```python
# импортируем основные библиотеки
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import sklearn
from sklearn.preprocessing import

import warnings
warnings.filterwarnings("ignore")
```

##*Выгрузка исходных датасетов*

```python
# датасеты загружены через облако google drive
X_bp = pd.read_excel('/content/drive/MyDrive/DatasetsVKR/X_bp.xlsx') # датасет с характеристиками базальтопластика
X_bp.drop('Unnamed: 0', axis = 1, inplace=True) # избавляемся от лишнего столбца индексов
print(X_bp.shape) # выводим размерность 1го датасета
X_bp.head()
```

```
(1023, 10)
```

{"summary":"{\n \"name\": \"X_bp\",\n \"rows\": 1023,\n \"fields\": [\n   {\n   \"column\": \"\\u0421\\u043e\\u043e\\u0442\\u043d\\u043e\\u0448\\u0435\\u043d\\u0438\\u0435 \\u043c\\u0430\\u0442\\u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\u043f\\u043e\\u043b\\u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.9132222362148383,\n   \"min\": 0.389402605178414,\n   \"max\": 5.59174159869754,\n   \"num_unique_values\": 1014,\n   \"samples\": [\n   3.62568575586845,\n   3.48220554972608,\n   2.36749513759979\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\u043a\\u0433/\\u043c3\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 73.7292305506539,\n   \"min\": 1731.764635096,\n   \"max\": 2207.77348061119,\n   \"num_unique_values\": 1013,\n   \"samples\": [\n   1965.75243927466,\n   1972.32028526551,\n   1964.8014374637\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u043c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\u0442\\u0438, \\u0413\\u041f\\u0430\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 330.2315805610269,\n   \"min\": 2.4369087535075,\n   \"max\": 1911.53647700054,\n   \"num_unique_values\": 1020,\n   \"samples\": [\n   132.549747560606,\n

```
569.930605163672,\n          471.506958157975\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041a\\u043e\\u043b\\u0438\\
u0447\\u0435\\u0441\\u0442\\u0432\\u043e \\u043e\\u0442\\u0432\\
u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\u043b\\u044f, \\u043c.%\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
28.295911288788833,\n        \"min\": 17.7402745562519,\n
\"max\": 198.953207190451,\n        \"num_unique_values\": 1005,\n
\"samples\": [\n          112.520527514754,\n
72.1819854684613,\n          106.988610225557\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u0421\\u043e\\u0434\\u0435\\
u0440\\u0436\\u0430\\u043d\\u0438\\u0435 \\u044d\\u043f\\u043e\\
u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445 \\u0433\\u0440\\
u0443\\u043f\\u043f,%_2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.4063012915294295,\n
\"min\": 14.2549854977161,\n        \"max\": 33.0,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
24.9011113658796,\n          23.5532191008226,\n
23.452082763333\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0422\\u0435\\u043c\\u043f\\u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\
u0430 \\u0432\\u0441\\u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 40.9432599529233,\n        \"min\": 100.0,\n        \"max\":
413.273418243566,\n        \"num_unique_values\": 1003,\n
\"samples\": [\n          286.701064731226,\n
397.151290856661,\n          269.840299552991\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u0430\\u044f \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u0433/\\u043c2\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 281.31469023666097,\n        \"min\":
0.603739925153945,\n        \"max\": 1399.54236233989,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
443.164680639388,\n          438.290234061128,\n
268.217818421553\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438 \\u043f\\u0440\\u0438 \\
u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\u0435\\u043d\\u0438\\u0438,
\\u0413\\u041f\\u0430\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.118982889469302,\n        \"min\":
64.0540605597917,\n        \"max\": 82.682051035271,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
75.1622076006716,\n          75.3033941703679,\n
72.1972680303294\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
```

```
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
485.62800627853596,\n        \"min\": 1036.85660535,\n        \"max\":
3848.43673187618,\n        \"num_unique_values\": 1004,\n
\"samples\": [\n          2948.96846718064,\n
2751.99820688163,\n          1904.38711713426\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0442\\u0440\\
u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\u0435 \\u0441\\u043c\\
u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 59.73593087332353,\n
\"min\": 33.8030255329625,\n        \"max\": 414.590628361534,\n
\"num_unique_values\": 1003,\n        \"samples\": [\n
180.068188823332,\n          196.82222775264,\n
190.875279016224\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"X_bp"}
```


```python
X_nup = pd.read_excel('/content/drive/MyDrive/DatasetsVKR/X_nup.xlsx')
# датасет с характеристиками углепластика
X_nup.drop('Unnamed: 0', axis = 1, inplace=True) # избавляемся от
лишнего столбца индексов
print(X_nup.shape) # выводим размерность 2го датасета
X_nup.head()
```

```
(1040, 3)
```


```
{"summary":"{\n  \"name\": \"X_nup\",\n  \"rows\": 1040,\n
\"fields\": [\n    {\n      \"column\": \"\\u0423\\u0433\\u043e\\u043b
\\u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\u0438, \\u0433\\u0440\\
u0430\\u0434\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 45,\n        \"min\": 0,\n
\"max\": 90,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          90,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"\\u0428\\u0430\\u0433 \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.5551805320335697,\n        \"min\":
0.0,\n        \"max\": 14.4405218753969,\n
\"num_unique_values\": 1006,\n        \"samples\": [\n
9.90150443857602,\n          7.19065011209914\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\
u043d\\u043e\\u0441\\u0442\\u044c \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12.332437966648541,\n        \"min\":
0.0,\n        \"max\": 103.988901301494,\n
\"num_unique_values\": 1005,\n        \"samples\": [\n
```

```
67.7188963415148,\n          50.8236748784086\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"X_nup"}
```


Согласно условию задания, необходимо замерджить датасеты по типу объединения INNER

```
df_X = X_bp.merge(X_nup, left_index = True, right_index = True, how =
'inner')
print(f'Итоговый датасет имеет размерность {df_X.shape},на
первоначальном этапе обработки данных было отброшено {X_nup.shape[0]-
df_X.shape[0]} строк из 2го датасета')
df_X.head()
```

```
Итоговый датасет имеет размерность (1023, 13),на первоначальном этапе
обработки данных было отброшено 17 строк из 2го датасета
```


```
{"summary":"{\n  \"name\": \"df_X\",\n  \"rows\": 1023,\n  \"fields\":
[\n    {\n      \"column\": \"\\u0421\\u043e\\u043e\\u0442\\u043d\\
u043e\\u0448\\u0435\\u043d\\u0438\\u0435 \\u043c\\u0430\\u0442\\
u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\u043f\\u043e\\u043b\\
u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.9132222362148383,\n
\"min\": 0.389402605178414,\n        \"max\": 5.59174159869754,\n
\"num_unique_values\": 1014,\n        \"samples\": [\n
3.62568575586845,\n          3.48220554972608,\n
2.36749513759979\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u043a\\u0433/\\u043c3\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 73.7292305506539,\n        \"min\":
1731.764635096,\n        \"max\": 2207.77348061119,\n
\"num_unique_values\": 1013,\n        \"samples\": [\n
1965.75243927466,\n          1972.32028526551,\n
1964.8014374637\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u043c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438, \\u0413\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
330.2315805610269,\n        \"min\": 2.4369087535075,\n
\"max\": 1911.53647700054,\n        \"num_unique_values\": 1020,\n
\"samples\": [\n          132.549747560606,\n
569.930605163672,\n          471.506958157975\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041a\\u043e\\u043b\\u0438\\
u0447\\u0435\\u0441\\u0442\\u0432\\u043e \\u043e\\u0442\\u0432\\
u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\u043b\\u044f, \\u043c.%\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
28.295911288788833,\n        \"min\": 17.7402745562519,\n
\"max\": 198.953207190451,\n        \"num_unique_values\": 1005,\n
```

