

Out[10]:	Соотношение матрица-наполнитель Плотность, кг/м3 модуль упругости, ГПа Количество отвердителя, м.% Содержание эпоксидных групп,%_2 Температура вспышки, С_2 Поверхностная плотность, г/м2 Модуль упругости при растяжении, ГПа Прочность при растяжении, МПа Потребление смолы, г/м2 Угол нашивки, град Шаг нашивки Плотность нашивки
0	1.857143 2030.000000 738.736842 30.000000 22.267857 100.000000 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 57.000000
1	1.857143 2030.000000 738.736842 50.000000 23.750000 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 60.000000
2	1.857143 2030.000000 738.736842 49.900000 33.000000 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 70.000000
3	1.857143 2030.000000 738.736842 129.000000 21.500000 300.000000 210.000000 70.000000 3000.000000 220.000000 0.0 5.000000 47.000000
4	2.771331 2030.000000 751.000000 111.860000 22.267857 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 5.000000 57.000000
...	...
1018	2.271346 1952.087902 912.855545 86.992183 20.232349 324.774576 209.198700 73.090961 2387.292495 125.007669 90.0 9.076380 47.019770
1019	3.444022 2050.089171 444.732634 145.981978 19.599769 254.215401 350.660830 72.920827 2360.392784 117.730099 90.0 10.565614 53.750790
1020	3.280604 1972.372865 416.836524 110.533477 23.957502 248.423047 740.142791 74.734344 2662.906040 236.606764 90.0 4.161154 67.629684
1021	3.705351 2065.797773 741.475517 141.397963 19.246945 275.779840 641.469152 74.042708 2071.715856 197.126067 90.0 6.313201 58.261074
1022	3.808020 1890.413468 417.316232 129.183416 27.474763 300.952708 758.747882 74.309704 2856.328932 194.754342 90.0 6.078902 77.434468

1023 rows × 13 columns

In [11]: #Проверка информации о датасете, проблем с типами данных в каждом столбце (типы признаков)
df.info()
Все переменные содержат значения float64, качественные характеристики отсутствуют. Пропусков не имеется. Ни одна из записей не является NaN, очистка не требуется. Объединенный файл имеет всего 1023 строки.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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   float64 
 11  Шаг нашивки                      1023 non-null   float64 
 12  Плотность нашивки                1023 non-null   float64 
dtypes: float64(13)
memory usage: 111.9 KB
```

In [12]: #Посчитать уникальных значений с помощью функции nunique

df.nunique() #Видим в основной общей число уникальных значений в каждом столбце, но в столбце "Угол нашивки" всего 2 значения. Поработаем с ним.

Out[12]: Соотношение матрица-наполнитель 1014
Плотность, кг/м3 1013
модуль упругости, ГПа 1028
Количество отвердителя, м.% 1085
Содержание эпоксидных групп,%_2 1094
Температура вспышки, С_2 1093
Поверхностная плотность, г/м2 1086
Модуль упругости при растяжении, ГПа 1084
Прочность при растяжении, МПа 1094
Потребление смолы, г/м2 1093
угол нашивки, град 2
шаг нашивки 989
плотность нашивки 988
dtype: int64

In [13]: #Поработаем со столбом "Угол нашивки"

In [14]: df['Угол нашивки, град'].nunique() #Для тех как кол-во уникальных значений в колонке Угол нашивки равно 2, можем привести данные в этой колонке к значениям 0 и 1

Out[14]: 2

In [15]: #Приведем кол-во элементов, где Угол нашивки равен 0 градусов

df['Угол нашивки, град'][df['Угол нашивки, град'] == 0].count()

Out[15]: 528

In [16]: #Приведем столбец "Угол нашивки" к значениям 0 и 1 и integer

df.replace({'Угол нашивки, град': {0.0 : 0, 90.0 : 1}})

df['Угол нашивки, град'] = df['Угол нашивки, град'].astype(int)

Out[15]:

In [17]: #Переименование столбцов

df = df.rename(columns={'Угол нашивки, град' : 'Угол нашивки'})

Out[17]:

Соотношение матрица-наполнитель Плотность, кг/м3 модуль упругости, ГПа Количество отвердителя, м.% Содержание эпоксидных групп,%_2 Температура вспышки, С_2 Поверхностная плотность, г/м2 Модуль упругости при растяжении, ГПа Прочность при растяжении, МПа Потребление смолы, г/м2 Угол нашивки Шаг нашивки Плотность нашивки	
0 1.857143 2030.000000 738.736842 30.000000 22.267857 100.000000 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 57.000000	
1 1.857143 2030.000000 738.736842 50.000000 23.750000 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 60.000000	
2 1.857143 2030.000000 738.736842 49.900000 33.000000 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 4.000000 70.000000	
3 1.857143 2030.000000 738.736842 129.000000 21.500000 300.000000 210.000000 70.000000 3000.000000 220.000000 0.0 5.000000 47.000000	
4 2.771331 2030.000000 751.000000 111.860000 22.267857 284.615385 210.000000 70.000000 3000.000000 220.000000 0.0 5.000000 57.000000	
...	...
1018 2.271346 1952.087902 912.855545 86.992183 20.232349 324.774576 209.198700 73.090961 2387.292495 125.007669 90.0 9.076380 47.019770	
1019 3.444022 2050.089171 444.732634 145.981978 19.599769 254.215401 350.660830 72.920827 2360.392784 117.730099 90.0 10.565614 53.750790	
1020 3.280604 1972.372865 416.836524 110.533477 23.957502 248.423047 740.142791 74.734344 2662.906040 236.606764 90.0 4.161154 67.629684	
1021 3.705351 2065.797773 741.475517 141.397963 19.246945 275.779840 641.469152 74.042708 2071.715856 197.126067 90.0 6.313201 58.261074	
1022 3.808020 1890.413468 417.316232 129.183416 27.474763 300.952708 758.747882 74.309704 2856.328932 194.754342 90.0 6.078902 77.434468	

1023 rows × 13 columns

In [18]: #Посчитаем количество элементов, где угол нашивки равен 0 градусов и убедимся, что количество не изменилось после наших манипуляций
df['Угол нашивки'][df['Угол нашивки'] == 0].count()

#После преобразования колонки Угол нашивки к значениям 0 и 1, кол-во элементов, где угол нашивки равен 0 не изменилось (520 до и после преобразования)

Out[18]: 520

In [19]: #Переведем столбцы с нумерацией 0 integer

df.index = df.index.astype('int')

Out[19]:

In [20]: #Сохраним итоговый датасет в отдельную папку с данными, чтобы долго не искался

df.to_excel("Itoig.xlsx")

Out[20]:

In [21]: #Изучим описательную статистику наших данных (исключительное, минимальное, квартили, медиана, стандартное отклонение, среднее значение и т.д.), посмотрим на основные параметры анализа данных

df.describe()

Соотношение матрица-наполнитель Плотность, кг/м3 модуль упругости, ГПа Количество отвердителя, м.% Содержание эпоксидных групп,%_2 Температура вспышки, С_2 Поверхностная плотность, г/м2 Модуль упругости при растяжении, ГПа Прочность при растяжении, МПа Потребление смолы, г/м2 Угол нашивки Шаг нашивки Плотность нашивки
count 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000 1023.000000
mean 2.930366 1975.734088 739.923233 110.507069 22.244390 285.882151 482.731833 73.328571 2466.922843 218.423144 0.491691 6.89922 57.153929
std 0.913222 73.729231 330.231581 28.295911 2.406301 40.94260 281.314690 3.118983 485.628006 59.735931 0.500175 2.563467 12.350969
min 0.384043 1731.646335 2.436909 17.740275 14.254985 100.000000 0.603740 64.054061 1036.856605 33.803026 0.000000 0.000000
25% 2.317887 1924.155467 500.047452 92.443497 20.608034 259.06528 266.816645 71.245018 2135.850448 179.627520 0.000000 5.080033 49.799212
50% 2.906878 1977.617657 739.64328 110.564840 22.230744 285.896812 451.864365 73.268095 2459.524526 219.198862 0.000000 6.916144 57.341920
75% 3.552660 2021.734735 961.812526 129.733066 23.961934 313.002106 693.225017 75.356512 276.7193119 1.000000 8.58629

• max - максимум

In [23]: # Пропуски данных

