

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("/content/Bangalore.csv")
```

```
df.head()
```

	BHK	sector	price	price_persqft	SuperArea	floor	Furnishing	facing	Car_Parking	Bathroom	Balcony	overlooking	city
0	3	Panathur	2.25	13966.0	1611.0	10+	Unfurnished	East	Available	2	1	Garden/Park	Bangalore
1	3	Panathur	2.13	12909.0	1650.0	10+	Unfurnished	East	Available	3	2	Garden/Park	Bangalore
2	3	Thanisandra	1.37	9648.0	1420.0	8	Unfurnished	North - East	Available	2	1	Garden/Park, Pool	Bangalore
3	3	Panathur	1.62	11000.0	1473.0	5	Unfurnished	North - East	Available	2	1	Garden/Park, Main Road	Bangalore

Next steps:

[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

```
df.describe()
```

	price	price_persqft	SuperArea
count	1440.000000	1439.000000	1439.000000
mean	1.705747	9639.820014	1671.038916
std	1.620426	4840.776656	1051.490276
min	0.085000	2591.000000	270.000000
25%	0.779500	6364.000000	1140.000000
50%	1.210000	8774.000000	1400.000000
75%	2.000000	11665.500000	1862.000000
max	18.900000	43077.000000	14000.000000

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1440 entries, 0 to 1439
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   BHK              1440 non-null   object
1   sector           1440 non-null   object
2   price            1440 non-null   float64
3   price_persqft    1439 non-null   float64
4   SuperArea        1439 non-null   float64
5   floor            1360 non-null   object
6   Furnishing       1424 non-null   object
7   facing           873 non-null    object
8   Car_Parking      1440 non-null   object
9   Bathroom         1431 non-null   object
10  Balcony          1011 non-null   object
11  overlooking       763 non-null    object
12  city             1440 non-null   object
dtypes: float64(3), object(10)
memory usage: 146.4+ KB
```


```
df.dropna()
```

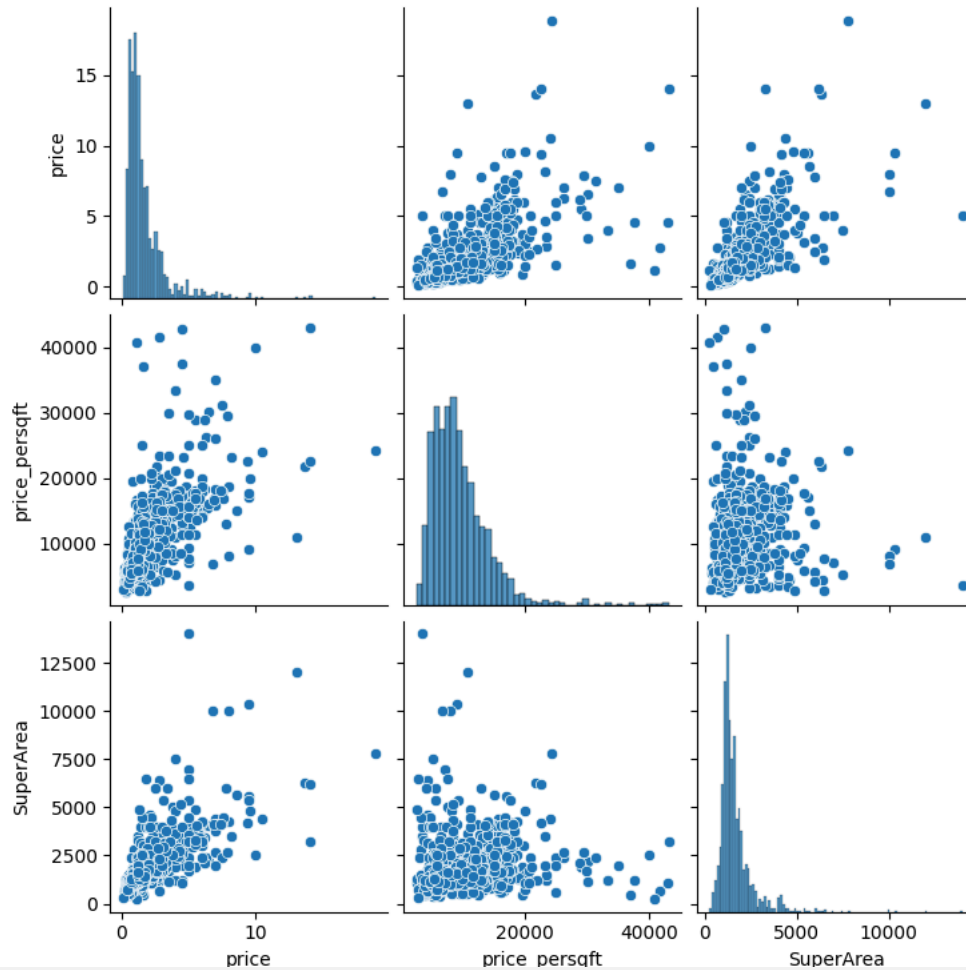
	BHK	sector	price	price_persqft	SuperArea	floor	Furnishing	facing	Car_Parking	Bathroom	Balcony	overlooking	city
0	3	Panathur	2.25	13966.0	1611.0	10+	Unfurnished	East	Available	2	1	Garden/Park	Bangalore
1	3	Panathur	2.13	12909.0	1650.0	10+	Unfurnished	East	Available	3	2	Garden/Park	Bangalore
2	3	Thanisandra	1.37	9648.0	1420.0	8	Unfurnished	North - East	Available	2	1	Garden/Park, Pool	Bangalore
3	3	Panathur	1.62	11000.0	1473.0	5	Unfurnished	North - East	Available	2	1	Garden/Park, Main Road	Bangalore
4	4	Whitefield	3.93	15023.0	2616.0	10+	Unfurnished	East	Available	3	2	Garden/Park	Bangalore
...
1433	2	JP Nagar	1.20	8219.0	1460.0	4	Unfurnished	East	May be not	2	4	Main Road	Bangalore
1435	3	others	1.74	10578.0	1645.0	2	Unfurnished	North - East	May be not	3	2	Garden/Park	Bangalore
1436	3	others	1.16	5859.0	1980.0	7	Unfurnished	East	Available	3	1	Garden/Park, Pool, Main Road	Bangalore