\"samples\": [\n   112.520527514754,\n
72.1819854684613,\n   106.988610225557\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u0421\\u043e\\u0434\\u0435\\
u0440\\u0436\\u0430\\u043d\\u0438\\u0435 \\u044d\\u043f\\u043e\\
u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445 \\u0433\\u0440\\
u0443\\u043f\\u043f,%_2\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 2.4063012915294295,\n
\"min\": 14.2549854977161,\n   \"max\": 33.0,\n
\"num_unique_values\": 1004,\n   \"samples\": [\n
24.9011113658796,\n   23.5532191008226,\n
23.452082763333\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u0422\\u0435\\u043c\\u043f\\u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\
u0430 \\u0432\\u0441\\u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\
n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 40.9432599529233,\n   \"min\": 100.0,\n   \"max\":
413.273418243566,\n   \"num_unique_values\": 1003,\n
\"samples\": [\n   286.701064731226,\n
397.151290856661,\n   269.840299552991\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u041f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u0430\\u044f \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u0433/\\u043c2\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 281.31469023666097,\n   \"min\":
0.603739925153945,\n   \"max\": 1399.54236233989,\n
\"num_unique_values\": 1004,\n   \"samples\": [\n
443.164680639388,\n   438.290234061128,\n
268.217818421553\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438 \\u043f\\u0440\\u0438 \\
u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\u0435\\u043d\\u0438\\u0438,
\\u0413\\u041f\\u0430\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 3.118982889469302,\n   \"min\":
64.0540605597917,\n   \"max\": 82.682051035271,\n
\"num_unique_values\": 1004,\n   \"samples\": [\n
75.1622076006716,\n   75.3033941703679,\n
72.1972680303294\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
485.62800627853596,\n   \"min\": 1036.85660535,\n   \"max\":
3848.43673187618,\n   \"num_unique_values\": 1004,\n
\"samples\": [\n   2948.96846718064,\n
2751.99820688163,\n   1904.38711713426\n   ],\n

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0442\\u0440\\
u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\u0435 \\u0441\\u043c\\
u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 59.73593087332353,\n
\"min\": 33.8030255329625,\n        \"max\": 414.590628361534,\n
\"num_unique_values\": 1003,\n        \"samples\": [\n
180.068188823332,\n          196.82222775264,\n
190.875279016224\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0423\\u0433\\u043e\\u043b \\u043d\\u0430\\u0448\\u0438\\u0432\\
u043a\\u0438, \\u0433\\u0440\\u0430\\u0434\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 45,\n
\"min\": 0,\n        \"max\": 90,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          90,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u0428\\u0430\\u0433 \\u043d\\
u0430\\u0448\\u0438\\u0432\\u043a\\u0438\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.5634670728338826,\n
\"min\": 0.0,\n        \"max\": 14.4405218753969,\n
\"num_unique_values\": 989,\n        \"samples\": [\n
8.56581217318028,\n          2.2156519282891\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\
u043d\\u043e\\u0441\\u0442\\u044c \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12.350968798651312,\n        \"min\":
0.0,\n        \"max\": 103.988901301494,\n
\"num_unique_values\": 988,\n        \"samples\": [\n
43.9076879570936,\n          41.2464520255753\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df_X"}
```


##*Проведение разведочного анализа данных (EDA)*

```
df_X.info() # выведем основую информацию о сформированном датасете
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   Соотношение матрица-наполнитель        1023 non-null   float64
 1   Плотность, кг/м3                       1023 non-null   float64
 2   модуль упругости, ГПа                  1023 non-null   float64
 3   Количество отвердителя, м.%            1023 non-null   float64
 4   Содержание эпоксидных групп,%_2        1023 non-null   float64
 5   Температура вспышки, С_2               1023 non-null   float64
 6   Поверхностная плотность, г/м2          1023 non-null   float64
```

```
 7   Модуль упругости при растяжении, ГПа  1023 non-null   float64
 8   Прочность при растяжении, МПа         1023 non-null   float64
 9   Потребление смолы, г/м2               1023 non-null   float64
 10  Угол нашивки, град                    1023 non-null   int64
 11  Шаг нашивки                           1023 non-null   float64
 12  Плотность нашивки                     1023 non-null   float64
dtypes: float64(12), int64(1)
memory usage: 111.9 KB
```

```
df_X.isnull().sum()
```

```
Соотношение матрица-наполнитель         0
Плотность, кг/м3                        0
модуль упругости, ГПа                   0
Количество отвердителя, м.%             0
Содержание эпоксидных групп,%_2         0
Температура вспышки, С_2                0
Поверхностная плотность, г/м2           0
Модуль упругости при растяжении, ГПа    0
Прочность при растяжении, МПа           0
Потребление смолы, г/м2                 0
Угол нашивки, град                      0
Шаг нашивки                             0
Плотность нашивки                       0
dtype: int64
```

```
# все переменные имеют тип float64/int64, качественных характеристик и
пробелов в датасете нет
df_X.nunique() # количество уникальных значений по каждому из
параметров
```

```
Соотношение матрица-наполнитель         1014
Плотность, кг/м3                        1013
модуль упругости, ГПа                   1020
Количество отвердителя, м.%             1005
Содержание эпоксидных групп,%_2         1004
Температура вспышки, С_2                1003
Поверхностная плотность, г/м2           1004
Модуль упругости при растяжении, ГПа    1004
Прочность при растяжении, МПа           1004
Потребление смолы, г/м2                 1003
Угол нашивки, град                         2
Шаг нашивки                              989
Плотность нашивки                        988
dtype: int64
```

```
# из всех параметров примечателен угол нашивки - у него всего 2
уникальных значения
df_X['Угол нашивки, град'].unique()
```

```
array([ 0, 90])
```

```python
df_X['Угол нашивки, град'] = df_X['Угол нашивки, град'].replace({0.0: 0, 90.0: 1}).astype(int) # заменяем значения градусов на целые числа 0 и 1
df_X = df_X.rename(columns={'Угол нашивки, град' : 'Угол нашивки', 'модуль упругости, ГПа' : 'Модуль упругости, ГПа'}) # так как мы теперь имеем дело не с градусами в качестве единицы измерения, изменяем название соответствующего столбца
df_X
```


```
{"summary":"{\n  \"name\": \"df_X\",\n  \"rows\": 1023,\n  \"fields\": [\n    {\n      \"column\": \"\\u0421\\u043e\\u043e\\u0442\\u043d\\u043e\\u0448\\u0435\\u043d\\u0438\\u0435 \\u043c\\u0430\\u0442\\u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\u043f\\u043e\\u043b\\u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.9132222362148383,\n        \"min\": 0.389402605178414,\n        \"max\": 5.59174159869754,\n        \"num_unique_values\": 1014,\n        \"samples\": [\n          3.62568575586845,\n          3.48220554972608,\n          2.36749513759979\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\u043a\\u0433/\\u043c3\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 73.7292305506539,\n        \"min\": 1731.764635096,\n        \"max\": 2207.77348061119,\n        \"num_unique_values\": 1013,\n        \"samples\": [\n          1965.75243927466,\n          1972.32028526551,\n          1964.8014374637\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\u0442\\u0438, \\u0413\\u041f\\u0430\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 330.2315805610269,\n        \"min\": 2.4369087535075,\n        \"max\": 1911.53647700054,\n        \"num_unique_values\": 1020,\n        \"samples\": [\n          132.549747560606,\n          569.930605163672,\n          471.506958157975\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u041a\\u043e\\u043b\\u0438\\u0447\\u0435\\u0441\\u0442\\u0432\\u043e \\u043e\\u0442\\u0432\\u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\u043b\\u044f, \\u043c.%\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 28.295911288788833,\n        \"min\": 17.7402745562519,\n        \"max\": 198.953207190451,\n        \"num_unique_values\": 1005,\n        \"samples\": [\n          112.520527514754,\n          72.1819854684613,\n          106.988610225557\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u0421\\u043e\\u0434\\u0435\\u0440\\u0436\\u0430\\u043d\\u0438\\u0435 \\u044d\\u043f\\u043e\\u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445 \\u0433\\u0440\\u0443\\u043f,%_2\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 2.4063012915294295,\n
```