```
# Пробирки не пропущены данные
df.isnull().sum()
```

```
# пропущенных данных нет = нулюхих значений нет, очистка не требуется
```

Out[24]:

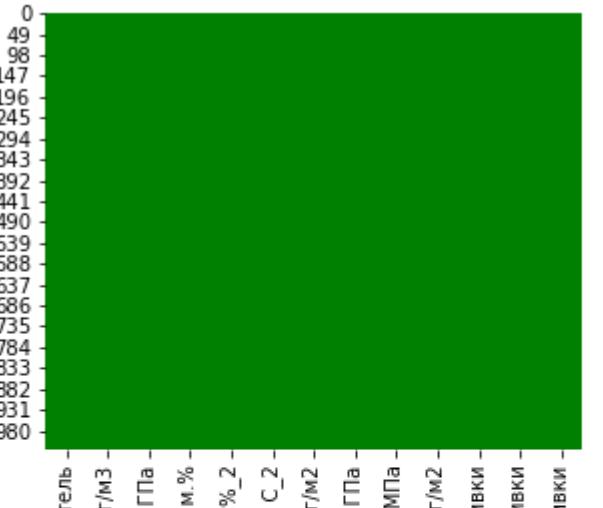
```
Соотношение матрица-наполнитель 0
Плотность, кг/м3 0
модуль упругости, ГПа 0
Количество отвердителя, м.% 0
Содержание эпоксидных групп,%_2 0
Температура вспышки, С_2 0
Поверхностная плотность, г/м2 0
Модуль упругости при растяжении, ГПа 0
Прочность при растяжении, МПа 0
Потребление смолы, г/м2 0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

In [25]: #всемо -зеленый - не пропущенные, темнозеленый - пропущенные данные

```
cols = df.columns
colours = ['#cfcfff', '#000000']
sns.heatmap(df[cols].isnull(), cmap = sns.color_palette(colours))
```

```
#Тепловая карта, так же как info() и функция ISNULL() показывает, что пропусков нет.
```

Out[25]:



In [26]:

```
for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    print('О %' + str(np.round(pct_missing*100)))
```

```
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 - 0%
Шаг нашивки - 0%
Плотность нашивки - 0%
```

In [27]: #Дубликаты

In [48]: #Проверка данных на дубликатах
df.duplicated().sum()

Out[28]:

0

In [29]: #по заданию необходимо получить среднее, медианное значение для каждой колонки
#среднее значение

In [30]: #получим среднее и медианное значения данных в колонках

```
mean_and_sd = df.describe()
```

```
mean_and_sd.loc[['mean', '50%']]
```

```
#всех ли одинаковые друг к другу значения
```

Out[30]:

Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки	
mean	2.930366	1975.734888	739.923233	110.50769	22.244390	285.882151	482.73183	73.328571	2466.92843	218.423144	0.491691	6.899222	57.153929
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920

In [31]: # среднее значение

Out[31]:

df.mean()

Out[32]:

```
Соотношение матрица-наполнитель 2.930366
Плотность, кг/м3 1975.734888
модуль упругости, ГПа 739.923233
Количество отвердителя, м.% 110.50769
Содержание эпоксидных групп,%_2 22.244390
Температура вспышки, С_2 285.882151
Поверхностная плотность, г/м2 482.731833
Модуль упругости при растяжении, ГПа 73.328571
Прочность при растяжении, МПа 2466.92843
Потребление смолы, г/м2 218.423144
Угол нашивки 0.491691
Шаг нашивки 6.899222
Плотность нашивки 57.153929
dtype: float64
```

In [33]: # медианное значение

Out[33]:

df.median()

Out[34]:

```
Соотношение матрица-наполнитель 2.930378
Плотность, кг/м3 1977.621657
модуль упругости, ГПа 739.664328
Количество отвердителя, м.% 110.564840
Содержание эпоксидных групп,%_2 22.230744
Температура вспышки, С_2 285.896812
Поверхностная плотность, г/м2 451.864365
Модуль упругости при растяжении, ГПа 73.268805
Прочность при растяжении, МПа 2459.524526
Потребление смолы, г/м2 219.198882
Угол нашивки 0.000000
Шаг нашивки 6.916144
Плотность нашивки 57.341920
dtype: float64
```

In [35]: # Вычисляем коэффициенты ранговой корреляции Кендалла. Статистической зависимости не наблюдаем.

```
df.corr(method = 'kendall')
```

Out[35]:

Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки	Шаг нашивки	Плотность нашивки	
1.000000	-0.003135	0.001410	0.010180	-0.009480	-0.002060	-0.004157	0.011614	0.035145	-0.021395	0.022723	0.002788		
Плотность, кг/м3	-0.003135	1.000000	-0.008059	-0.021963	-0.007758	-0.019947	0.037302	0.021151	0.047426	-0.017079	-0.051525	-0.031220	0.052935
модуль упругости, ГПа	0.021247	-0.008059	1.000000	0.022382	0.021028	-0.000442	0.005458	0.022959	0.005161	-0.031695	-0.008305	0.049347	
Количество отвердителя, м.%	0.001410	-0.021963	0.022382	1.000000	0.000010	0.059034	0.033110	-0.043140	-0.046507	-0.003677	0.024690	0.006232	0.016607
Содержание эпоксидных групп,%_2	0.010180	-0.007758	0.002351	0.000010	1.000000	-0.002170	-0.006859	0.041994	-0.013441	0.009756	0.004668	-0.004539	-0.021968
Температура вспышки, С_2	-0.009480	-0.019947	0.021028	0.059034	1.000000	0.017196	0.0116481	-0.019106	0.035313	0.017880	0.029552	0.005268	
Поверхностная плотность, г/м2	-0.002060	0.037302	-0.008059	0.033110	-0.006859	1.000000	0.024051	-0.005099	-0.004446	0.045452	0.025514	-0.022320	
Модуль упругости при растяжении, ГПа	-0.004157	-0.021151	0.005458	-0.043140	0.041994	0.016481	1.000000	-0.006599	0.024814	0.024231	-0.010024	-0.002600	
Прочность при растяжении, МПа	0.011614	-0.047426	0.022382	-0.046507	-0.013441	-0.019106	-0.005099	1.000000	0.013580	0.020609	-0.048049	0.009821	
Потребление смолы, г/м2	0.035145	-0.017079	0.000000	-0.003677	0.009756	-0.004446	0.024051	-0.006599	1.000000	-0.002402	0.005962	0.010792	
Угол нашивки	-0.021395	-0.051525	-0.031695	0.024690	0.004668	0.017880	0.045452	0.022431	0.020609	-0.002402	1.000000	0.082142	
Шаг нашивки	0.022723	-0.031220	-0.008059	0.006232	-0.004539	0.029552	0.025514	-0.010024	0.0048049	0.005962	0.021178	1.000000	0.000658
Плотность нашивки	0.002788	0.052935	0.049347	0.016607	-0.021968	0.005268	-0.002320	0.009821	0.010792	0.082142	0.000658	1.000000	

In [36]: #Вычисляем коэффициенты корреляции Пирсона. Статистической зависимости не наблюдаем.

```
df.corr(method = 'pearson')
```

Out[36]:

Соотношение матрица-наполнитель	Плотность, кг/м ³	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп, %_2	Температура вспышки, С_2	Поверхностная плотность, г/м ²	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м ²	Угол нашивки	Шаг нашивки	Плотность нашивки	
0.000000	0.003841	0.031700	-0.006445	0.019766	-0.004776	-0.006272	-0.008411	0.024148	0.072531	-0.031073	0.036437	-0.004652	
Плотность, кг/м ³	0.003841	1.000000	-0.009647	-0.039911	-0.008278	-0.020695	0.044930	-0.017602	-0.069981	-0.015937	-0.068474	-0.061015	0.080304
модуль упругости, ГПа	0.031700	-0.009647	1.000000	0.024049	0.006804	0.011174	-0.005306	0.023267	0.041868	0.001840	-0.025417	-0.009875	0.056346
Количество отвердителя, м.%	-0.006445	-0.039911	0.024049	1.000000	-0.006684	0.095193	0.055198	-0.065592	-0.075375	0.007446	0.038570	0.014887	0.017248
Содержание эпоксидных групп, %_2	0.019766	-0.008278	-0.006684	-0.006684	1.000000	-0.009769	-0.012940	0.056828	-0.023899	0.015165	0.008052	0.003022	-0.059073
Температура вспышки, С_2	-0.004776	-0.020695	0.031714	0.095193	-0.009769	1.000000	0.020121	0.028414	0.031763	0.059954	0.020695	0.025795	0.011391
Поверхностная плотность, г/м ²	-0.006272	0.044930	-0.009306	0.055198	-0.012940	0.020121	1.000000	0.036702	-0.03210	0.015692	0.052299	0.038332	-0.049923
Модуль упругости при растяжении, ГПа	-0.008411	-0.017602	0.023267	-0.065329	0.056828	0.028414	0.036702	1.000000	-0.009009	0.050938	0.023003	-0.029468	0.006476
Прочность при растяжении, МПа	0.024148	-0.069981	0.041868	-0.075375	-0.023899	-0.031763	-0.003210	-0.009009	1.000000	0.028602	0.023398	-0.059547	0.019604
Потребление смолы, г/м ²	0.072531	-0.015937	0.001840	0.007446	0.015165	0.059954	0.015692	0.050938	0.028602	1.000000	-0.015334	0.013394	0.012239
Угол нашивки	-0.031073	-0.068474	-0.025417	0.038570	0.008052	0.020695	0.052299	0.023003	0.023398	-0.015334	1.000000	0.023616	0.107947
Шаг нашивки	0.036437	-0.061015	-0.009875	0.014887	0.003022	0.025795	0.038332	-0.029468	-0.059547	0.013394	0.023616	1.000000	0.003487
Плотность нашивки	-0.004652	0.080304	0.056346	0.017248	-0.039073	0.011391	-0.049923	0.006476	0.019604	0.012239	0.107947	0.003487	1.000000

