rk1

April 2, 2023

1 1

1.1

[ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_wine
```

1.2 №13

( ) “ - 1 / X”.

1.2.1

[ ]:

```
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    #
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

[ ]:

```
#
data = pd.read_csv('cars.csv', sep=",")

data.head()
```

[ ]:

```
             car_name  reviews_count fuel_type  engine_displacement  \
0     Maruti Alto K10             51    Petrol                  998
1       Maruti Brezza             86    Petrol                 1462
2       Mahindra Thar            242    Diesel                 2184
3     Mahindra XUV700            313    Diesel                 2198
4  Mahindra Scorpio-N            107    Diesel                 2198
```

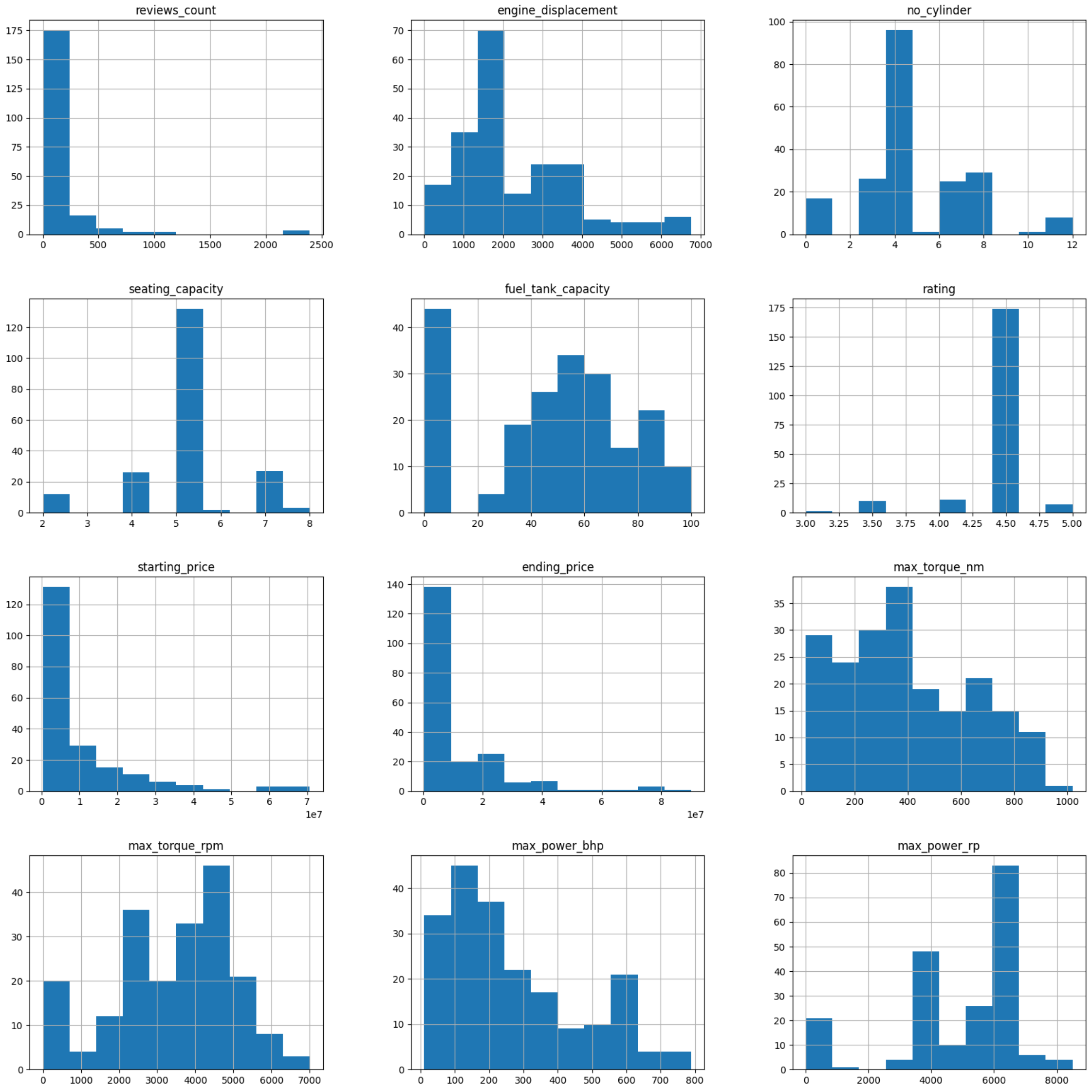
```
   no_cylinder  seating_capacity transmission_type  fuel_tank_capacity  \
0            3               5.0         Automatic                27.0
1            4               5.0         Automatic                48.0
2            4               4.0         Automatic                57.0
3            4               7.0         Automatic                60.0
4            4               7.0         Automatic                57.0

   body_type  rating  starting_price  ending_price  max_torque_nm  \
0  Hatchback     4.5          399000        583000           89.0
1        SUV     4.5          799000       1396000          136.8
2        SUV     4.5         1353000       1603000          300.0
3        SUV     4.5         1318000       2458000          450.0
4        SUV     4.5         1199000       2390000          400.0

   max_torque_rpm  max_power_bhp  max_power_rp
0            3500          65.71          5500
1            4400         101.65          6000
2            2800         130.00          3750
3            2800         182.38          3500
4            2750         172.45          3500
```