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

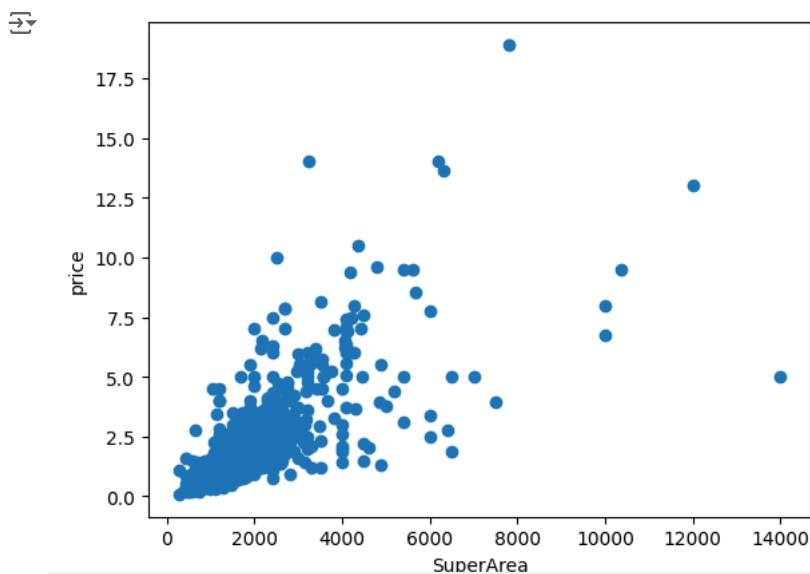
	0
BHK	0
sector	0
price	0
price_persqft	1
SuperArea	1
floor	80
Furnishing	16
facing	567
Car_Parking	0
Bathroom	9
Balcony	429
overlooking	677
city	0

```
sns.pairplot(df)
```