In [37]: # Создадим переменную для назначения всех столбцов. Это нам пригодится при построении моделей. И передадим к визуализации данных

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

column_names = df.columns

Визуализируем сырье данные и проведем анализ

- Построим гистограммы распределения каждой из переменных и боксплоты (несколько разных способов визуализации).
- диаграммы «ящиков с усами» (несколько вариантов).
- попарные графики рассеяния точек (несколько вариантов)
- графики квантиль-квантиль без нормализации и исключения шумов

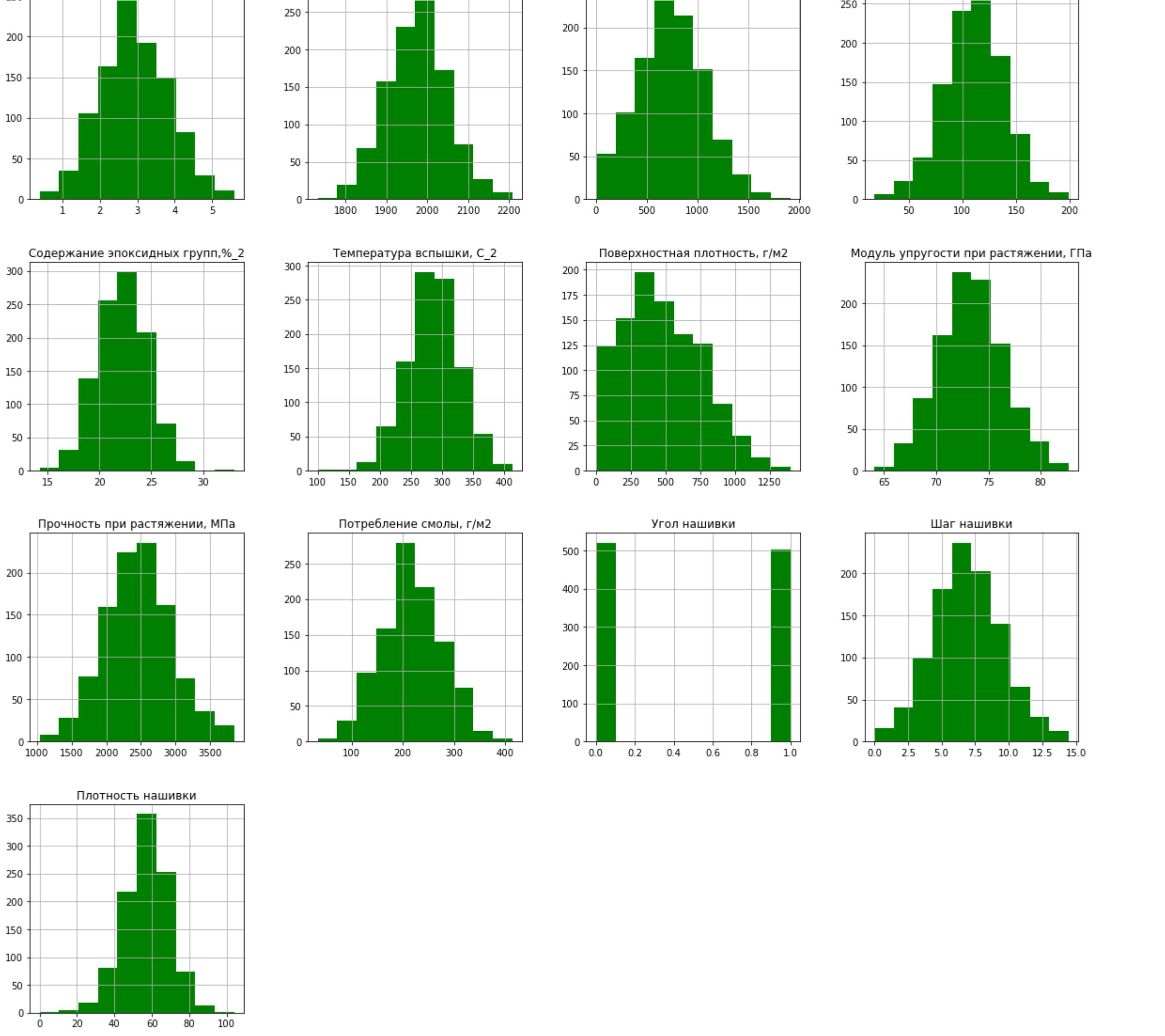
Т.к. беглый взгляд на общий файл и дополнительный анализ в excel не дал каких-то явных и бросающихся в глаза закономерностей, то используем разные варианты визуализации в надежде, что получится увидеть какую-то корреляцию. И разные варианты одного и того же типа визуализации используются для отображения результата, потому что какие-то графики отображаются в jupyter, но не работают в colab, какие-то не работают в Github

Показатели описательной статистики и визуализация гистограмм и/или дза-грамм размаха «ящиков с усами» позволяют получить наглядное представление о характеристах распределений переменных. Такое частотное распределение показывает, какие именно конкретные значения или диапазоны значений исследуемой переменной встречаются наиболее часто, насколько различаются эти значения, расположено ли большинство наблюдений около среднего значения, является распределение симметричным или асимметричным, многомодальным (т.е. имеет две или более вершины) или одномодальным и т.д. По форме распределения можно судить о природе исследуемой переменной (например, бимодальное распределение позволяет предположить, что выборка не является однородной и содержит наблюдения, принадлежащие двум различным множествам, которые в свою очередь нормально распределены).

In [38]: # Построим гистограммы распределения каждой из переменных без нормализации и исключения шумов

df.hist(figsize = (20,20), color = "g")

plt.show()



При проведении анализа выявлены параметры близкие к нормальному: Соотношение матрица-наполнитель; Плотность, кг/м³; Модуль упругости, ГПа; Количество отвердителя, м.%; Содержание эпоксидных групп, %_2; Температура вспышки, С_2; Поверхностная плотность, г/м²; Модуль упругости при растяжении, ГПа; Прочность при растяжении, МПа; Потребление смолы, г/м²; Угол нашивки; Шаг нашивки. Преимущественно данные стремятся к нормальному распределению. Угол нашивки, как и отражено в датасете, имеет только два значения 90 градусов и 0 градусов, что отражает общий подход к проведению нашивки материалов, а также может быть использовано при обработке данных. Учитывая отсутствие иных показателей для угла нашивки, предлагаем в прогнозе использовать категориальный, а не непрерывный подход при анализе данного параметра.

