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

from pandas.plotting import scatter_matrix
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.svm import SVR
%matplotlib inline

# regression dataset INSURANCE
import pandas as pd
df = pd.read_csv('insurance.csv')
df['smoker'] = df['smoker'].replace({'yes':1,'no':0})


df.describe()
```

	age	bmi	children	smoker	charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.663397	1.094918	0.204783	13270.422265
std	14.049960	6.098187	1.205493	0.403694	12110.011237
min	18.000000	15.960000	0.000000	0.000000	1121.873900
25%	27.000000	26.296250	0.000000	0.000000	4740.287150
50%	39.000000	30.400000	1.000000	0.000000	9382.033000
75%	51.000000	34.693750	2.000000	0.000000	16639.912515
max	64.000000	53.130000	5.000000	1.000000	63770.428010

```
df
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	1	southwest	16884.92400
1	18	male	33.770	1	0	southeast	1725.55230
2	28	male	33.000	3	0	southeast	4449.46200
3	33	male	22.705	0	0	northwest	21984.47061
4	32	male	28.880	0	0	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	0	northwest	10600.54830
1334	18	female	31.920	0	0	northeast	2205.98080
1335	18	female	36.850	0	0	southeast	1629.83350
1336	21	female	25.800	0	0	southwest	2007.94500
1337	61	female	29.070	0	1	northwest	29141.36030

1338 rows × 7 columns

```
# Check if any columns have zero values.. EXploratory data analysis
df.isnull().sum()
```

```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

```
# Check if any columns have zero values.. //
df.isnull()
```

	age	sex	bmi	children	smoker	region	charges
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
1333	False	False	False	False	False	False	False
1334	False	False	False	False	False	False	False
1335	False	False	False	False	False	False	False
1336	False	False	False	False	False	False	False
1337	False	False	False	False	False	False	False

1338 rows × 7 columns

```
from sklearn.preprocessing import LabelEncoder
#sex
sex = LabelEncoder()
sex.fit(df.sex.drop_duplicates())
df.sex = sex.transform(df.sex)
df.sex # Outputs 1338 where
```

```
0       0
1       1
2       1
3       1
4       1
       ..
1333    1
1334    0
1335    0
1336    0
1337    0
Name: sex, Length: 1338, dtype: int64
```

```
smoker = LabelEncoder()
smoker.fit(df.smoker.drop_duplicates())
df.smoker = smoker.transform(df.smoker)
df.smoker
```

```
0       1
1       0
2       0
3       0
4       0
       ..
1333    0
1334    0
1335    0
1336    0
1337    1
Name: smoker, Length: 1338, dtype: int64
```

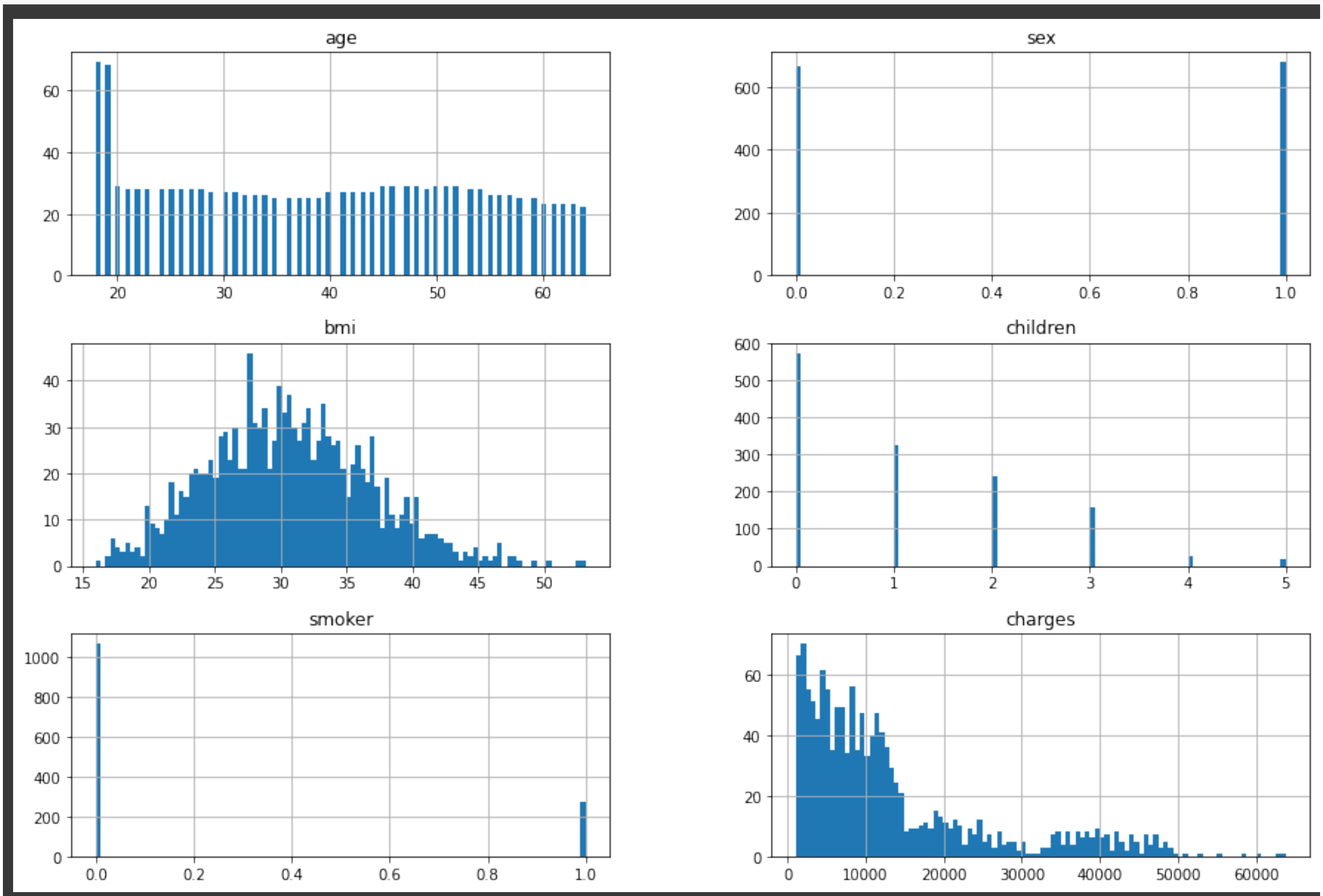
```
# Check how many males and females there are.
print(f"Male: {sum(df.sex == 1)}  +  female: {sum(df.sex == 0)} ")
```