 <seaborn.axisgrid.PairGrid at 0x78982e2e8880>

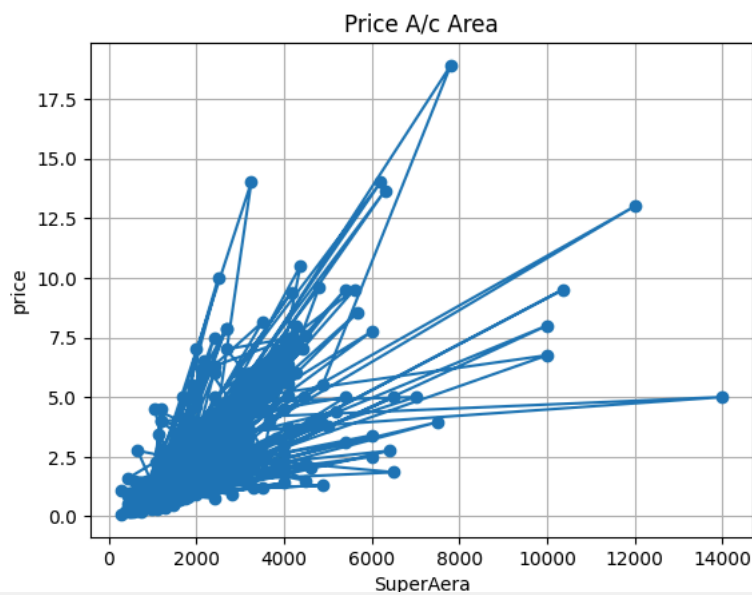


```
plt.scatter(df["SuperArea"],df["price"])
plt.xlabel("SuperArea")
plt.ylabel("price")
plt.show()
```



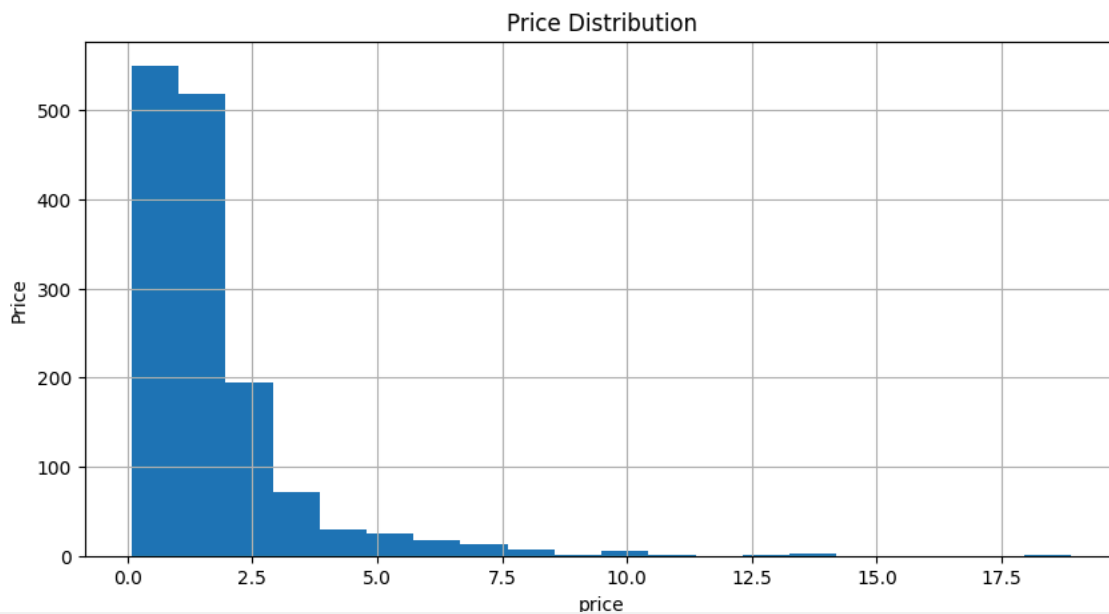
```
# Line Plot
```

```
plt.plot(df['SuperArea'], df['price'], marker='o')
plt.xlabel('SuperAera')
plt.ylabel('price')
plt.title('Price A/c Area')
plt.grid(True)
plt.show()
```