```
\"min\": 14.2549854977161,\n        \"max\": 33.0,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
24.9011113658796,\n          23.5532191008226,\n
23.452082763333\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0422\\u0435\\u043c\\u043f\\u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\
u0430 \\u0432\\u0441\\u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 40.9432599529233,\n        \"min\": 100.0,\n        \"max\":
413.273418243566,\n        \"num_unique_values\": 1003,\n
\"samples\": [\n          286.701064731226,\n
397.151290856661,\n          269.840299552991\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u0430\\u044f \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u0433/\\u043c2\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 281.31469023666097,\n        \"min\":
0.603739925153945,\n        \"max\": 1399.54236233989,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
443.164680639388,\n          438.290234061128,\n
268.217818421553\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438 \\u043f\\u0440\\u0438 \\
u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\u0435\\u043d\\u0438\\u0438,
\\u0413\\u041f\\u0430\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.118982889469302,\n        \"min\":
64.0540605597917,\n        \"max\": 82.682051035271,\n
\"num_unique_values\": 1004,\n        \"samples\": [\n
75.1622076006716,\n          75.3033941703679,\n
72.1972680303294\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
485.62800627853596,\n        \"min\": 1036.85660535,\n        \"max\":
3848.43673187618,\n        \"num_unique_values\": 1004,\n
\"samples\": [\n          2948.96846718064,\n
2751.99820688163,\n          1904.38711713426\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0442\\u0440\\
u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\u0435 \\u0441\\u043c\\
u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 59.73593087332353,\n
\"min\": 33.8030255329625,\n        \"max\": 414.590628361534,\n
\"num_unique_values\": 1003,\n        \"samples\": [\n
180.068188823332,\n          196.82222775264,\n
```

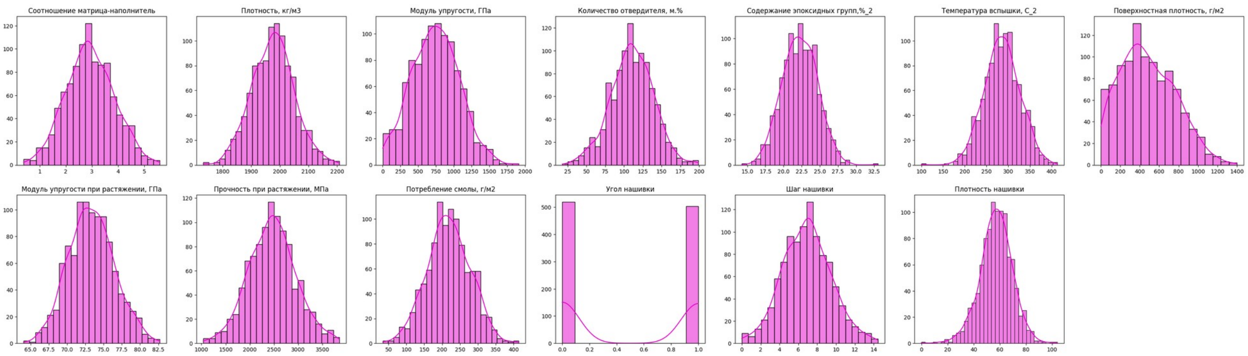
```
190.875279016224\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u0423\\u0433\\u043e\\u043b \\u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          1,\n          0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u0428\\u0430\\u0433 \\u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 2.5634670728338826,\n        \"min\": 0.0,\n        \"max\": 14.4405218753969,\n        \"num_unique_values\": 989,\n        \"samples\": [\n          8.56581217318028,\n          2.2156519282891\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c \\u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 12.350968798651312,\n        \"min\": 0.0,\n        \"max\": 103.988901301494,\n        \"num_unique_values\": 988,\n        \"samples\": [\n          43.9076879570936,\n          41.2464520255753\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe","variable_name":"df_X"}
```


```python
# проверка на наличие дубликатов
dpl = df_X.duplicated().sum()
if dpl == 0:
    print('Дубликатов нет')
else:
    print(f'Количество дубликатов в датасете: {dpl}')
```

```
Дубликатов нет
```

```python
# отображение гистограмм распределения параметров

# размер сетки
r = 7
c = 7
pl_c = 1 # счетчик
plt.figure(figsize = (40, 40))
for col in df_X.columns:
    plt.subplot(r, c, pl_c)
    sns.histplot(data = df_X[col], kde=True, color=(0.9, 0, 0.8))
    plt.xlabel(None)
    plt.ylabel(None)
    plt.title(col, size = 12)
    pl_c += 1

# можно предположить наличие выбросов у параметров плотности,
```

```
содержания эпоксидных групп, температуры вспышки и плотности нашивки
# визуально все параметры, за исключением угла нашивки, приближены к
нормальному распределению
```



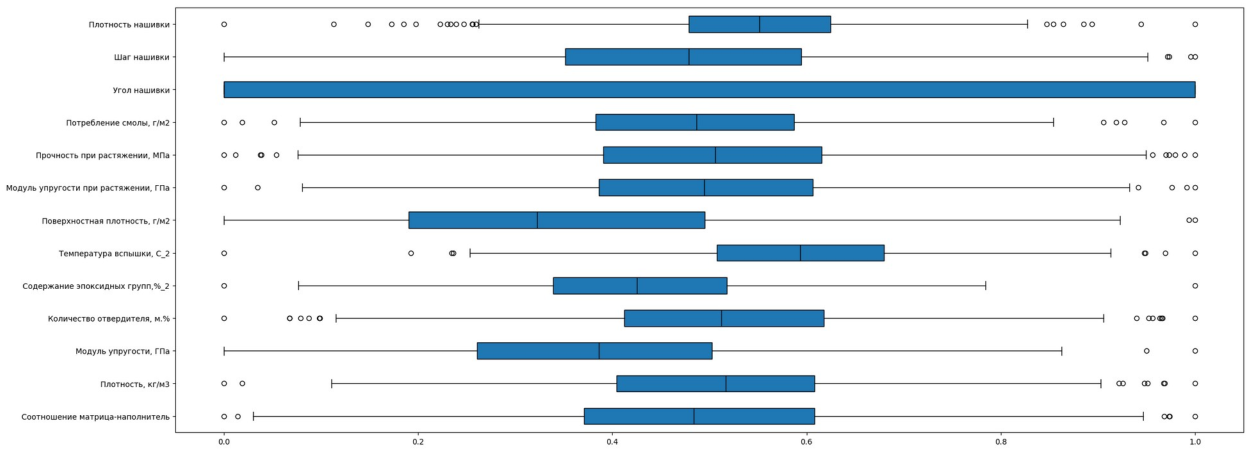

