

Dataset from Kaggle: <https://www.kaggle.com/datasets/syednwarafri/vehicle-sales-data>

1. year int64
2. make object
3. model object
4. trim object
5. body object
6. transmission object
7. vin object
8. state object
9. condition float64
10. odometer float64
11. color object
12. interior object
13. seller object
14. mmr int64
15. sellingprice int64
16. saledate object

```
Ввод [49]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
%matplotlib inline
sns.set(style="ticks")
```

```
Ввод [50]: data = pd.read_csv('car_prices.csv', sep=",")
```

Ввод [51]: `data.isna().sum()`

```
Out[51]: year          0
         make         10301
         model        10399
         trim         10651
         body         13195
         transmission  65352
         vin           4
         state         0
         condition    11820
         odometer      94
         color         749
         interior      749
         seller         0
         mmr           38
         sellingprice  12
         saledate      12
         dtype: int64
```

Ввод [52]: `data.dtypes`

```
Out[52]: year          int64
         make          object
         model          object
         trim           object
         body           object
         transmission   object
         vin            object
         state          object
         condition      float64
         odometer       float64
         color          object
         interior       object
         seller         object
         mmr            float64
         sellingprice   float64
         saledate       object
         dtype: object
```

Ввод [53]: `data.shape`

```
Out[53]: (558837, 16)
```

```

Ввод [54]: def diagnostic_plots(df, variable, title):
fig, ax = plt.subplots(figsize=(10,7))
# гистограмма
plt.subplot(2, 2, 1)
df[variable].hist(bins=30)
## Q-Q plot
plt.subplot(2, 2, 2)
stats.probplot(df[variable], dist="norm", plot=plt)
# ящик с усами
plt.subplot(2, 2, 3)
sns.violinplot(x=df[variable])
# ящик с усами
plt.subplot(2, 2, 4)
sns.boxplot(x=df[variable])
fig.suptitle(title)
plt.show()

```

```

Ввод [55]: diagnostic_plots(data, 'condition', 'Condition of the vehicle.')

```

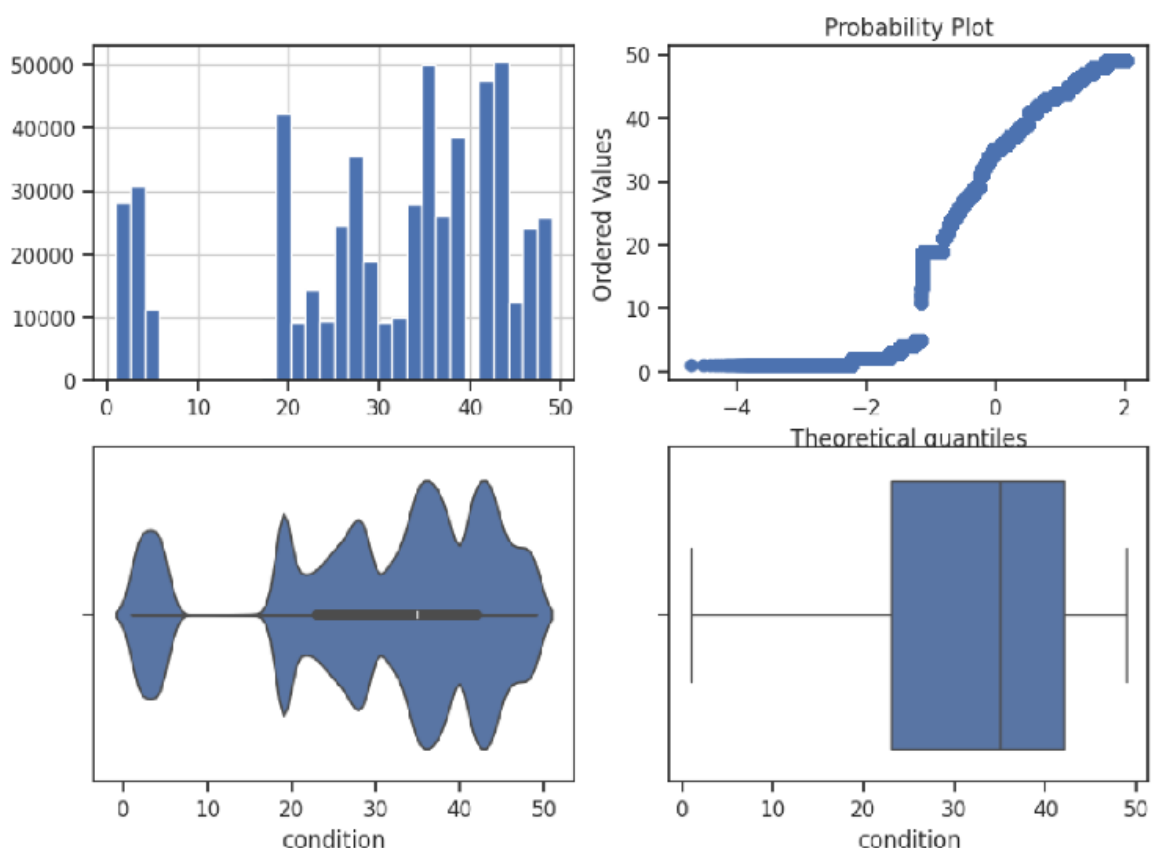
<ipython-input-54-766c933c159f>:4: MatplotlibDeprecationWarning: Auto-removal of overlapping axes is deprecated since 3.6 and will be removed two minor releases later; explicitly call ax.remove() as needed.

```

plt.subplot(2, 2, 1)