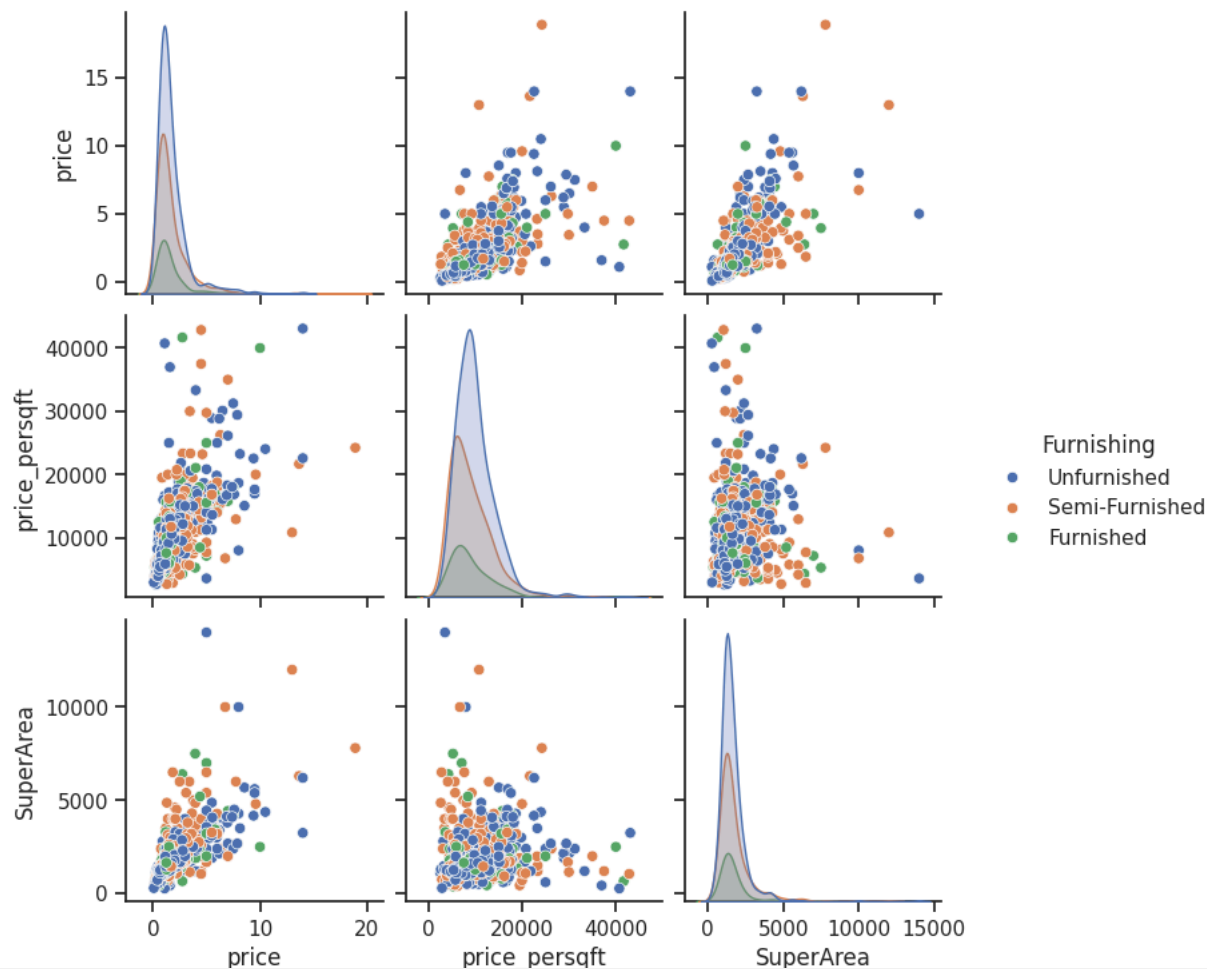
```
plt.figure(figsize=(10,5)) # set figure size
```

```
df["price"].hist(bins=20)
plt.xlabel("price")
plt.ylabel("Price")
plt.title("Price Distribution")
plt.grid(True)
plt.show()
```




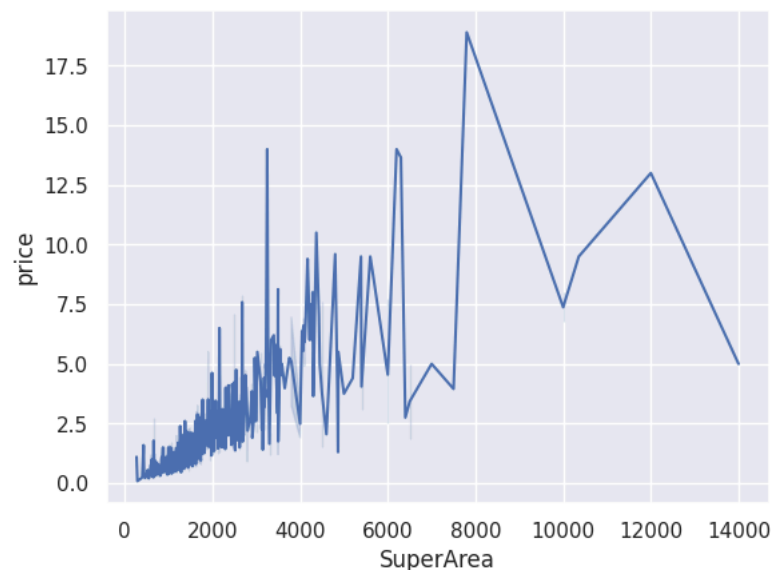
```
sns.set_theme(style="ticks")
sns.pairplot(df,hue="Furnishing")
```