```
Male: 676  +  female: 662
```

```
import matplotlib.pyplot as plt
# Lets get started with regression::
df.describe()
# // Looks like all the colums have the same amount of rows,
```

	age	sex	bmi	children	smoker	charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	0.505232	30.663397	1.094918	0.204783	13270.422265
std	14.049960	0.500160	6.098187	1.205493	0.403694	12110.011237
min	18.000000	0.000000	15.960000	0.000000	0.000000	1121.873900
25%	27.000000	0.000000	26.296250	0.000000	0.000000	4740.287150
50%	39.000000	1.000000	30.400000	1.000000	0.000000	9382.033000
75%	51.000000	1.000000	34.693750	2.000000	0.000000	16639.912515
max	64.000000	1.000000	53.130000	5.000000	1.000000	63770.428010

```
# Creating distributional histogram is important to see how the data distributes
df.hist(bins = 100, figsize =(15,10))
plt.show()
```



```
df
```

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	1	southwest	16884.92400
1	18	1	33.770	1	0	southeast	1725.55230
2	28	1	33.000	3	0	southeast	4449.46200
3	33	1	22.705	0	0	northwest	21984.47061
4	32	1	28.880	0	0	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	1	30.970	3	0	northwest	10600.54830
1334	18	0	31.920	0	0	northeast	2205.98080
1335	18	0	36.850	0	0	southeast	1629.83350
1336	21	0	25.800	0	0	southwest	2007.94500
1337	61	0	29.070	0	1	northwest	29141.36030

1338 rows × 7 columns

```
from sklearn.model_selection import train_test_split
# train test
train, test = train_test_split(df, test_size = 0.4) # 60% train, 40% test
val, test = train_test_split(test, test_size = 0.2) # 20% validation set and 20%

charges = train.copy()

corr_charges = charges.corr()
corr_charges["age"].sort_values(ascending = False)
```

```
age         1.000000
charges     0.319711
bmi         0.076376
children    0.044988
smoker     -0.009986
sex        -0.059103
Name: age, dtype: float64
```

```
corr_charges = charges.corr()
corr_charges["children"].sort_values(ascending = False)
```

```
children    1.000000
age         0.044988
charges     0.040443
bmi         0.018450
sex         0.016983
smoker     -0.010935
Name: children, dtype: float64
```

```
corr_charges = charges.corr()
corr_charges["bmi"].sort_values(ascending = False)
```

```
bmi         1.000000
charges     0.144544
age         0.076376
sex         0.022883
children    0.018450
smoker     -0.021673
Name: bmi, dtype: float64
```