```
# "Необходимо также для каждой колонке получить среднее, медианное
значение"
df_X.describe().loc[['mean', '50%']]
# для большинства параметров (кроме поверхностной плотности и угла
нашивки) средние и медианные значения очень близки, что указывает на
симметричное распределение данных
```

{"summary":"{\n \"name\": \"# \\u0434\\u043b\\u044f \\u0431\\u043e\\
u043b\\u044c\\u0448\\u0438\\u043d\\u0441\\u0442\\u0432\\u0430 \\
u043f\\u0430\\u0440\\u0430\\u043c\\u0435\\u0442\\u0440\\u043e\\u0432
(\\u043a\\u0440\\u043e\\u043c\\u0435 \\u043f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u043e\\u0439 \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u0438 \\u0438
\\u0443\\u0433\\u043b\\u0430 \\u043d\\u0430\\u0448\\u0438\\u0432\\
u043a\\u0438) \\u0441\\u0440\\u0435\\u0434\\u043d\\u0438\\u0435 \\
u0438 \\u043c\\u0435\\u0434\\u0438\\u0430\\u043d\\u043d\\u044b\\u0435
\\u0437\\u043d\\u0430\\u0447\\u0435\\u043d\\u0438\\u044f \\u043e\\
u0447\\u0435\\u043d\\u044c \\u0431\\u043b\\u0438\\u0437\\u043a\\u0438,
\\u0447\\u0442\\u043e \\u0443\\u043a\\u0430\\u0437\\u044b\\u0432\\
u0430\\u0435\\u0442 \\u043d\\u0430 \\u0441\\u0438\\u043c\\u043c\\
u0435\\u0442\\u0440\\u0438\\u0447\\u043d\\u043e\\u0435 \\u0440\\
u0430\\u0441\\u043f\\u0440\\u0435\\u0434\\u0435\\u043b\\u0435\\u043d\\
u0438\\u0435 \\u0434\\u0430\\u043d\\u043d\\u044b\\u0445\",\n
\"rows\": 2,\n \"fields\": [\n   {\n   \"column\": \"\\u0421\\
u043e\\u043e\\u0442\\u043d\\u043e\\u0448\\u0435\\u043d\\u0438\\u0435
\\u043c\\u0430\\u0442\\u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\
u043f\\u043e\\u043b\\u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.016608611119472427,\n   \"min\": 2.90687765033521,\n
\"max\": 2.9303657734325483,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   2.90687765033521,\n
2.9303657734325483\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\

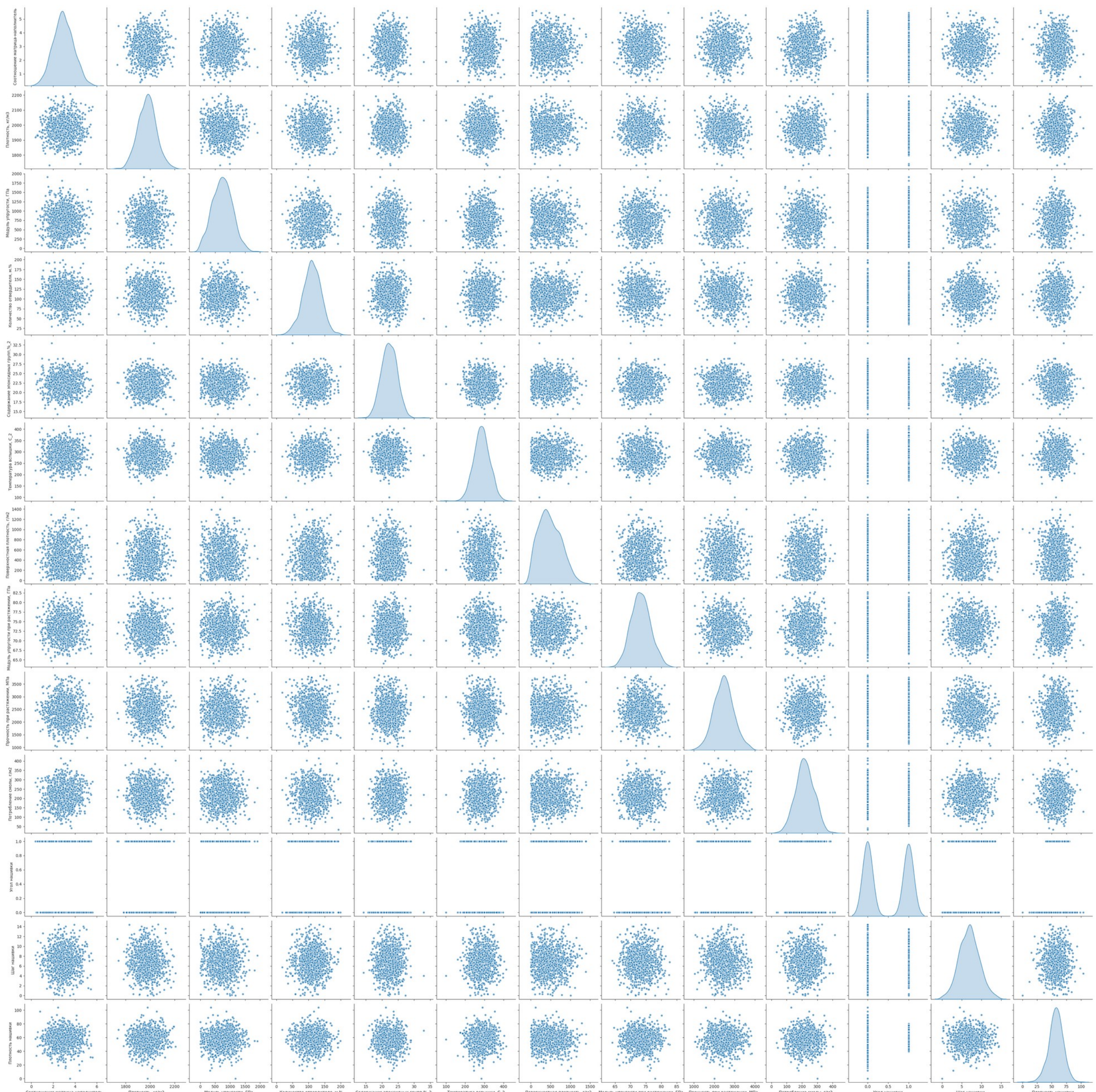
u043a\\u0433/\\u043c3\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 1.3341469284591576,\n   \"min\":
1975.7348881101545,\n   \"max\": 1977.62165679058,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
1977.62165679058,\n   1975.7348881101545\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u041c\\u043e\\u0434\\u0443\\
u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\
u0442\\u0438, \\u0413\\u041f\\u0430\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.18307352239336158,\n
\"min\": 739.664327697792,\n   \"max\": 739.9232327560721,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
739.664327697792,\n   739.9232327560721\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u041a\\u043e\\u043b\\u0438\\
u0447\\u0435\\u0441\\u0442\\u0432\\u043e \\u043e\\u0442\\u0432\\
u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\u043b\\u044f, \\u043c.%\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
0.004192261660671894,\n   \"min\": 110.564839894065,\n
\"max\": 110.57076864736254,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   110.564839894065,\n
110.57076864736254\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u0421\\u043e\\u0434\\u0435\\u0440\\u0436\\u0430\\u043d\\u0438\\u0435
\\u044d\\u043f\\u043e\\u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445
\\u0433\\u0440\\u0443\\u043f\\u043f,%_2\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.009649031876369868,\n
\"min\": 22.2307437560244,\n   \"max\": 22.24438954776773,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
22.2307437560244,\n   22.24438954776773\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u0422\\u0435\\u043c\\u043f\\
u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\u0430 \\u0432\\u0441\\
u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
0.010366878104705912,\n   \"min\": 285.88215135162187,\n
\"max\": 285.896812331237,\n   \"num_unique_values\": 2,\n
\"samples\": [\n   285.896812331237,\n
285.88215135162187\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\
u041f\\u043e\\u0432\\u0435\\u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\
u043d\\u0430\\u044f \\u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\
u0441\\u0442\\u044c, \\u0433/\\u043c2\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 21.826595838581802,\n
\"min\": 451.86436518306,\n   \"max\": 482.73183303841853,\n
\"num_unique_values\": 2,\n   \"samples\": [\n
451.86436518306,\n   482.73183303841853\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"\\u041c\\u043e\\u0434\\u0443\\

