```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("USA_Housing.csv")
veri = df.copy()
veri.head()
```

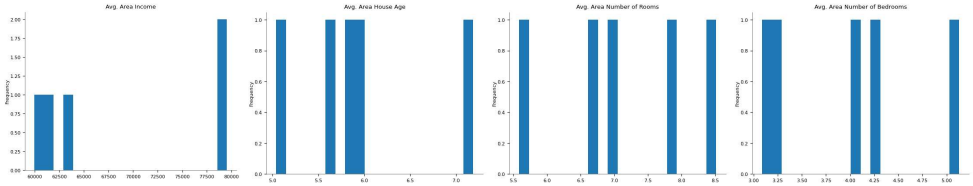
1 to 5 of 5 entries Filter

index	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
0	79545.45857431678	5.682861321615587	7.009188142792237	4.09	23086.800502686456	1059033.5578701235
1	79248.64245482568	6.0028998082752425	6.730821019094919	3.09	40173.07217364482	1505890.91484695
2	61287.067178656784	5.865889840310001	8.512727430375099	5.13	36882.15939970458	1058987.9878760849
3	63345.24004622798	7.1882360945186425	5.586728664827653	3.26	34310.24283090706	1260616.8066294468
4	59982.197225708034	5.040554523106283	7.839387785120487	4.23	26354.109472103148	630943.4893385402

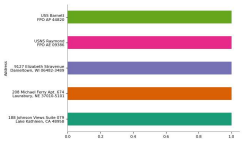
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

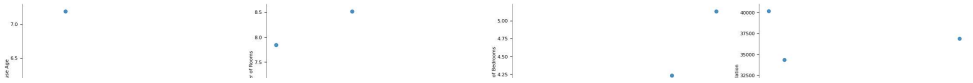
Distributions



Categorical distributions



2-d distributions



```
veri.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             5000 non-null   float64
 1   Avg. Area House Age          5000 non-null   float64
 2   Avg. Area Number of Rooms    5000 non-null   float64
 3   Avg. Area Number of Bedrooms 5000 non-null   float64
 4   Area Population              5000 non-null   float64
 5   Price                        5000 non-null   float64
 6   Address                      5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

```
veri= veri.drop(columns="Address",axis=1)
df.describe().T
```

	count	mean	std	min	25%	5
Avg. Area Income	5000.0	6.858311e+04	10657.991214	17796.631190	61480.562388	6.880429e+
Avg. Area House Age	5000.0	5.977222e+00	0.991456	2.644304	5.322283	5.970429e+
Avg. Area Number of Rooms	5000.0	6.987792e+00	1.005833	3.236194	6.299250	7.002902e+

```
sns.pairplot(veri)
```

```
<seaborn.axisgrid.PairGrid at 0x7964179ae1a0>
```

```
sns.heatmap(veri.corr(),annot=True)
```

```
<Axes: >
```


```
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

y = veri["Price"]
X= veri.drop(columns="Price",axis=1)

cons = sm.add_constant(X)
vif= pd.DataFrame()
vif["variables"]=X.columns
vif["vif"]=[variance_inflation_factor(cons,i+1) for i in range(X.shape[1])]
vif
```

	variables	vif
0	Avg. Area Income	1.001159
1	Avg. Area House Age	1.000577
2	Avg. Area Number of Rooms	1.273535
3	Avg. Area Number of Bedrooms	1.274413
4	Area Population	1.001266

```
from sklearn.model_selection import train_test_split,cross_val_score

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()

X_train = ss.fit_transform(X_train)
X_test = ss.transform(X_test)
import sklearn.metrics as mt

def cross_val(model):
    vali=cross_val_score(model,X,y,cv=10)
    return vali.mean()
def success(true_,pred):
    rmse=mt.mean_absolute_error(true_,pred)
    r2=mt.r2_score(true_,pred)
    return[rmse,r2]
```