 <seaborn.axisgrid.PairGrid at 0x78980ed88760>



```
sns.set_theme(style="darkgrid")
sns.lineplot(x="SuperArea",y="price",data=df)
```

 <Axes: xlabel='SuperArea', ylabel='price'>



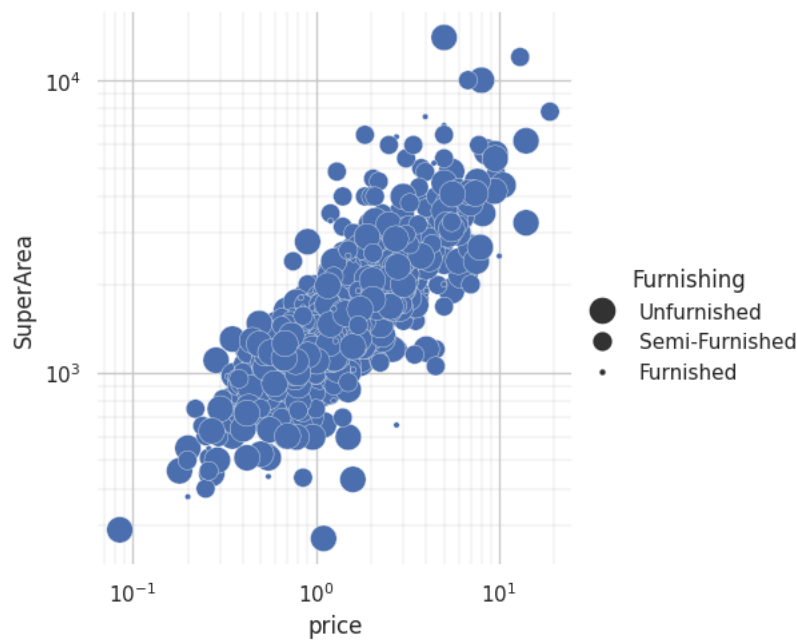
```
sns.set_theme(style="whitegrid")
```

```
cmap = sns.cubehelix_palette(rot=-.2, as_cmap=True)
g = sns.relplot(
    data=df,
    x="price", y="SuperArea",
    size="Furnishing",
    palette=cmap, sizes=(10, 200),
)
```

```
g.set(xscale="log", yscale="log")
```

```
g.ax.xaxis.grid(True, "minor", linewidth=.25)
g.ax.yaxis.grid(True, "minor", linewidth=.25)
g.despine(left=True, bottom=True)
```

```
<ipython-input-23-0e4834a0cd30>:4: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
g = sns.relplot(
<seaborn.axisgrid.FacetGrid at 0x78980b6125f0>
```




```
# Horizontal boxplot with observations
```

```
df.columns
```

```
<ipython-input-23-0e4834a0cd30>:4: UserWarning: Ignoring `palette` because no `hue` variable has been assigned.
Index(['BHK', 'sector', 'price', 'price_persqft', 'SuperArea', 'floor',