```
u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\
u0442\\u0438 \\u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\
u044f\\u0436\\u0435\\u043d\\u0438\\u0438, \\u0413\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.04226140756441898,\n        \"min\": 73.2688045943481,\n
\"max\": 73.32857125009068,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          73.2688045943481,\n
73.32857125009068\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5.231399904267147,\n        \"min\": 2459.52452600309,\n
\"max\": 2466.922842697902,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          2459.52452600309,\n
2466.922842697902\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u043e\\u0442\\u0440\\u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\
u0435 \\u0441\\u043c\\u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.548529966923402,\n        \"min\": 218.42314367654285,\n
\"max\": 219.198882195134,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          219.198882195134,\n
218.42314367654285\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0423\\u0433\\u043e\\u043b \\u043d\\u0430\\u0448\\u0438\\u0432\\
u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.347678114307755,\n        \"min\":
0.0,\n        \"max\": 0.4916911045943304,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          0.0,\n
0.4916911045943304\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0428\\u0430\\u0433 \\u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\
u0438\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.011965504167338914,\n        \"min\": 6.8992220776750175,\n
\"max\": 6.9161438559491,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          6.9161438559491,\n
6.8992220776750175\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043d\\u0430\\u0448\\u0438\\u0432\\u043a\\u0438\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.1329292966331075,\n        \"min\": 57.153929432857645,\n
\"max\": 57.3419198469929,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          57.3419198469929,\n
57.153929432857645\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```
# диаграммы "ящика с усами" - проверка на выбросы

sc = MinMaxScaler()
sc.fit(df_X)
plt.figure(figsize = (25, 10))
plt.boxplot(pd.DataFrame(sc.transform(df_X)), labels = df_X.columns,
patch_artist = True, vert = False, medianprops = dict(color =
'black'))
plt.show()
```

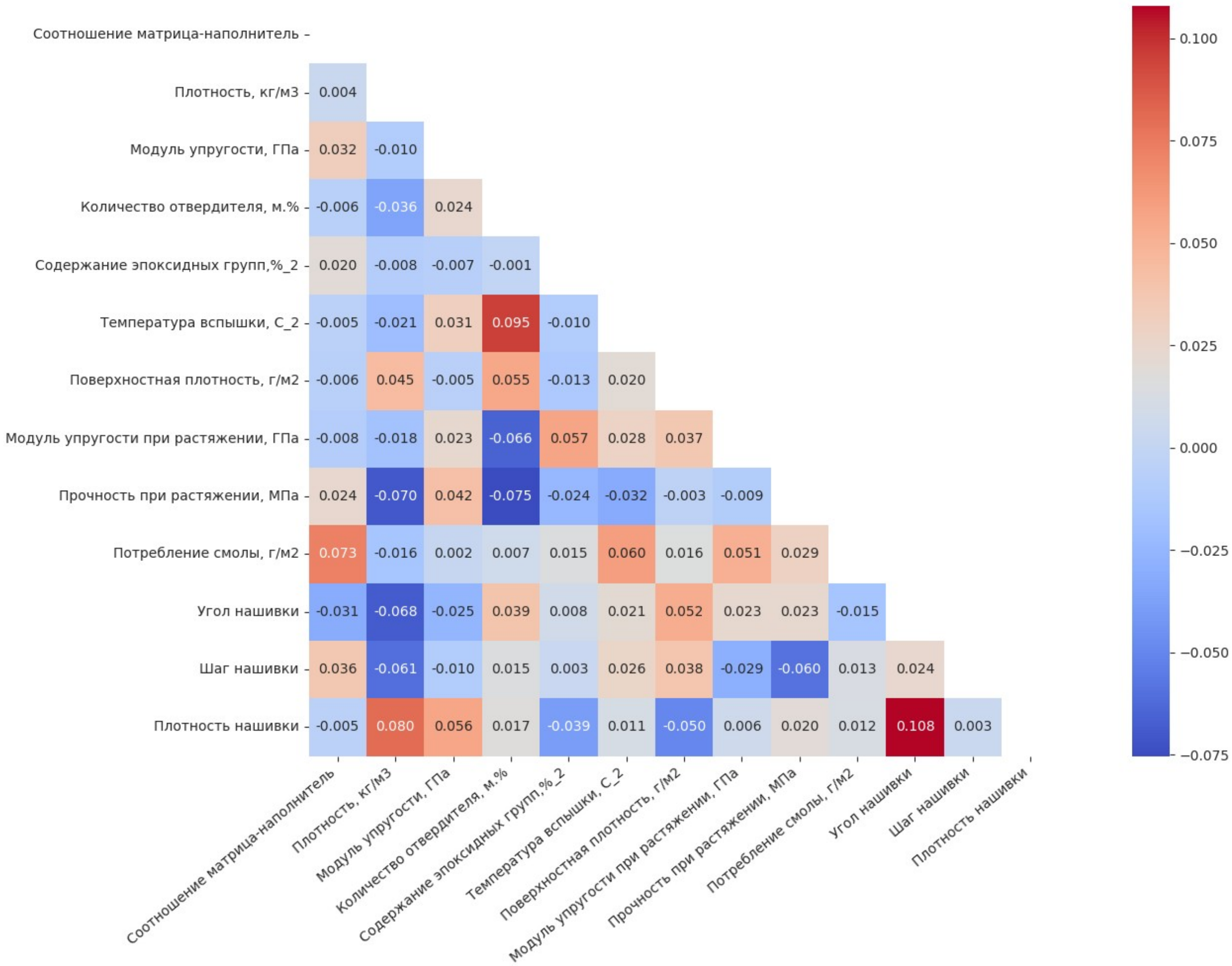



```
# как видно из диаграмм, выбросы присутствуют во всех параметрах
(кроме угла нашивки), далее необходимо построить попарные графики
рассеяния точек
sns.set_style('ticks')
sns.pairplot(df_X,
             diag_kind = 'kde',  # использование ядерной оценки
плотности на диагонали
             kind = 'scatter',   # тип графика рассеяния
             plot_kws = dict(s=30, alpha=0.7, edgecolor="w",
linewidth=1),
             diag_kws = dict(shade=True),  # параметры диагональных
графиков
             height = 3,  # высота каждого подграфика
             aspect = 1)

<seaborn.axisgrid.PairGrid at 0x79780f57ea10>
```



```
# как можно увидеть из графиков, не наблюдаются четкие линейные
зависимостей между парами параметров, то есть изменения одного
параметра не приводят к пропорциональным изменениям другого параметра
# простроим тепловую корреляционнуую карту
plt.figure(figsize=(15, 10))
sns.heatmap(
        df_X.corr(),
        mask = np.triu(df_X.corr()),
        annot=True,
        fmt=".3f",
        square = True,
        cmap="coolwarm",
```

```
        cbar=True
     )
plt.xticks(rotation=40, ha = 'right')
plt.subplots_adjust(right=3)
plt.tight_layout()
plt.show()
```




```
# низкие значения (максимальное значение составляет 0,108 - корреляция
между плотностьюн нашивки и углом нашивки) свидетельствуют об
отсутствии сильных корреляционных связей между переменными

# метод 3-х сигм
def outliers_z_score(df_X):
    outliers = {}
    for column in df_X.columns:
        mean = df_X[column].mean()
        std_dev = df_X[column].std()
        z_scores = (df_X[column] - mean) / std_dev
        outliers[column] = len(z_scores[np.abs(z_scores) > 3])
```