```
from sklearn.linear_model import LinearRegression,Ridge,Lasso,ElasticNet
li_model=LinearRegression()
li_model.fit(X_train,y_train)
li_pred = li_model.predict(X_test)

ridge_model=Ridge(alpha=0.1)
ridge_model.fit(X_train,y_train)
ridge_pred = ridge_model.predict(X_test)

lasso_model=Lasso(alpha=0.1)
lasso_model.fit(X_train,y_train)
lasso_pred = lasso_model.predict(X_test)


elas_model=ElasticNet(alpha=0.1)
elas_model.fit(X_train,y_train)
elas_pred = elas_model.predict(X_test)
result=[["Linear model",success(y_test,li_pred)[0],success(y_test,li_pred)[1],cross_val(li_model)],
        ["Ridge model",success(y_test,ridge_pred)[0],success(y_test,ridge_pred)[1],cross_val(ridge_model)],
        ["Lasso model",success(y_test,lasso_pred)[0],success(y_test,lasso_pred)[1],cross_val(lasso_model)],
        ["ElasticNet model",success(y_test,elas_pred)[0],success(y_test,elas_pred)[1],cross_val(elas_model)]
        ]
pd.options.display.float_format="{:.4f}".format
result=pd.DataFrame(result,columns=["Model","RMSE","R2","Verification"])
result
```

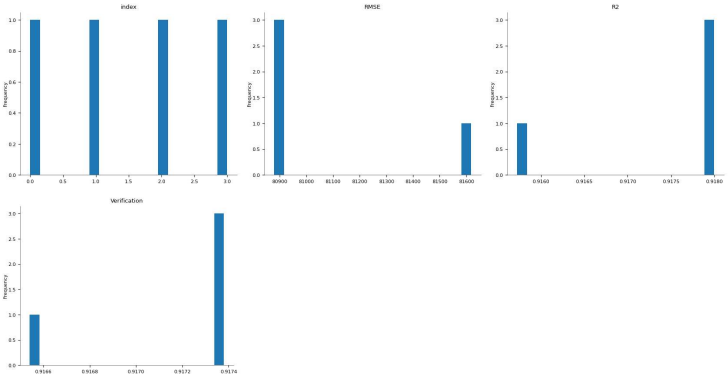
1 to 4 of 4 entries Filter

index	Model	RMSE	R2	Verification
0	Linear model	80879.09723489445	0.9179971706834331	0.9173792621537343
1	Ridge model	80878.96379824344	0.9179971814072102	0.9173792629422894
2	Lasso model	80879.09100489116	0.9179971842462386	0.9173792621933086
3	ElasticNet model	81617.90477553412	0.9157267746303287	0.9165415849526827

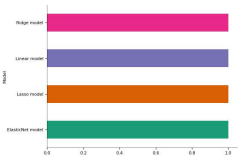
Show 10 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

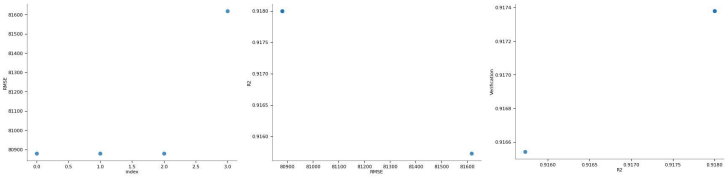
Distributions



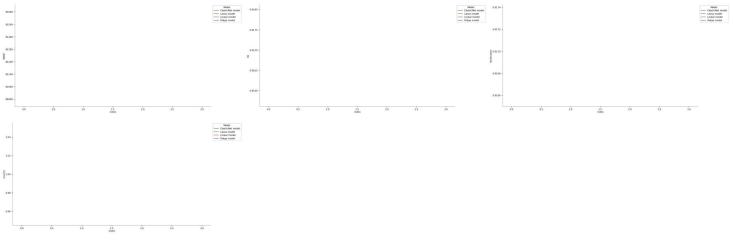
Categorical distributions



2-d distributions



Time series



```
from matplotlib import pyplot as plt
```

```
import seaborn as sns
_df_21.groupby('Model').size().plot(kind='barh', color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```