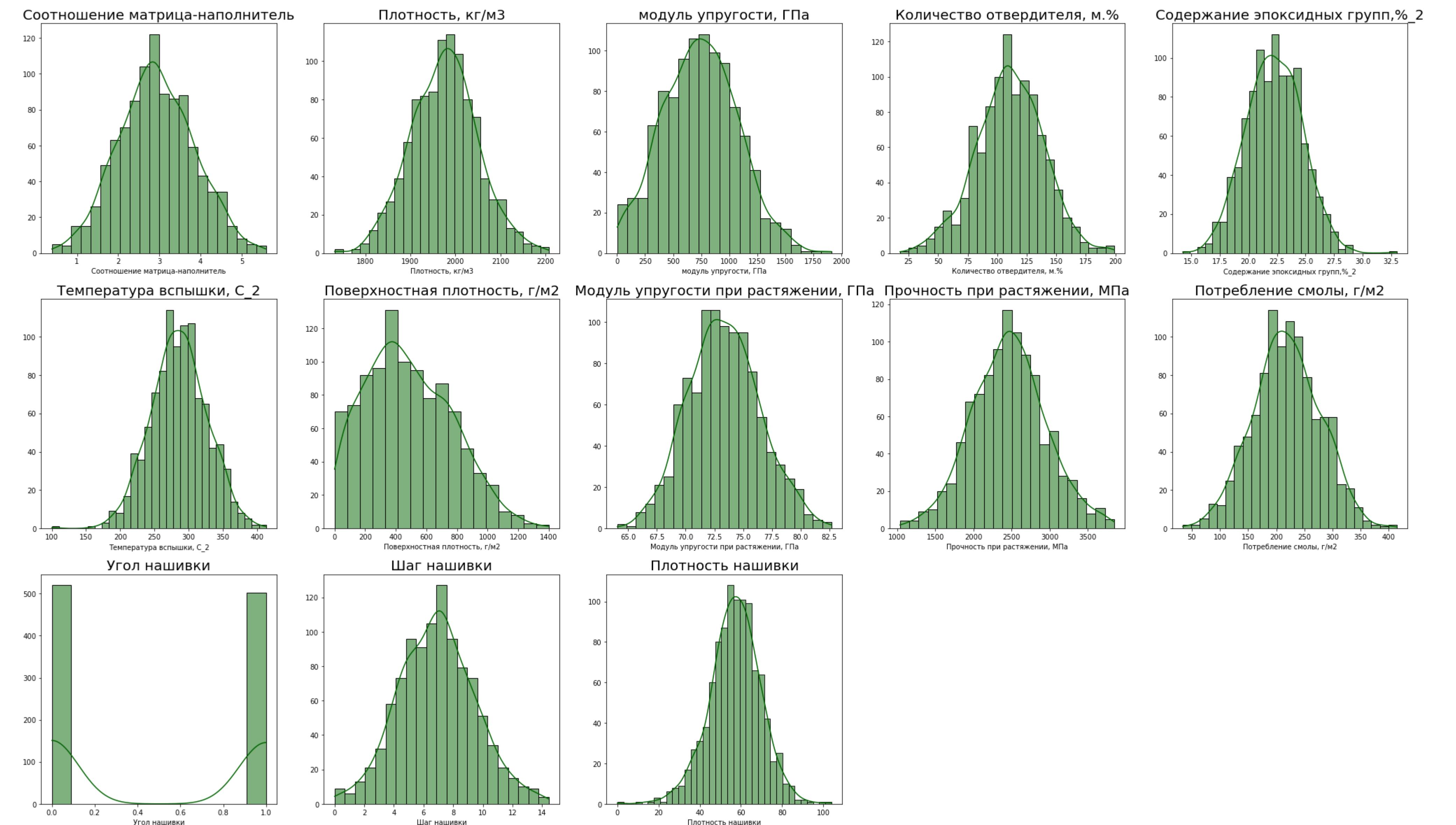
In [39]:

```
# Гистограммы распределения (второй вариант)
a = 5 # количество строк
b = 5 # количество столбцов
c = 1 # количество ячеек plot counter
plot_counter = 1
plt.figure(figsize = (35,35))
plt.suptitle('Гистограммы переменных', fontsize = 38)
for col in df.columns:
    for row in df.columns:
        if row == col:
            plt.subplot(a, b, c)
            plt.title(f'{col}', loc='center')
            sns.histplot(data = df[col], kde=True, color = "darkgreen")
            plt.xlabel('')
            plt.title(col, size = 20)
            plt.xticks(size = 10)
            plt.yticks(size = 10)
            c += 1
        else:
            plt.subplot(a, b, plot_counter)
            plt.title(f'{row}', loc='center')
            sns.histplot(data = df[row], kde=True, color = "darkgreen")
            plt.xlabel('')
            plt.title(row, size = 20)
            plt.xticks(size = 10)
            plt.yticks(size = 10)
            plot_counter += 1
            c += 1
```

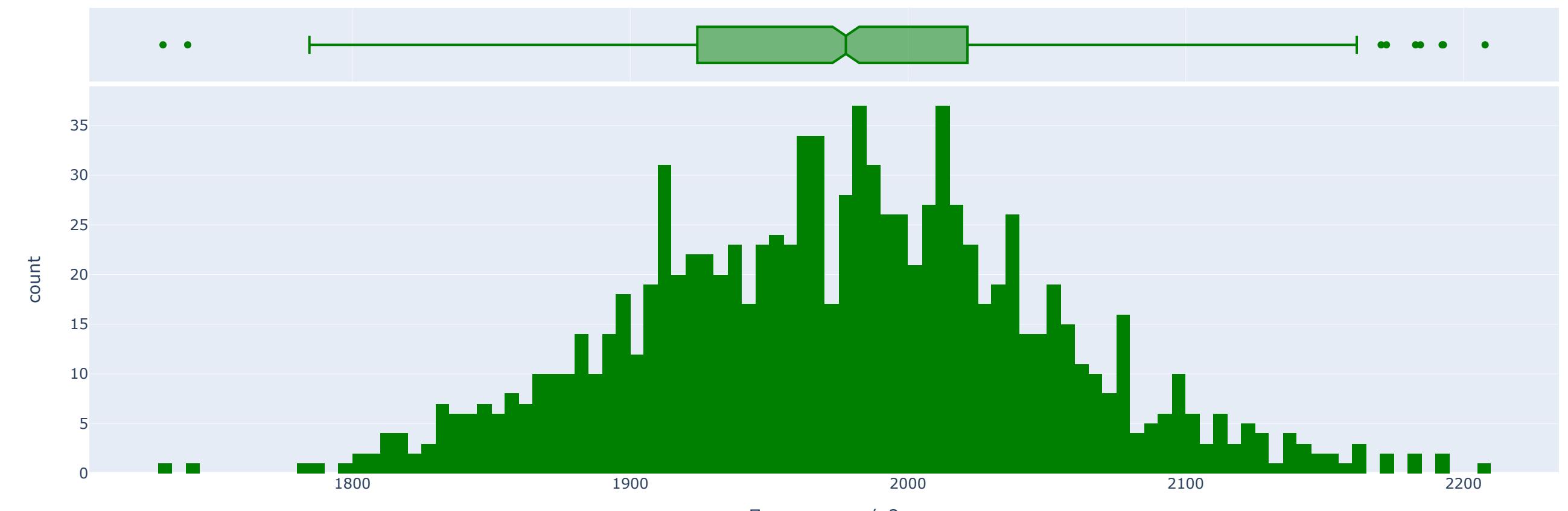
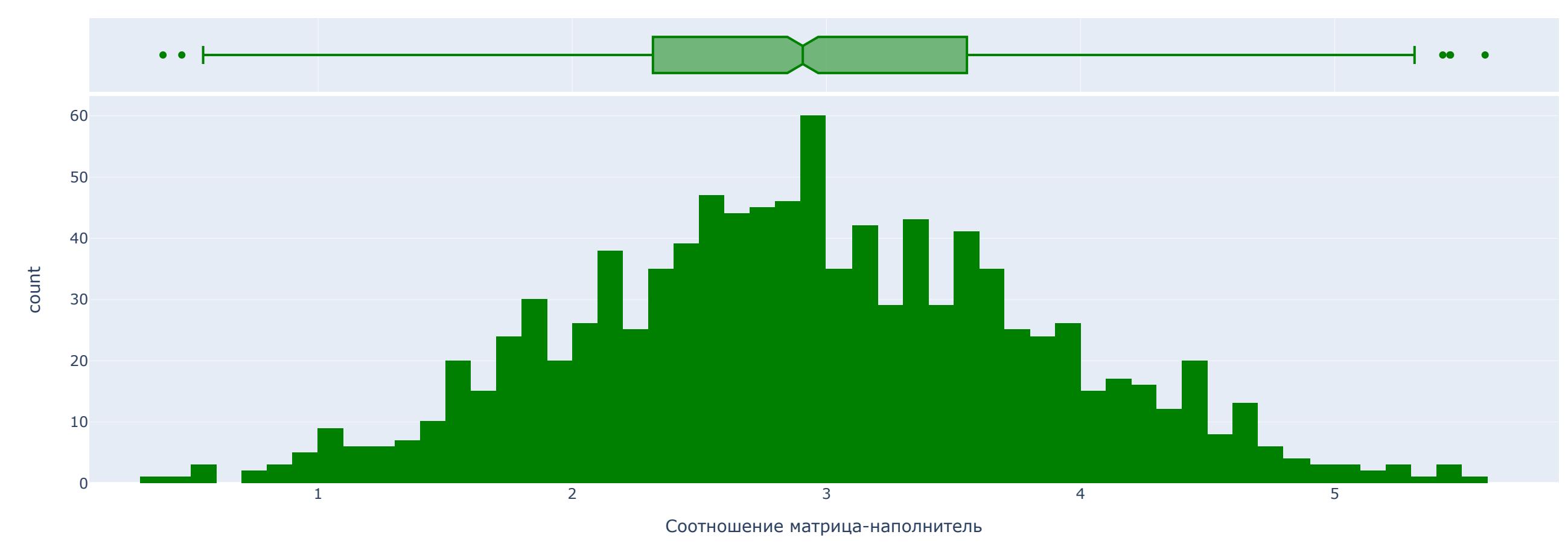
#Гистограммы показывают присутствие выбросов в столбцах: плотность, содержание эпоксидных групп, температура вспышки, плотность нашивки.