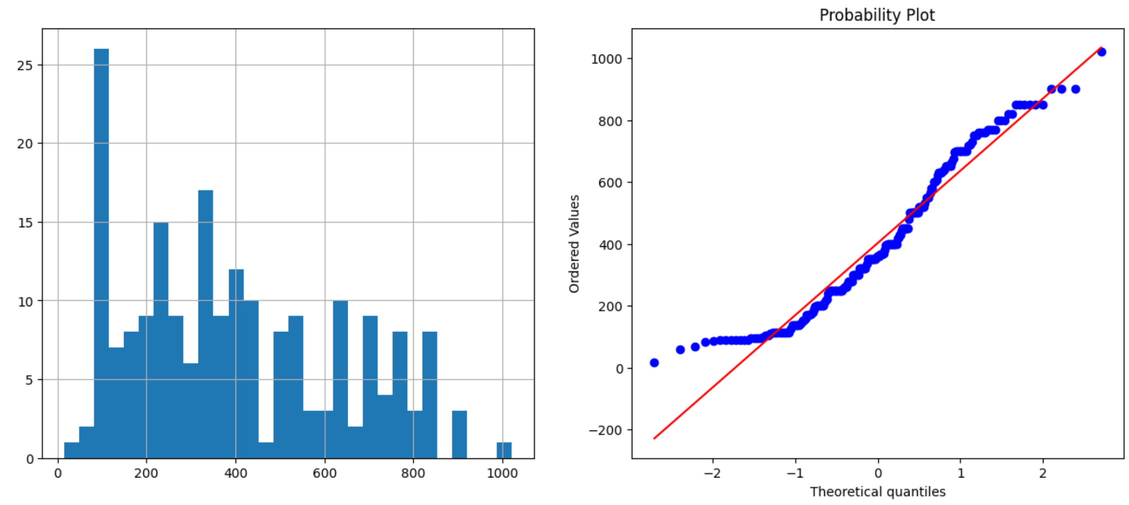
[ ]:

```
data.hist(figsize=(20,20))
plt.show()
```



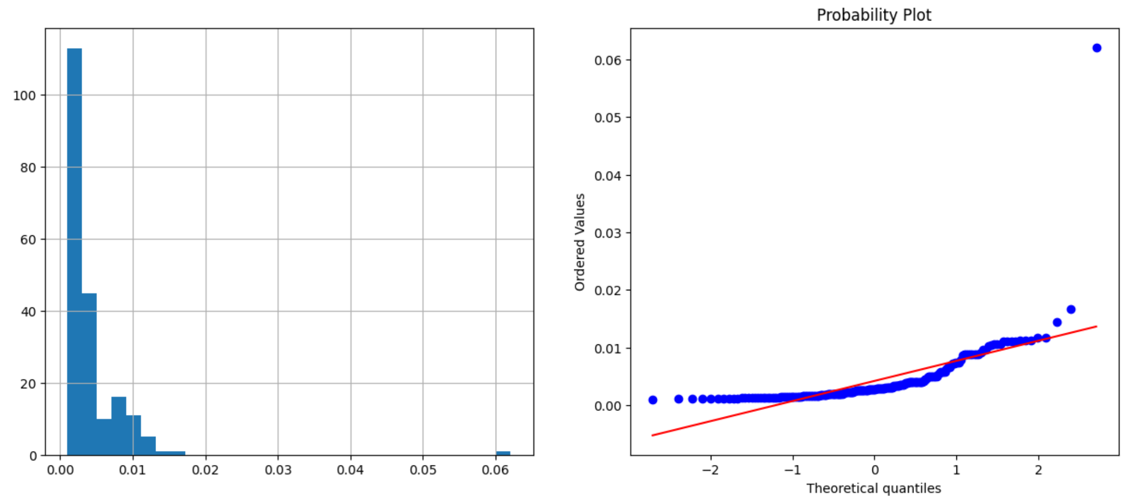
1.2.2

```
[ ]: diagnostic_plots(data, 'max_torque_nm')
```



1.2.3

```
[ ]: data['max_torque_nm_reciprocal'] = 1 / (data['max_torque_nm'])
     diagnostic_plots(data, 'max_torque_nm_reciprocal')
```



1.3 №33

(feature selection). (wrapper method),
(exhaustive feature selection).

