
```

66	1
65	1
19	1
29	1
44	1
60	1
64	1
74	1
51	1
45	1
32	1
71	1
20	1

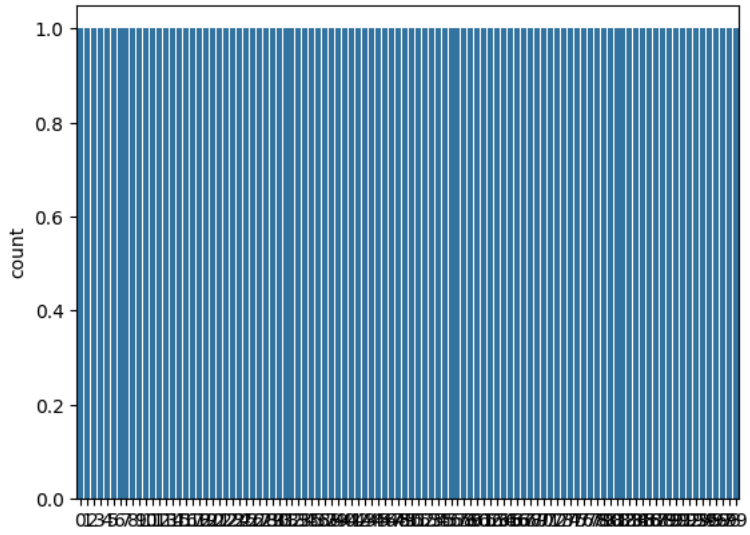
```
70    1
63    1
62    1
56    1
30    1
54    1
47    1
```

```
/tmp/ipython-input-4031182476.py:1: 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(df["weight"])
<Axes: xlabel='weight', ylabel='Density'>
```



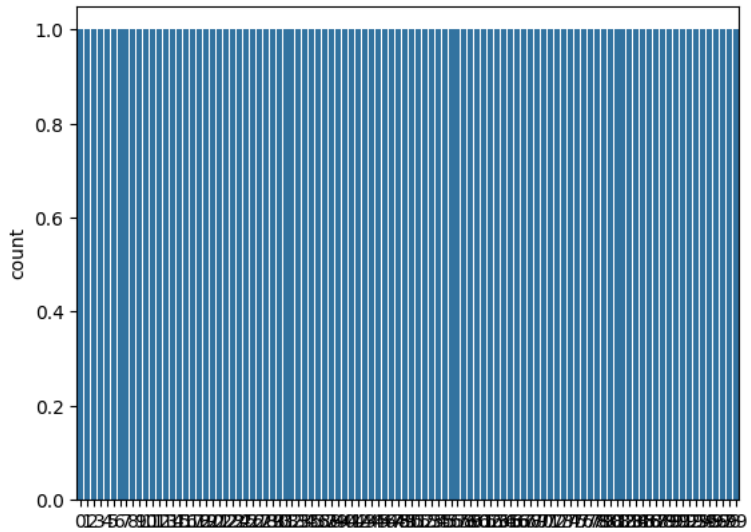