      'Furnishing', 'facing', 'Car_Parking', 'Bathroom', 'Balcony',
      'overlooking', 'city'],
      dtype='object')
```

```
sns.boxplot(
    df, x="Furnishing", y="BHK",
    whis=[0, 100], width=.6, palette="vlag"
)
sns.stripplot(df, x="Furnishing", y="BHK", size=4, color=".3")

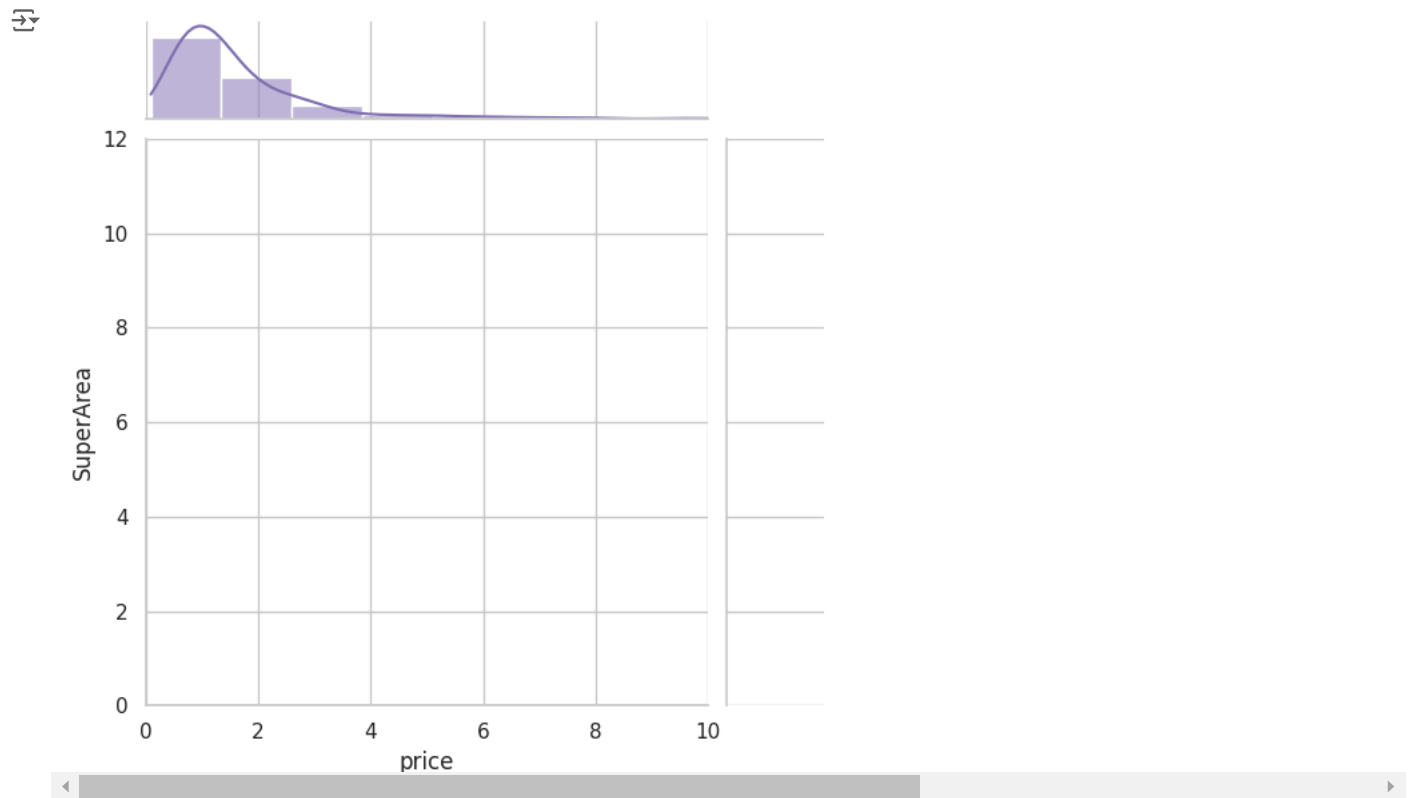
plt.grid(True)
plt.ylabel("BHK")
plt.xlabel("Furnishing")
sns.despine(trim=True, left=True)
plt.show()
```

 <ipython-input-35-691b9010d1b7>:1: FutureWarning:

```
# Linear regression with marginal distributions
```

```
sns.hoxplot(
sns.set_theme(style="darkgrid")

g=sns.jointplot(
    data=df,
    x="price", y="SuperArea",
    kind="reg", truncate=False,xlim=(0,10),ylim=(0,12),color="m",
    marginal_kws=dict(bins=15, fill=True)
)
```



```
# matplotlib
```

```
# pairwise
```

```
import matplotlib.pyplot as plt
import numpy as np
```

```
plt.style.use('_mpl-gallery')
```

```
# make data
```

```
# plot
fig, ax = plt.subplots()
```

```
ax.fill_between(x, y1, y2, alpha=.5, linewidth=0)
```