```
corr_charges = charges.corr()
corr_charges["sex"].sort_values(ascending = False)
```

```
sex         1.000000
smoker      0.050688
bmi         0.022883
charges     0.022187
children    0.016983
age        -0.059103
Name: sex, dtype: float64
```

```
corr_charges = charges.corr()
corr_charges["charges"].sort_values(ascending = False)
```

```
charges     1.000000
smoker      0.783890
age         0.319711
bmi         0.144544
children    0.040443
sex         0.022187
Name: charges, dtype: float64
```

```
df['smoker'].head(30)
```

```
0     1
1     0
2     0
3     0
4     0
5     0
6     0
7     0
8     0
9     0
10    0
11    1
12    0
13    0
14    1
15    0
16    0
17    0
18    0
19    1
20    0
21    0
22    0
23    1
24    0
25    0
26    0
27    0
28    0
29    1
Name: smoker, dtype: int64
```

```
corr_charges = charges.corr()
corr_charges["smoker"].sort_values(ascending = False)
```

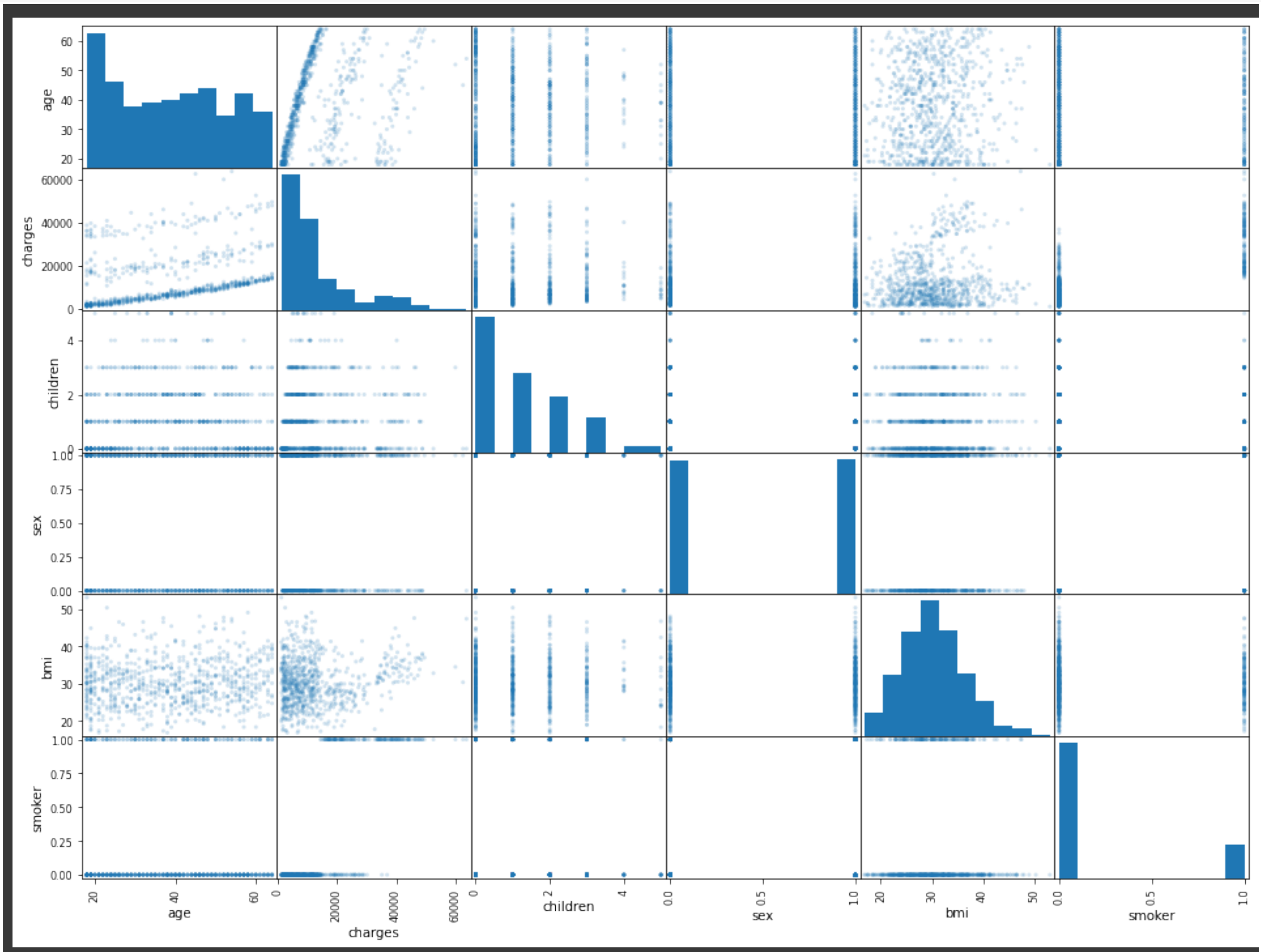
```
smoker      1.000000
charges     0.783890
sex         0.050688
age        -0.009986
children   -0.010935
bmi        -0.021673
Name: smoker, dtype: float64
```