```
sns.countplot(df["age"])
```

```
<Axes: ylabel='count'>
```



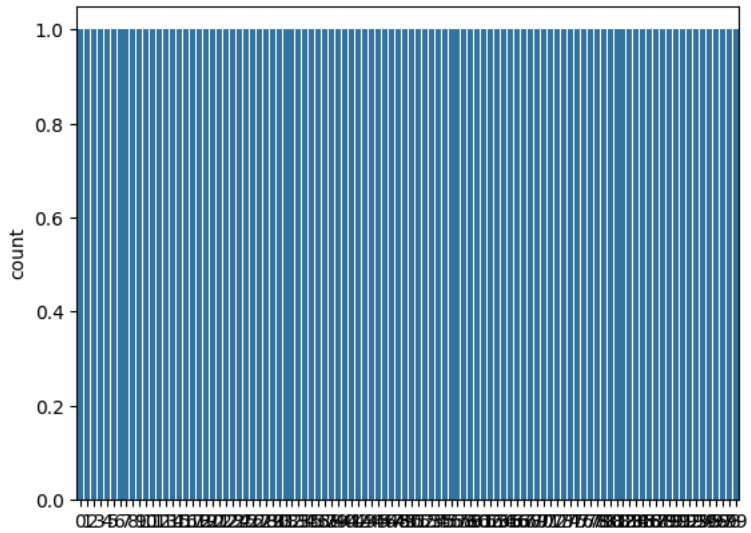

```
sns.countplot(df["weight"])
```

```
<Axes: ylabel='count'>
```



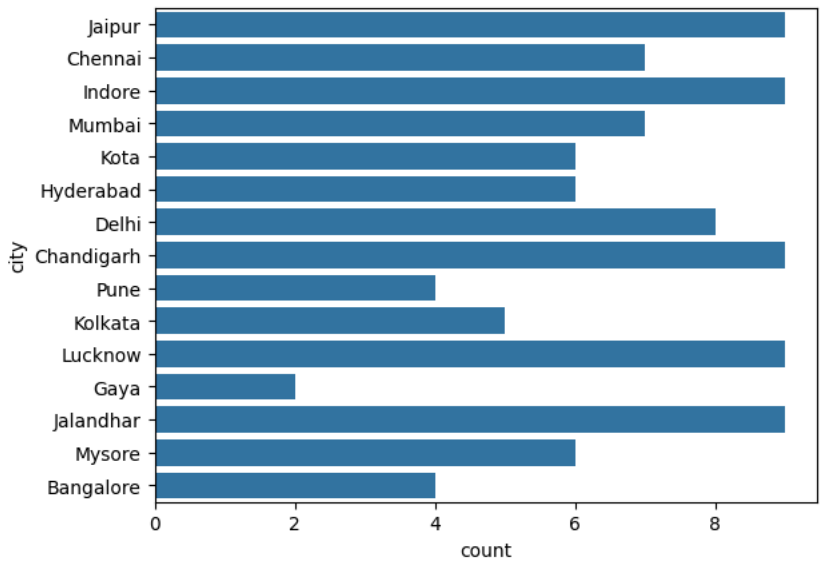

```
sns.countplot(df["income_lpa"])
```

```
<Axes: ylabel='count'>
```



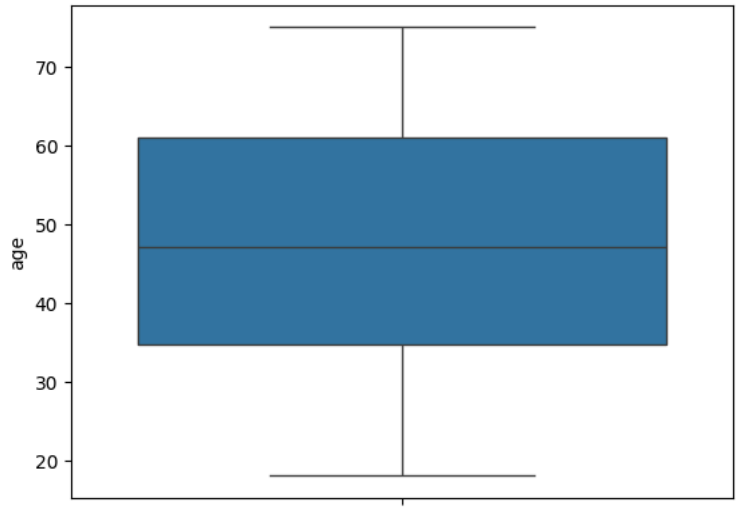

```
sns.countplot(df["city"])
```

```
<Axes: xlabel='count', ylabel='city'>
```



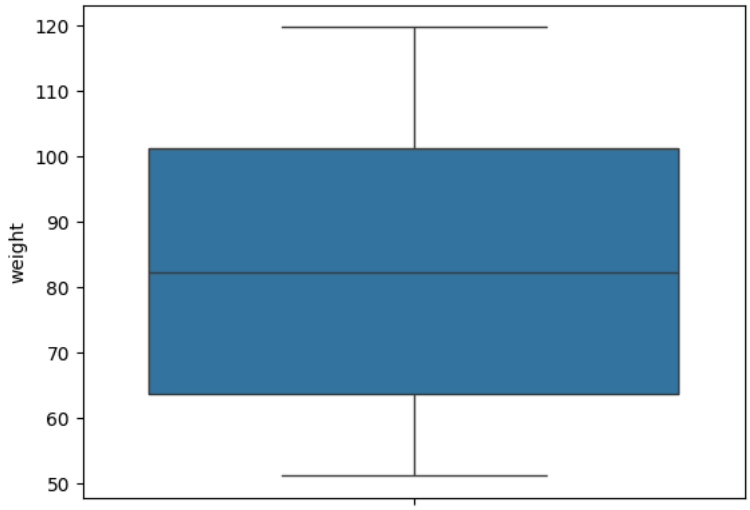
```
sns.boxplot(df["age"])
```