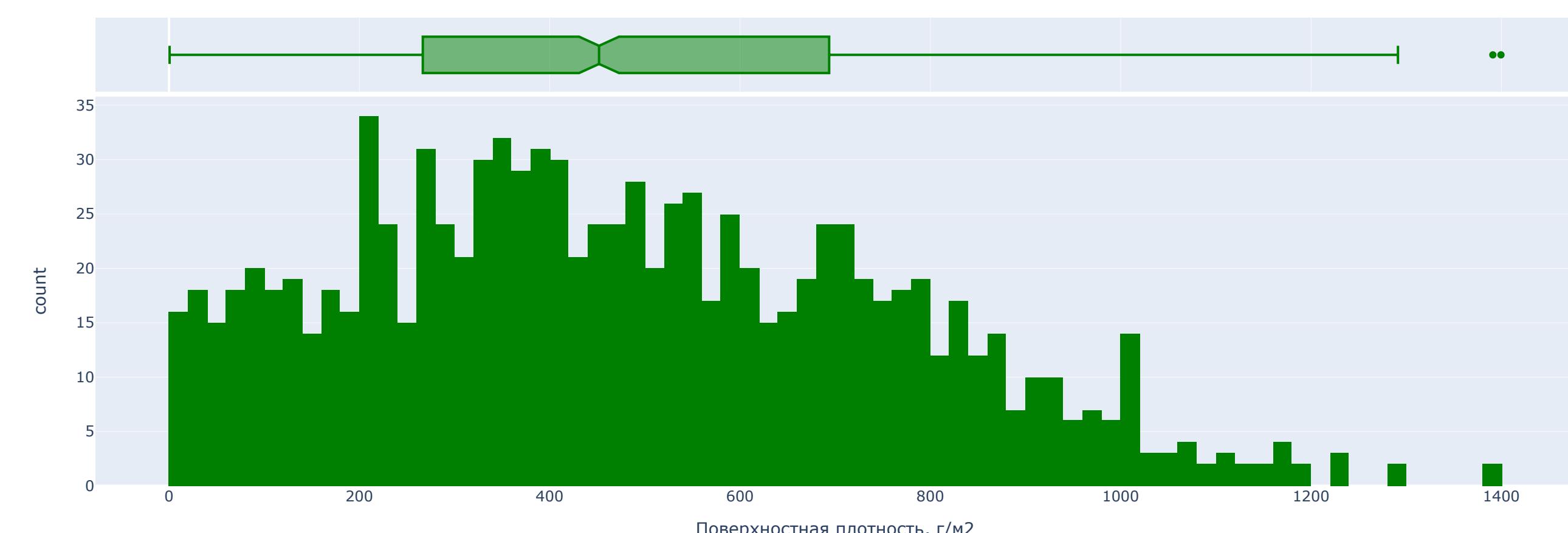
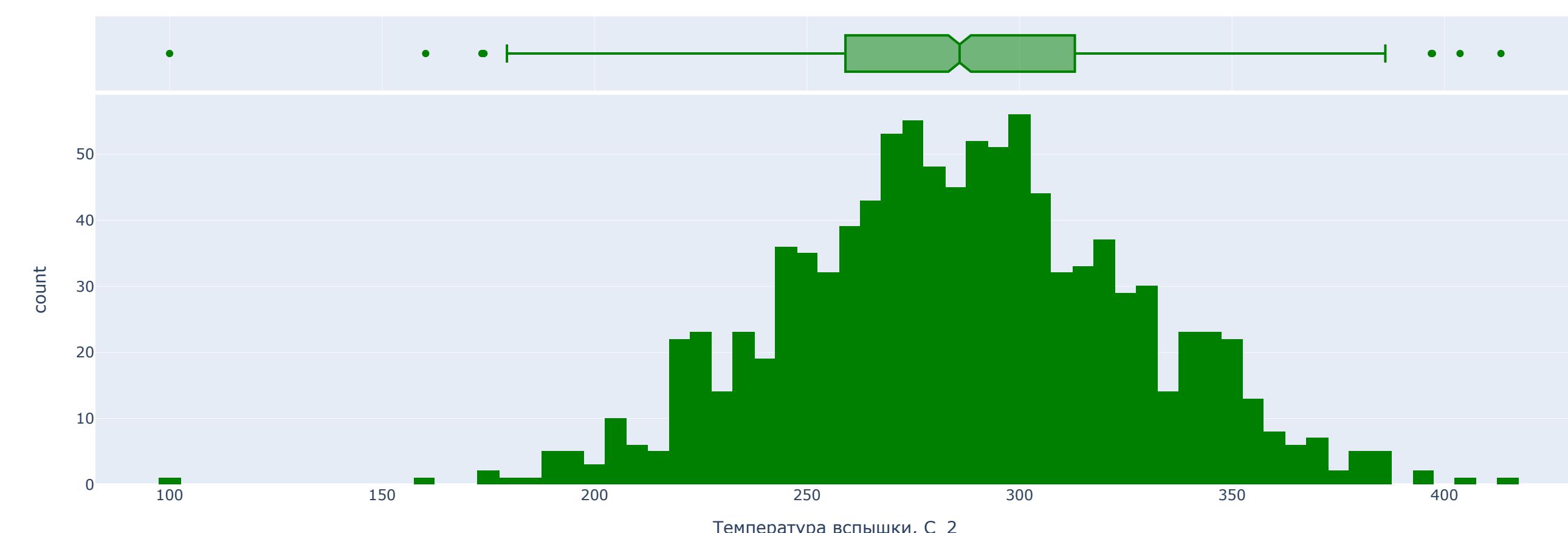
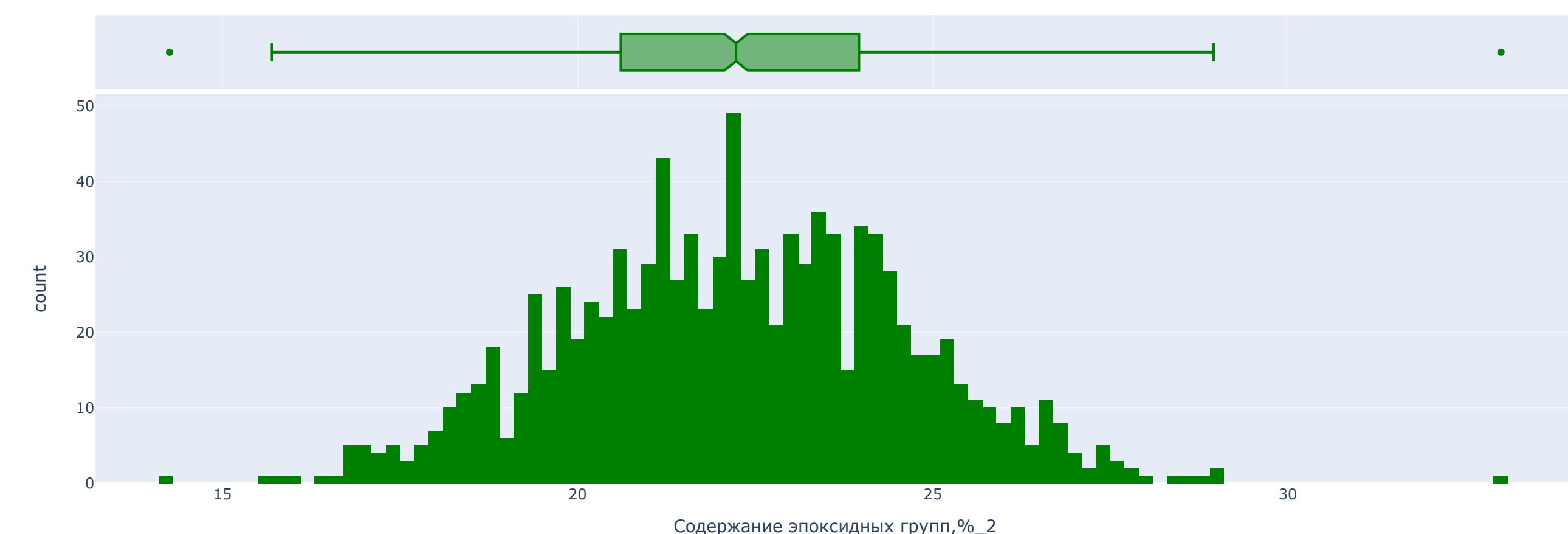
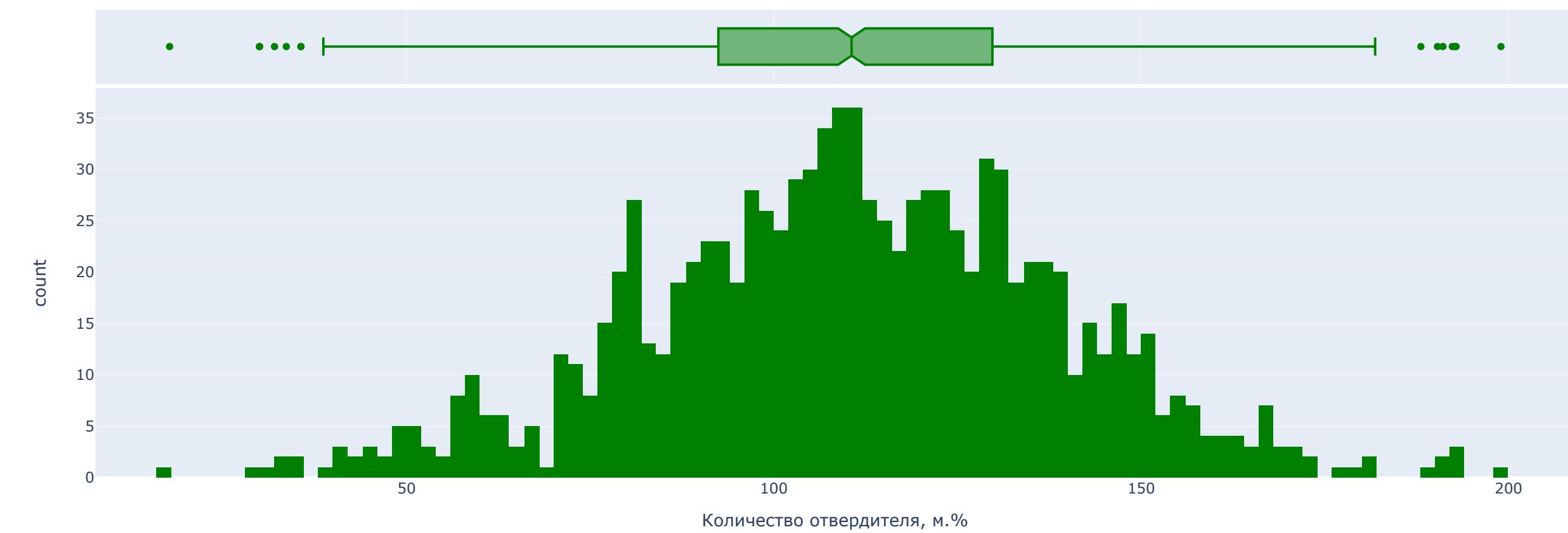
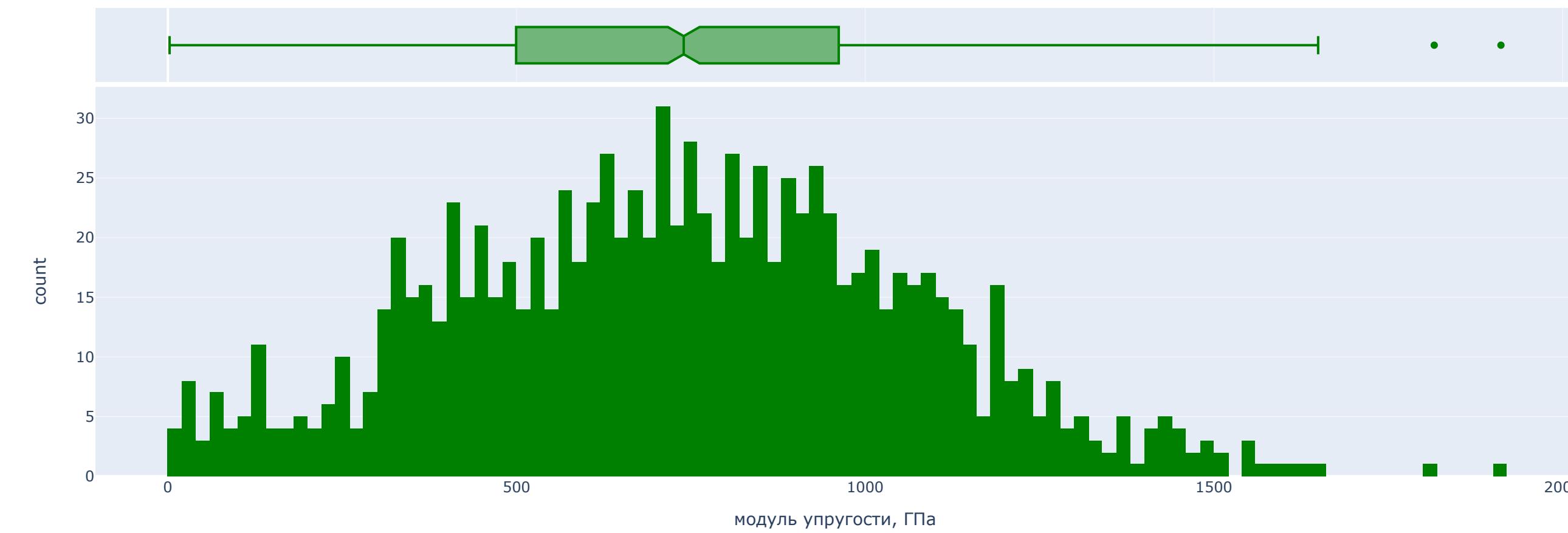
И данные стремятся к нормальному распределению практически безде, кроме угла нашивки, имеющим только 2 значения, с которым мы уже работали ранее.