```
## From above examples can we conduct there facts: there is a slight correlation

# min max normalization
y = df['charges']
x = df.drop(['charges', 'sex', 'smoker', 'region'], axis = 1)

df_min_maxed_norm = ((x - x.min() ) / (x.max() - x.min()))
df_min_maxed_norm
```

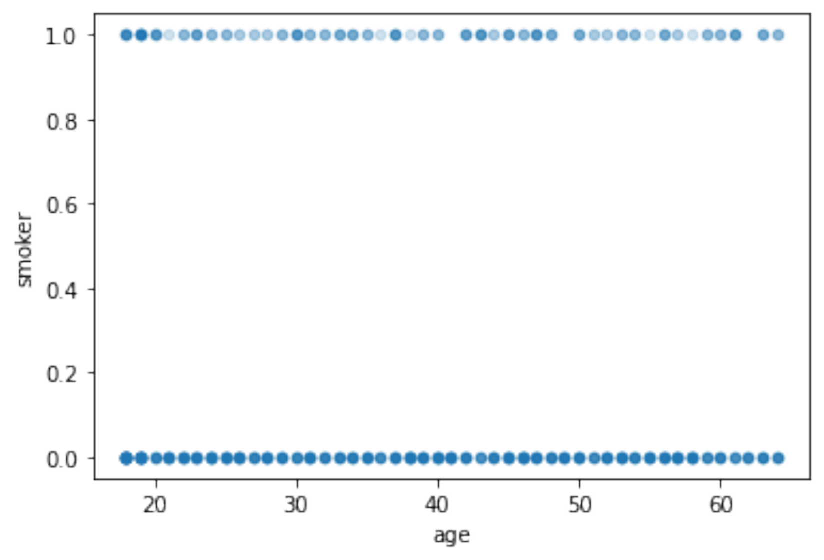
	age	bmi	children
0	0.021739	0.321227	0.0
1	0.000000	0.479150	0.2
2	0.217391	0.458434	0.6
3	0.326087	0.181464	0.0
4	0.304348	0.347592	0.0
...	...	...	...
1333	0.695652	0.403820	0.6
1334	0.000000	0.429379	0.0
1335	0.000000	0.562012	0.0
1336	0.065217	0.264730	0.0
1337	0.934783	0.352704	0.0

1338 rows × 3 columns

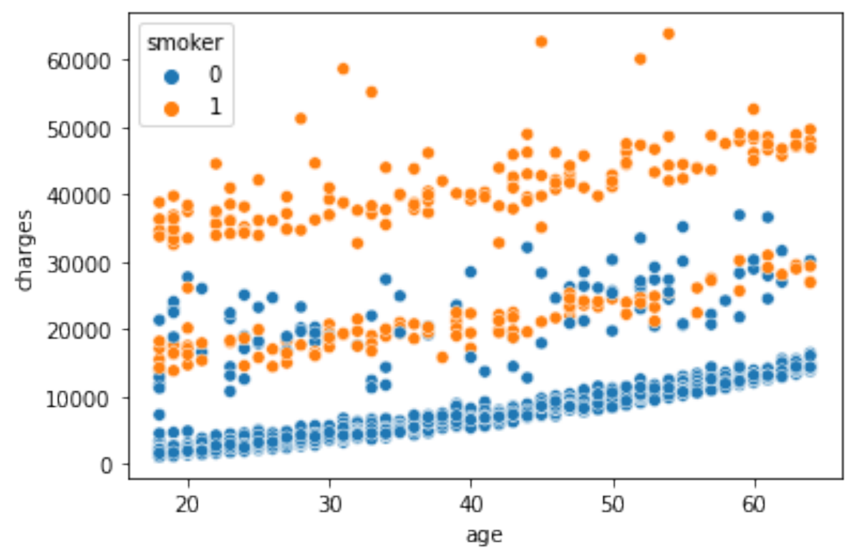