```

Condition of the vehicle.



Значение condition заполняется модой.

```

Ввод [56]: temp_data = data[['condition']].values
           size = temp_data.shape[0]

           IQR = data['condition'].quantile(0.75) - data['condition'].quantile(0.25)

           from sklearn.impute import SimpleImputer

           imputer = SimpleImputer(strategy='constant', fill_value=data['condition'].qu
           all_data = imputer.fit_transform(temp_data)

           median_df = data.copy()
           median_df['condition'] = all_data

```

```

Ввод [57]: median_df.isna().sum()

```

```

Out[57]: year          0
         make         10301
         model        10399
         trim         10651
         body         13195
         transmission  65352
         vin           4
         state         0
         condition     0
         odometer      94
         color         749
         interior      749
         seller         0
         mmr           38
         sellingprice  12
         saledate      12
         dtype: int64

```

```

Ввод [58]: from sklearn.feature_selection import VarianceThreshold
           selector_1211 = VarianceThreshold(threshold=0.15)
           selector_1211.fit(median_df[['year', 'condition', 'odometer']])
           # Значения дисперсий для каждого признака
           selector_1211.variances_

```

```

Out[58]: array([1.57359788e+01, 2.08677136e+02, 2.85139927e+09])

```

Добавляем колонку в силу отсутствия достаточно малых дисперсий:

```

Ввод [59]: pseudovar = np.random.normal(loc=10, scale = 0.1, size = (median_df.shape[0]
median_df.insert(16, "PSV", pseudovar, True)

           selector_1211 = VarianceThreshold(threshold=0.15)
           selector_1211.fit(median_df[['year', 'condition', 'odometer', 'PSV']])
           # Значения дисперсий для каждого признака
           selector_1211.variances_

```

```

Out[59]: array([1.57359788e+01, 2.08677136e+02, 2.85139927e+09, 9.95924918e-03])

```

```
Ввод [60]: # Константный и псевдоконстантный признаки удалены
selector_1211.transform(median_df[['year', 'condition', 'odometer', 'PSV']])
```

```
Out[60]: array([[2.0150e+03, 5.0000e+00, 1.6639e+04],
                [2.0150e+03, 5.0000e+00, 9.3930e+03],
                [2.0140e+03, 4.5000e+01, 1.3310e+03],
                ...,
                [2.0120e+03, 4.8000e+01, 5.0561e+04],
                [2.0150e+03, 3.8000e+01, 1.6658e+04],
                [2.0140e+03, 3.4000e+01, 1.5008e+04]])
```

```
Ввод [61]: #Удалим колонку
median_df=median_df.drop(columns = ['PSV'])
```

```
Ввод [62]: median_df.shape
```

```
Out[62]: (558837, 16)
```

```
Ввод [63]: data.dtypes
```

```
Out[63]: year          int64
make          object
model         object
trim          object
body          object
transmission   object
vin           object
state         object
condition      float64
odometer       float64
color          object
interior       object
seller         object
mmr           float64
sellingprice   float64
saledate       object
dtype: object
```

```
Ввод [64]: # Диаграмма рассеяния
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='condition', y='sellingprice', data = median_df)
plt.xlabel('condition')
plt.ylabel('sellingprice')
```

```
Out[64]: Text(0, 0.5, 'sellingprice')
```