1.3.1

[ ]:

```
wine = load_wine()
wine_X = wine.data
wine_y = wine.target
wine_feature_names = wine['feature_names']
wine_x_df = pd.DataFrame(data=wine['data'], columns=wine['feature_names'])
```

1.3.2 (wrapper methods)

[ ]:

```
knn = KNeighborsClassifier(n_neighbors=3)
```

[ ]:

```
efs1 = EFS(knn,
           min_features=2,
           max_features=4,
           scoring='accuracy',
           print_progress=True,
           cv=5)

efs1 = efs1.fit(wine_X, wine_y)
# efs1 = efs1.fit(iris_X, iris_y, custom_feature_names=iris_feature_names)

print('Best accuracy score: %.2f' % efs1.best_score_)
print('Best subset (indices):', efs1.best_idx_)
print('Best subset (corresponding names):', efs1.best_feature_names_)
```

```
Features: 1079/1079

Best accuracy score: 0.94
Best subset (indices): (0, 5, 6, 9)
Best subset (corresponding names): ('0', '5', '6', '9')
```

[ ]:

```
efs2 = EFS(knn,
           min_features=1,
           max_features=2,
           scoring='accuracy',
           print_progress=True,
           cv=5)

efs2 = efs2.fit(wine_X, wine_y)
# efs2 = efs2.fit(iris_X, iris_y, custom_feature_names=iris_feature_names)

print('Best accuracy score: %.2f' % efs2.best_score_)
print('Best subset (indices):', efs2.best_idx_)
print('Best subset (corresponding names):', efs2.best_feature_names_)
```

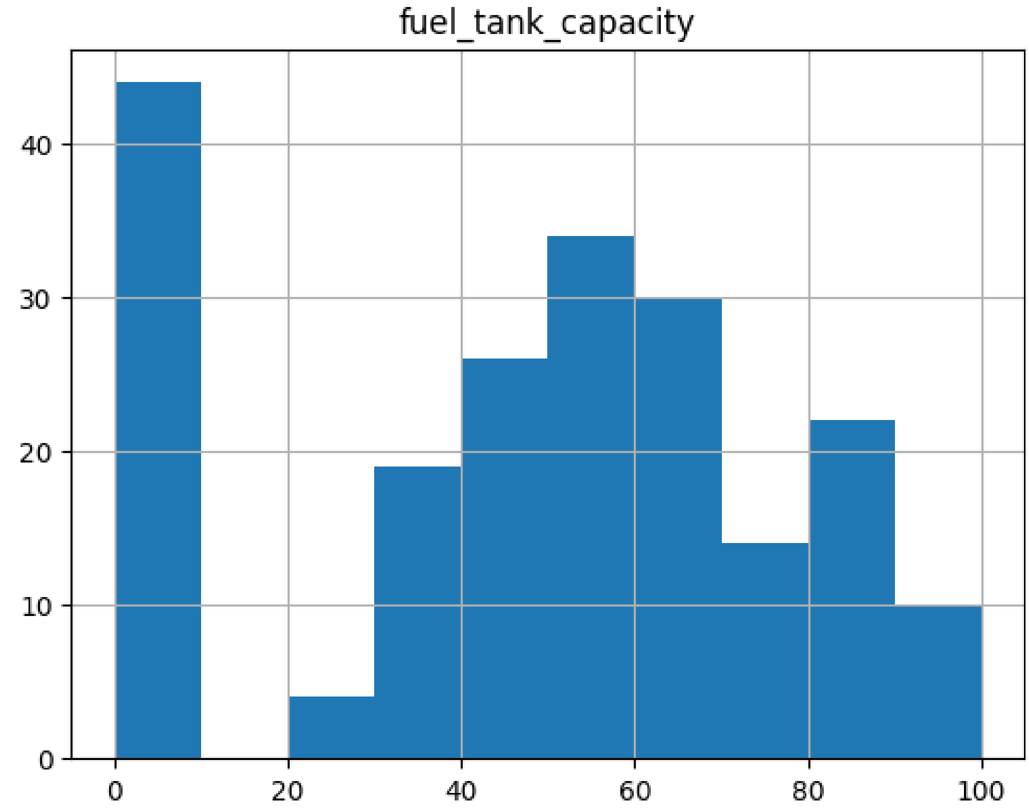
```
Features: 91/91

Best accuracy score: 0.93
Best subset (indices): (6, 9)
Best subset (corresponding names): ('6', '9')
```

1.3.3

5-22 , 5 -22 - .

```
[ ]: data.hist('fuel_tank_capacity')
     plt.show()
```


fuel_tank_capacity