```
from pandas.plotting import scatter_matrix
df_names = ["age","charges", "children", "sex", "bmi", "smoker"]
scatter_matrix(charges[df_names], figsize = (16,12), alpha = 0.2);
```



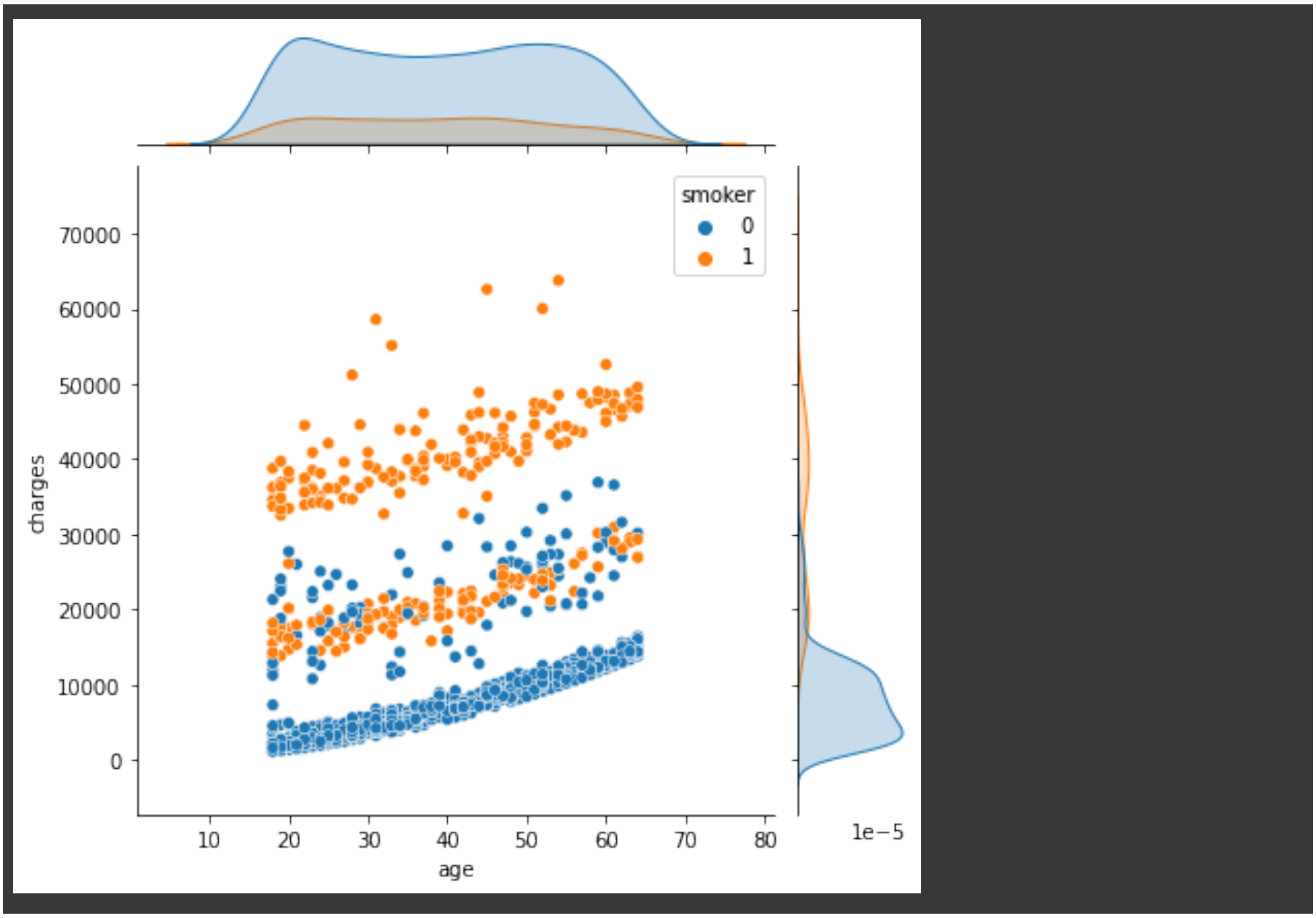
```
charges.plot(kind="scatter", x="age", y="smoker",alpha=0.2);
```



```
from matplotlib.transforms import BboxTransform
# Scatterplot the length and with of sepal.
import seaborn as sns
import matplotlib.pyplot as plt


sns.scatterplot(x='age', y='charges', hue='smoker', data=df, )

plt.show()
```



```
sns.jointplot(x='age',y='charges', hue = 'smoker', data=df)
plt.show()
```


smoker
charges
age
1e−5