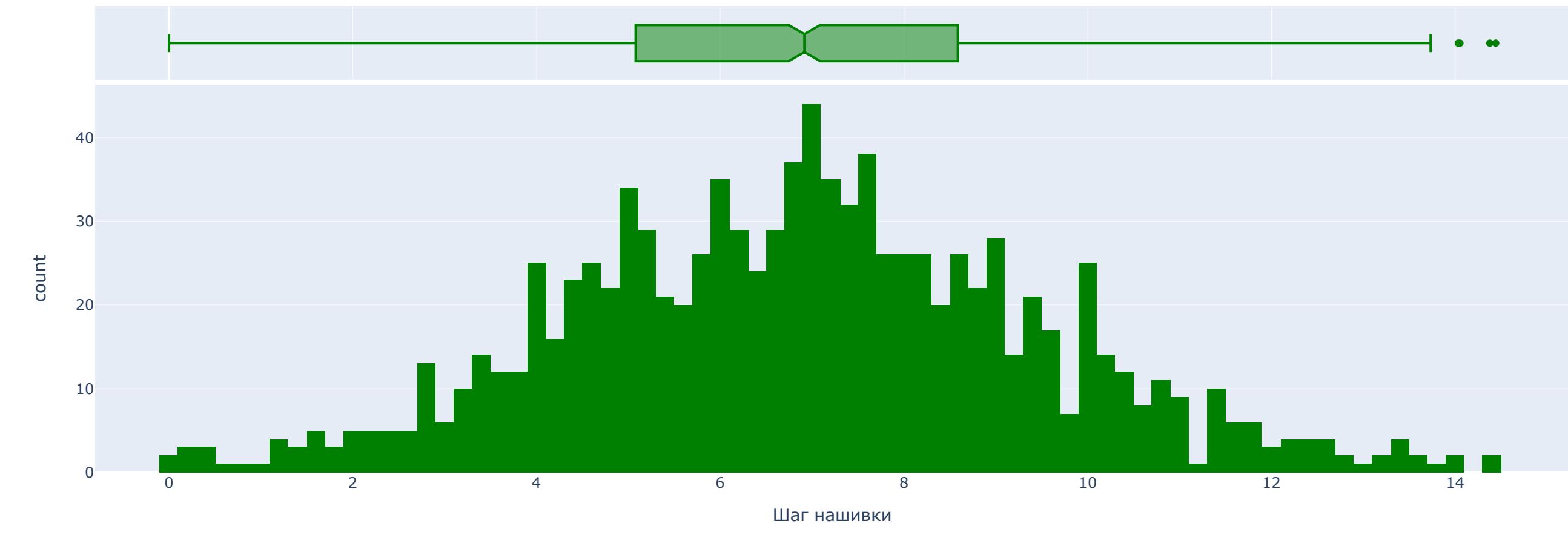
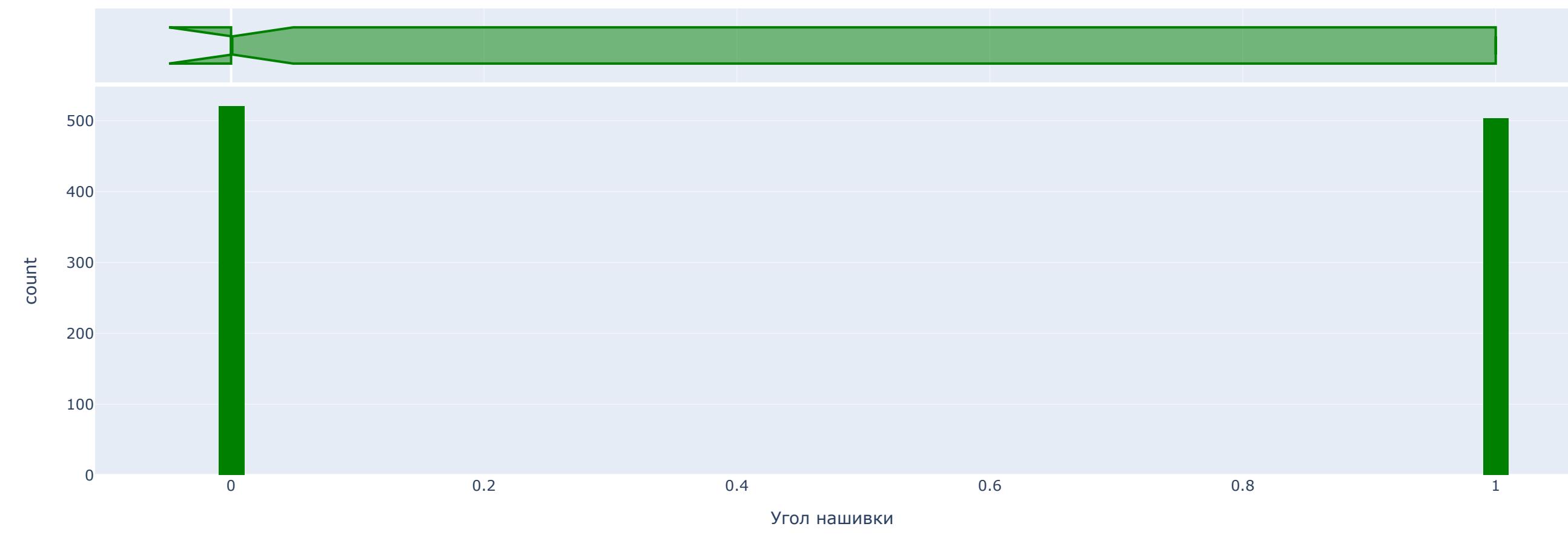
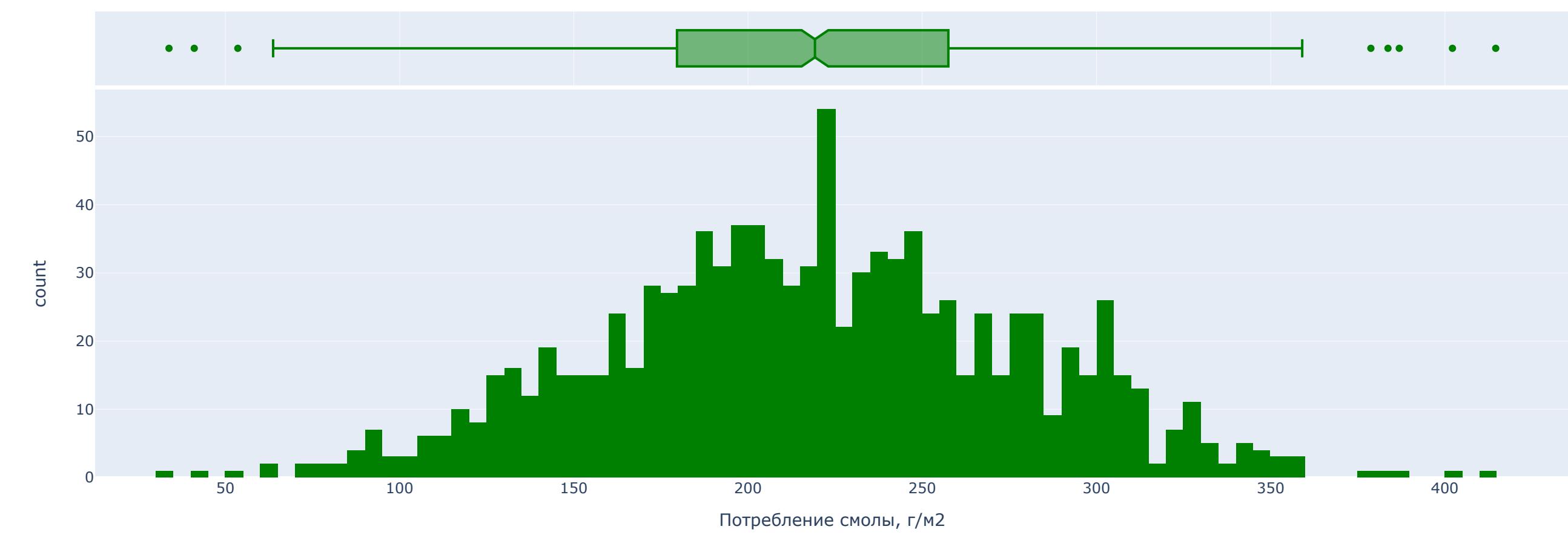
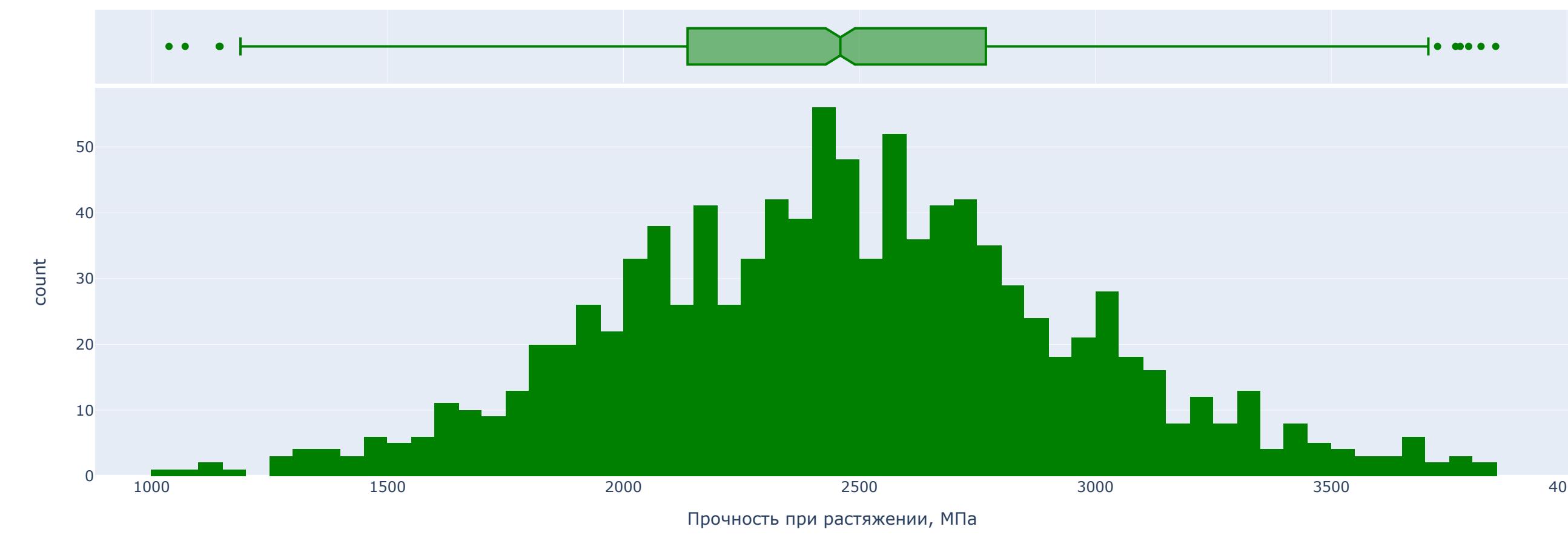
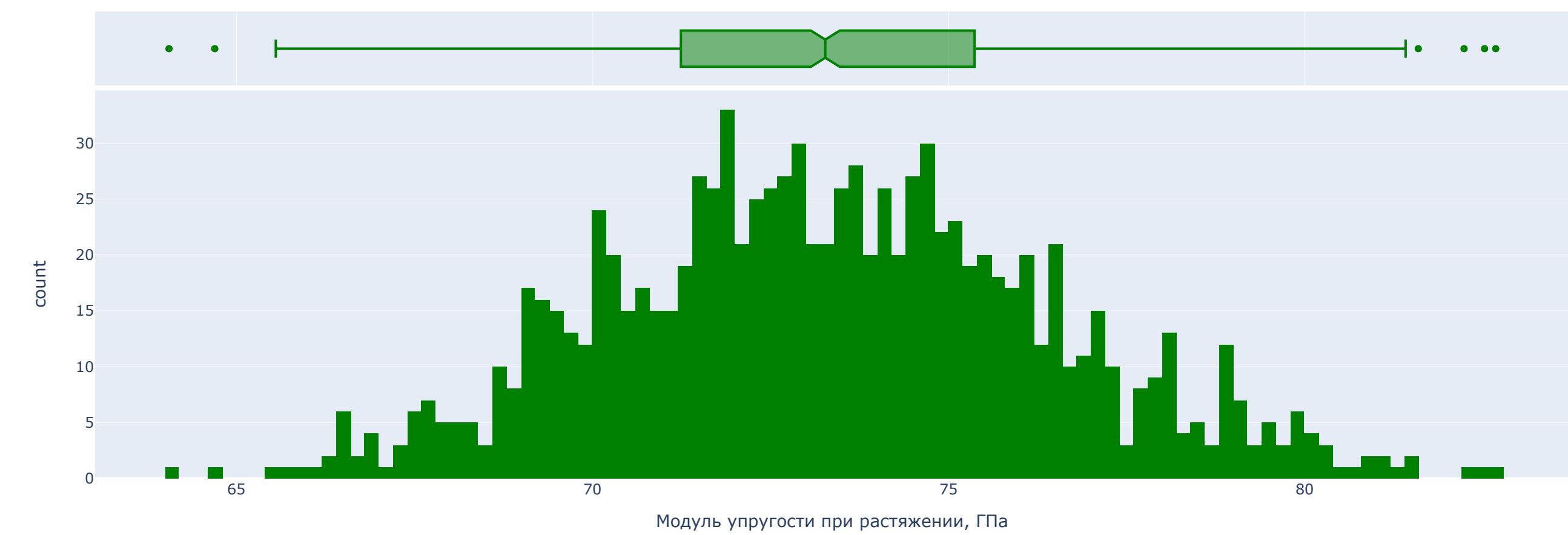
Гистограммы переменных