```
    return outliers

# метод межквартильных расстояний (IQR)
def outliers_iqr(df_X):
    outliers = {}
    for column in df_X.columns:
        q1 = df_X[column].quantile(0.25)
        q3 = df_X[column].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        outliers[column] = len(df_X[(df_X[column] < lower_bound) |
(df_X[column] > upper_bound)])
    return outliers

outliers_z_score = outliers_z_score(df_X)
outliers_iqr = outliers_iqr(df_X)

comparison_df = pd.DataFrame({
    'Метод 3-х сигм': outliers_z_score,
    'Метод IQR': outliers_iqr
})

print("Результаты выявления выбросов:")
print(comparison_df)
```

```
Результаты выявления выбросов:
                                      Метод 3-х сигм  Метод IQR
Соотношение матрица-наполнитель                    0          6
Плотность, кг/м3                                   3          9
Модуль упругости, ГПа                              2          2
Количество отвердителя, м.%                        2         14
Содержание эпоксидных групп,%_2                    2          2
Температура вспышки, С_2                           3          8
Поверхностная плотность, г/м2                      2          2
Модуль упругости при растяжении, ГПа               0          6
Прочность при растяжении, МПа                      0         11
Потребление смолы, г/м2                            3          8
Угол нашивки                                       0          0
Шаг нашивки                                        0          4
Плотность нашивки                                  7         21
```

```
# метод межквартильных расстояний захватывает бОльшую долю выбросов,
нежели метод 3-х сигм, в связи с чем был выбран именно первый способ,
отсекающий все выбросы
def remove_outliers_iqr(df_X):
    df = df_X.copy()
    for column in df.columns:
        q1 = df_X[column].quantile(0.25)
        q3 = df_X[column].quantile(0.75)
```

```
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        df = df[(df_X[column] >= lower_bound) & (df_X[column] <=
upper_bound)]
    return df

df = remove_outliers_iqr(df_X)

print(f'Размерность датасета до отсечения выбросов {df_X.shape}, после
- {df.shape}, были удалены {df_X.shape[0] - df.shape[0]} выбросов')
print('\n')
df
```

```
Размерность датасета до отсечения выбросов (1023, 13), после - (936,
13), были удалены 87 выбросов
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 936,\n \"fields\": [\n   {\n   \"column\": \"\\u0421\\u043e\\u043e\\u0442\\u043d\\u043e\\u0448\\u0435\\u043d\\u0438\\u0435 \\u043c\\u0430\\u0442\\u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\u043f\\u043e\\u043b\\u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8937115835492558,\n   \"min\": 0.547391007365624,\n   \"max\": 5.3141436851035,\n   \"num_unique_values\": 930,\n   \"samples\": [\n   3.27902264127646,\n   4.44914679844715,\n   2.66769692914197\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\u043a\\u0433/\\u043c3\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 70.80812042119867,\n   \"min\": 1784.48224524858,\n   \"max\": 2161.56521646388,\n   \"num_unique_values\": 929,\n   \"samples\": [\n   1959.08522780269,\n   1968.40896282451,\n   1968.40138797479\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\u0442\\u0438, \\u0413\\u041f\\u0430\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 328.70866543439183,\n   \"min\": 2.4369087535075,\n   \"max\": 1649.41570587587,\n   \"num_unique_values\": 935,\n   \"samples\": [\n   429.919091587865,\n   105.154466533956,\n   340.827747445295\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u041a\\u043e\\u043b\\u0438\\u0447\\u0435\\u0441\\u0442\\u0432\\u043e \\u043e\\u0442\\u0432\\u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\u043b\\u044f, \\u043c.%\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\":

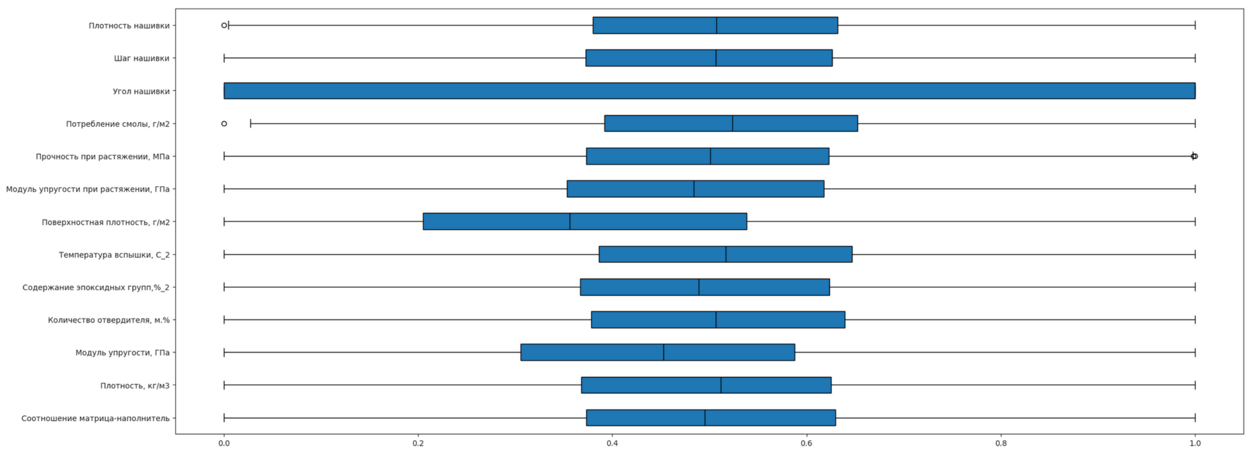
```
27.037890759871665,\n        \"min\": 38.6685003343557,\n
\"max\": 181.82844779488,\n        \"num_unique_values\": 919,\n
\"samples\": [\n          79.3468498323662,\n
107.549456408651,\n          115.168415221245\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u0421\\u043e\\u0434\\u0435\\
u0440\\u0436\\u0430\\u043d\\u0438\\u0435 \\u044d\\u043f\\u043e\\
u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445 \\u0433\\u0440\\
u0443\\u043f\\u043f,%_2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.3948709278370313,\n
\"min\": 15.6958938036288,\n        \"max\": 28.9550943746499,\n
\"num_unique_values\": 919,\n        \"samples\": [\n
21.874256956087,\n          17.3005393046688,\n
22.4632663490591\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0422\\u0435\\u043c\\u043f\\u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\
u0430 \\u0432\\u0441\\u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 39.400676670895265,\n        \"min\": 179.37439137039,\n
\"max\": 386.067991779505,\n        \"num_unique_values\": 918,\n
\"samples\": [\n          321.184936349113,\n
292.625715602231,\n          288.044440777829\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u0430\\u044f \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u0433/\\u043c2\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 280.19037672865227,\n        \"min\":
0.603739925153945,\n        \"max\": 1291.34011463545,\n
\"num_unique_values\": 920,\n        \"samples\": [\n
143.030854441705,\n          126.591897067563,\n
1133.44794057603\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438 \\u043f\\u0440\\u0438 \\
u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\u0435\\u043d\\u0438\\u0438,
\\u0413\\u041f\\u0430\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.037380544281799,\n        \"min\":
65.5533359100584,\n        \"max\": 81.4171259444738,\n
\"num_unique_values\": 920,\n        \"samples\": [\n
68.8304957365155,\n          69.1750753574035,\n
73.6899970800352\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
463.8389109845878,\n        \"min\": 1250.39280220501,\n
\"max\": 3705.67252336168,\n        \"num_unique_values\": 920,\n
```

```
\"samples\": [\n          2434.85721385732,\n
2029.26595658323,\n          2859.46492693206\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0442\\u0440\\
u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\u0435 \\u0441\\u043c\\
u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 57.82725543406986,\n
\"min\": 63.6856983307984,\n        \"max\": 359.052219789673,\n
\"num_unique_values\": 919,\n        \"samples\": [\n
174.349764717947,\n          160.115494585427,\n
317.257168827879\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0423\\u0433\\u043e\\u043b \\u043d\\u0430\\u0448\\u0438\\u0432\\
u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"\\u0428\\u0430\\u0433 \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.5096724756644173,\n        \"min\":
0.0376389366987437,\n        \"max\": 13.732404403383,\n
\"num_unique_values\": 904,\n        \"samples\": [\n
4.37830659592764,\n          7.8632361533103\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\
u043d\\u043e\\u0441\\u0442\\u044c \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11.239330624406607,\n        \"min\":
27.2729280063294,\n        \"max\": 86.0124270098611,\n
\"num_unique_values\": 903,\n        \"samples\": [\n
52.3916690992692,\n          62.2023085103011\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
def remove_outliers_iqr1(df):
    df1 = df.copy()
    for column in df.columns:
        q1 = df[column].quantile(0.25)
        q3 = df[column].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        df1 = df1[(df[column] >= lower_bound) & (df[column] <=
upper_bound)]
    return df1

df1 = remove_outliers_iqr(df)

print(f'Размерность датасета до второго отсечения выбросов {df.shape},
```

```
после - {df1.shape}, были удалены {df.shape[0] - df1.shape[0]}
выбросов')
print('\n')

sc.fit(df1)
plt.figure(figsize = (25, 10))
plt.boxplot(pd.DataFrame(sc.transform(df1)), labels = df_X.columns,
patch_artist = True, vert = False, medianprops = dict(color =
'black'))
plt.show()
```