```
<Axes: ylabel='age'>
```



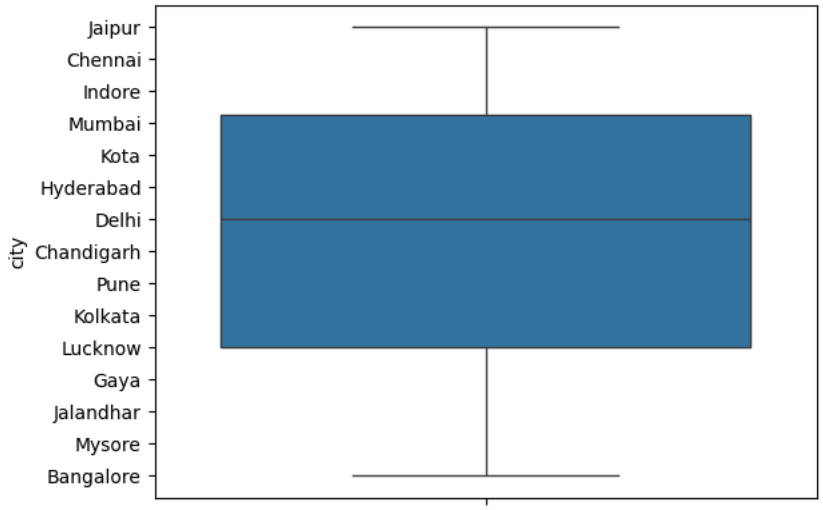

```
sns.boxplot(df["weight"])
```

```
<Axes: ylabel='weight'>
```



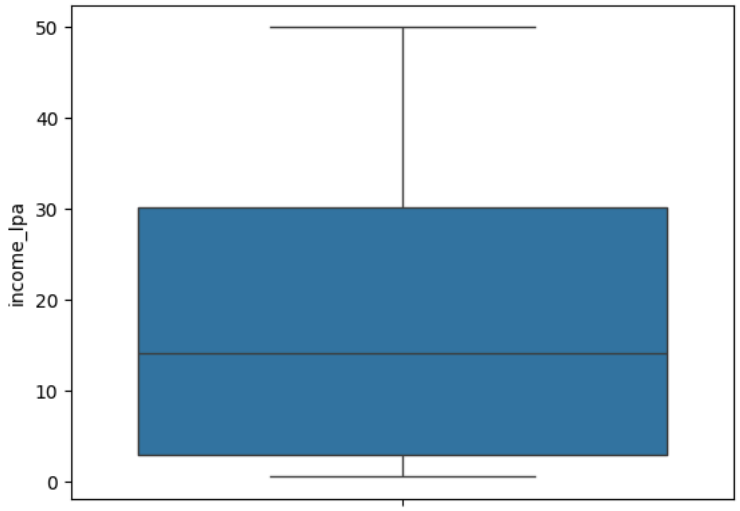

```
sns.boxplot(df["city"])
```

```
<Axes: ylabel='city'>
```



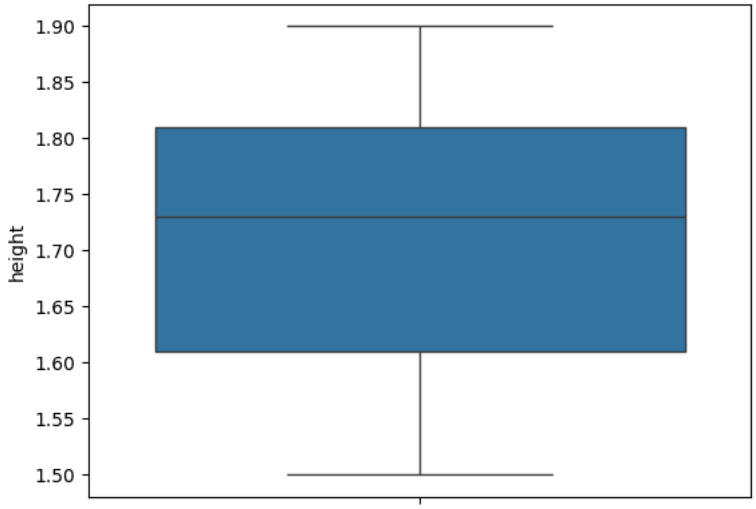
```
sns.boxplot(df["income_lpa"])
```

```
<Axes: ylabel='income_lpa'>
```




```
sns.boxplot(df["height"])
```

```
<Axes: ylabel='height'>
```




```
sns.boxplot(x="age",y="weight",data=df)
```

```
<Axes: xlabel='age', ylabel='weight'>
```