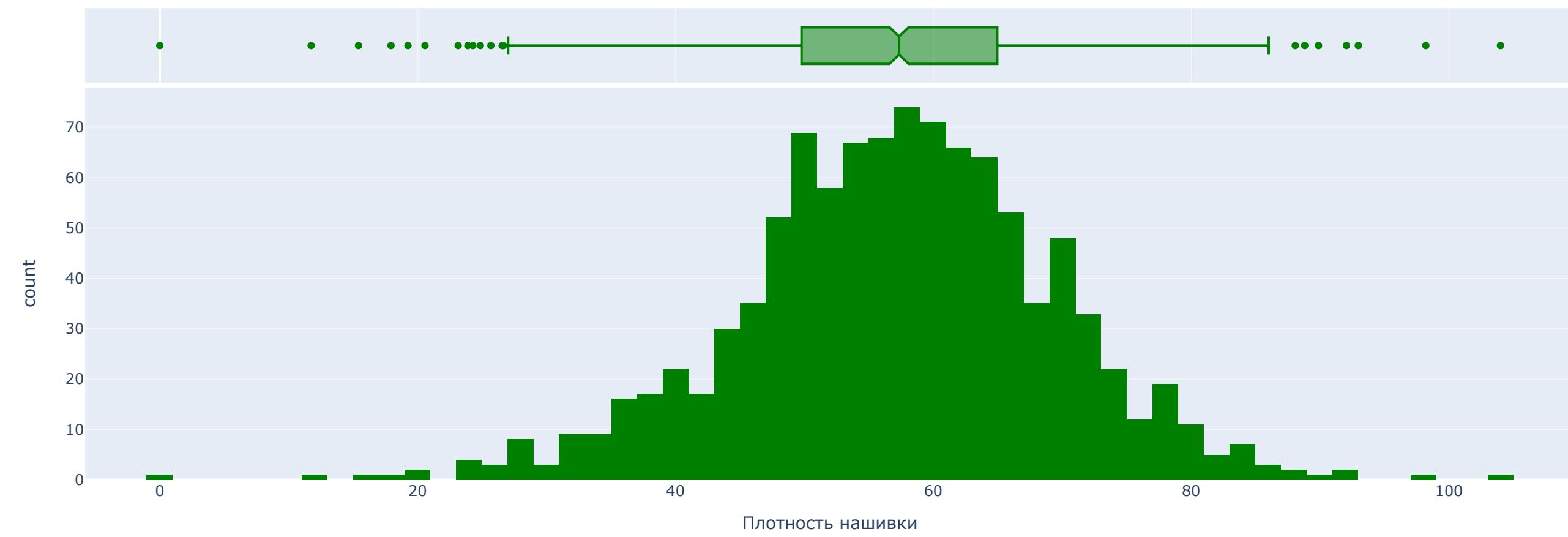


```
In [48]: # гистограмма распределения и боксплоты (пример для одного)
for column in df.columns:
    fig = px.histogram(df, x=column, color_discrete_sequence = ['green'], nbins = 100, marginal = "box")
    fig.show()
```