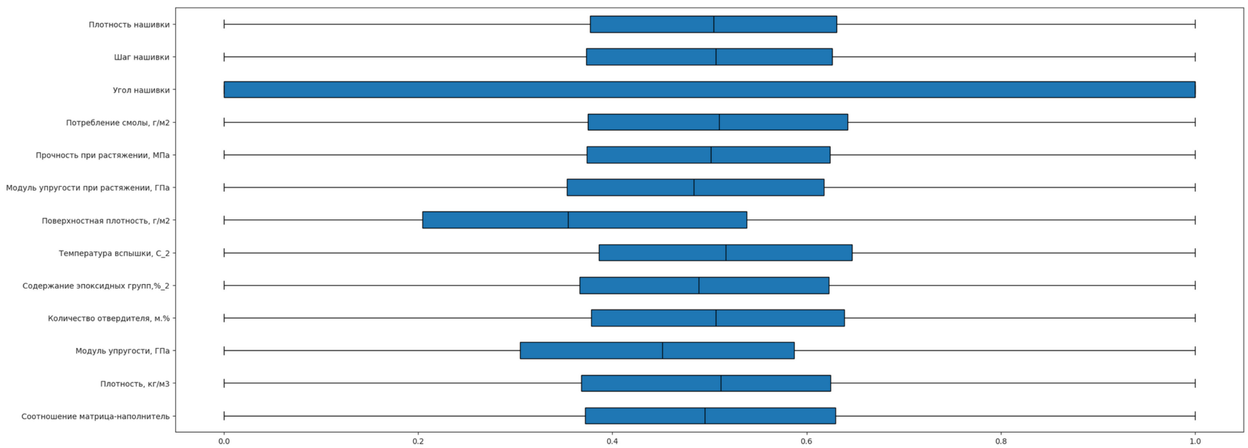
```
Размерность датасета до второго отсечения выбросов (936, 13), после -
(926, 13), были удалены 10 выбросов
```



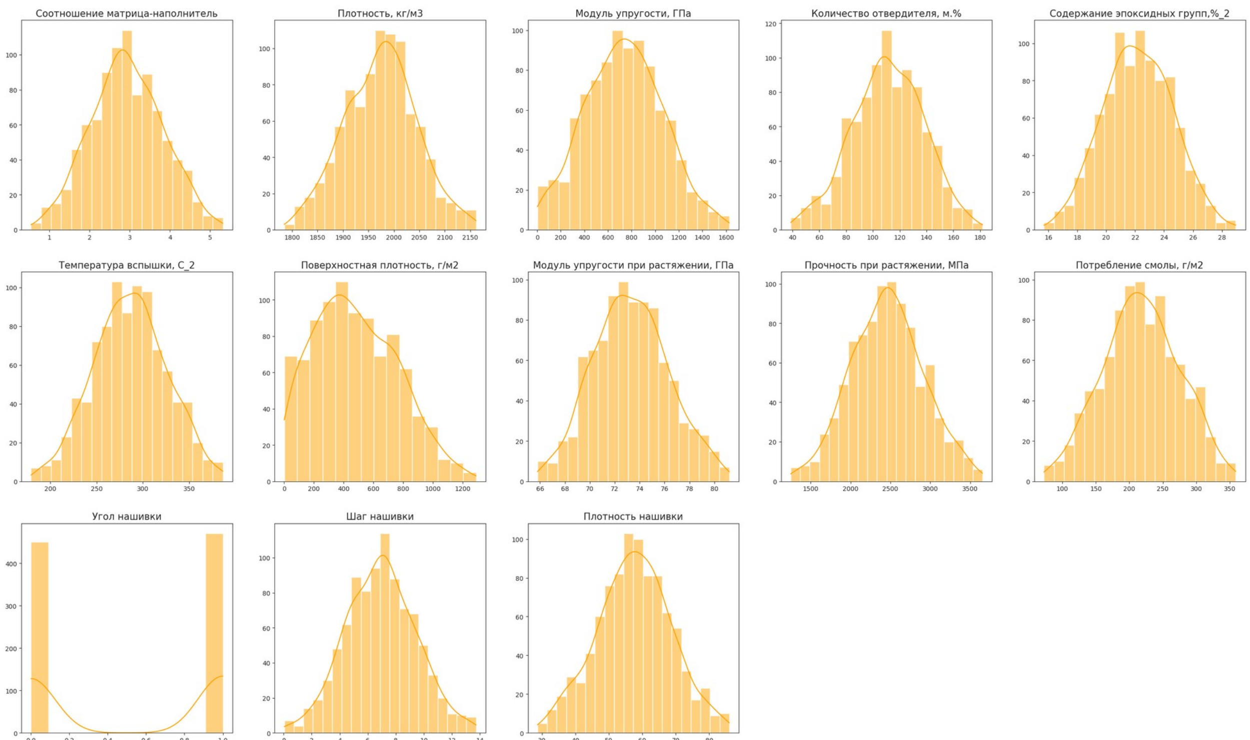

```
# большая часть выбросов была исключена, но они до сих пор
присутствуют в датасете

def remove_outliers_iqr1(df1):
    df2 = df1.copy()
    for column in df.columns:
        q1 = df1[column].quantile(0.25)
        q3 = df1[column].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        df2 = df1[(df1[column] >= lower_bound) & (df1[column] <=
upper_bound)]
    return df2

df2 = remove_outliers_iqr(df1)

print(f'Размерность датасета до второго отсечения выбросов
{df1.shape}, после - {df2.shape}, были удалены {df1.shape[0] -
df2.shape[0]} выбросов')
```

```
print('\n')

sc.fit(df2)
plt.figure(figsize = (25, 10))
plt.boxplot(pd.DataFrame(sc.transform(df2)), labels = df_X.columns,
patch_artist = True, vert = False, medianprops = dict(color =
'black'))
plt.show()
```

```
Размерность датасета до второго отсечения выбросов (926, 13), после -
(922, 13), были удалены 4 выбросов
```




```
# готово, теперь наш датасет не имеет выбросов
df = df2

# отобразим гистрограммы каждого из параметров без нормализации, в
исходных единицах
r = 5
c = 5
pl_c = 1
plt.figure(figsize=(35,35))
for col in df.columns:
    plt.subplot(r, c, pl_c)
    sns.histplot(data = df[col], kde = True, color = "orange")
    plt.xlabel(None)
    plt.ylabel(None)
    plt.title(col, size = 15)
    pl_c += 1

# все параметры стремятся к нормальному распределению (опять же, за
исключением угла нашивки)
```




```
# готовый для дальнейшего анализа датасет сохраняем в папку
df.to_excel('df_norm.xlsx')
df = pd.read_excel('/content/drive/MyDrive/DatasetsVKR/df_norm.xlsx')
df
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 922,\n \"fields\": [\n   {\n   \"column\": \"Unnamed: 0\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 266,\n   \"min\": 1,\n   \"max\": 922,\n   \"num_unique_values\": 922,\n   \"samples\": [\n   320,\n   378,\n   539\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u0421\\u043e\\u043e\\u0442\\u043d\\u043e\\u0448\\u0435\\u043d\\u0438\\u0435 \\u043c\\u0430\\u0442\\u0440\\u0438\\u0446\\u0430-\\u043d\\u0430\\u043f\\u043e\\u043b\\u043d\\u0438\\u0442\\u0435\\u043b\\u044c\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.8954717320562058,\n   \"min\": 0.547391007365624,\n   \"max\": 5.3141436851035,\n   \"num_unique_values\": 916,\n   \"samples\": [\n   2.71722263251102,\n   0.968086542252719,\n   3.72306752097199\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"\\u041f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\u043a\\u0433/\\u043c3\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 71.04064831235215,\n   \"min\": 1784.48224524858,\n   \"max\": 2161.56521646388,\n   \"num_unique_values\": 915,\n   \"samples\": [\n   2026.11047511909,\n

```
1899.37285197669,\n          1986.9445606882\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041c\\u043e\\u0434\\u0443\\
u043b\\u044c \\u0443\\u043f\\u0440\\u0443\\u0433\\u043e\\u0441\\
u0442\\u0438, \\u0413\\u041f\\u0430\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 327.60700751220486,\n
\"min\": 2.4369087535075,\n        \"max\": 1628.0,\n
\"num_unique_values\": 921,\n        \"samples\": [\n
718.492637623506,\n          1001.28579276492,\n
597.1415168851 8\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041a\\u043e\\u043b\\u0438\\u0447\\u0435\\u0441\\u0442\\u0432\\u043e
\\u043e\\u0442\\u0432\\u0435\\u0440\\u0434\\u0438\\u0442\\u0435\\
u043b\\u044f, \\u043c.%\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 26.753227905108954,\n
\"min\": 38.6685003343557,\n        \"max\": 181.82844779488,\n
\"num_unique_values\": 905,\n        \"samples\": [\n
126.919151502633,\n          143.535238468246,\n
57.9497800545227\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0421\\u043e\\u0434\\u0435\\u0440\\u0436\\u0430\\u043d\\u0438\\u0435
\\u044d\\u043f\\u043e\\u043a\\u0441\\u0438\\u0434\\u043d\\u044b\\u0445
\\u0433\\u0440\\u0443\\u043f\\u043f,%_2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 2.3939260328968763,\n
\"min\": 15.6958938036288,\n        \"max\": 28.9550943746499,\n
\"num_unique_values\": 905,\n        \"samples\": [\n
21.2453708727899,\n          22.0241086190884,\n
25.3902830373374\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0422\\u0435\\u043c\\u043f\\u0435\\u0440\\u0430\\u0442\\u0443\\u0440\\
u0430 \\u0432\\u0441\\u043f\\u044b\\u0448\\u043a\\u0438, \\u0421_2\",\
n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 39.42076363132895,\n        \"min\": 179.37439137039,\n
\"max\": 386.067991779505,\n        \"num_unique_values\": 904,\n
\"samples\": [\n          206.005127700645,\n
248.132367449544,\n          271.126312461753\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0432\\u0435\\
u0440\\u0445\\u043d\\u043e\\u0441\\u0442\\u043d\\u0430\\u044f \\
u043f\\u043b\\u043e\\u0442\\u043d\\u043e\\u0441\\u0442\\u044c, \\
u0433/\\u043c2\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 280.437328786617,\n        \"min\":
0.603739925153945,\n        \"max\": 1291.34011463545,\n
\"num_unique_values\": 906,\n        \"samples\": [\n
660.247949897597,\n          384.207155178403,\n
334.700978120372\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041c\\u043e\\u0434\\u0443\\u043b\\u044c \\u0443\\u043f\\u0440\\
u0443\\u0433\\u043e\\u0441\\u0442\\u0438 \\u043f\\u0440\\u0438 \\
```

```
u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\u0435\\u043d\\u0438\\u0438,
\\u0413\\u041f\\u0430\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3.0258643883622773,\n        \"min\":
65.7938449666054,\n        \"max\": 81.203146720828,\n
\"num_unique_values\": 906,\n        \"samples\": [\n
72.3662210329381,\n          70.916879218537,\n
78.0716510274565\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u041f\\u0440\\u043e\\u0447\\u043d\\u043e\\u0441\\u0442\\u044c \\
u043f\\u0440\\u0438 \\u0440\\u0430\\u0441\\u0442\\u044f\\u0436\\
u0435\\u043d\\u0438\\u0438, \\u041c\\u041f\\u0430\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
453.5647335545092,\n        \"min\": 1250.39280220501,\n
\"max\": 3654.43435901371,\n        \"num_unique_values\": 906,\n
\"samples\": [\n          2618.95064622698,\n
3636.8929917828,\n          2174.61538963401\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043e\\u0442\\u0440\\
u0435\\u0431\\u043b\\u0435\\u043d\\u0438\\u0435 \\u0441\\u043c\\
u043e\\u043b\\u044b, \\u0433/\\u043c2\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 57.13747497853371,\n
\"min\": 72.5308733761696,\n        \"max\": 359.052219789673,\n
\"num_unique_values\": 905,\n        \"samples\": [\n
208.929445658,\n          239.457167408696,\n
227.924477322423\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"\\
u0423\\u0433\\u043e\\u043b \\u043d\\u0430\\u0448\\u0438\\u0432\\
u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"\\u0428\\u0430\\u0433 \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.5141838419664064,\n        \"min\":
0.0376389366987437,\n        \"max\": 13.732404403383,\n
\"num_unique_values\": 891,\n        \"samples\": [\n
10.0671231646161,\n          6.80636570246087\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"\\u041f\\u043b\\u043e\\u0442\\
u043d\\u043e\\u0441\\u0442\\u044c \\u043d\\u0430\\u0448\\u0438\\
u0432\\u043a\\u0438\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11.122204405356028,\n        \"min\":
28.6616316278123,\n        \"max\": 86.0124270098611,\n
\"num_unique_values\": 890,\n        \"samples\": [\n
77.2777281396634,\n          74.8856664463349\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```

```
a = df.describe()
a.T.loc[:, ['min', 'max']].drop(['Unnamed: 0'], axis = 0)
```

{"summary":"{\n \"name\": \"a\",\n \"rows\": 13,\n \"fields\": [\n   {\n   \"column\": \"min\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 568.7945007672928,\n   \"min\": 0.0,\n   \"max\": 1784.48224524858,\n   \"num_unique_values\": 13,\n   \"samples\": [\n   0.0376389366987437,\n   72.5308733761696,\n   0.547391007365624\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"max\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1124.1569511505072,\n   \"min\": 1.0,\n   \"max\": 3654.43435901371,\n   \"num_unique_values\": 13,\n   \"samples\": [\n   13.732404403383,\n   359.052219789673,\n   5.3141436851035\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}