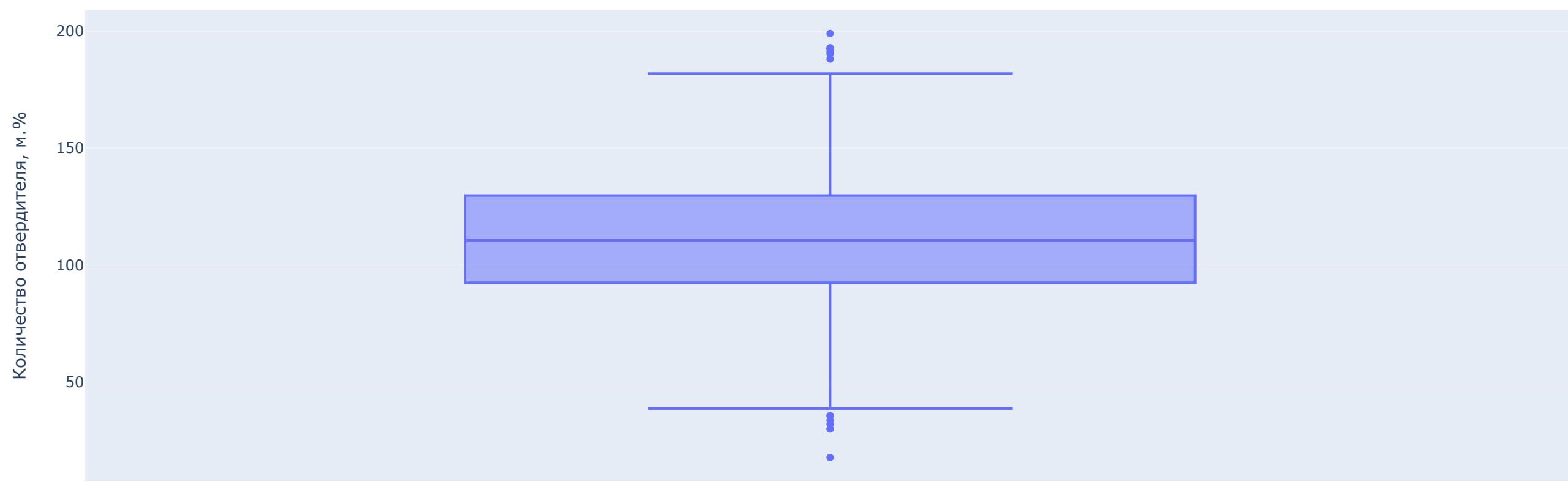
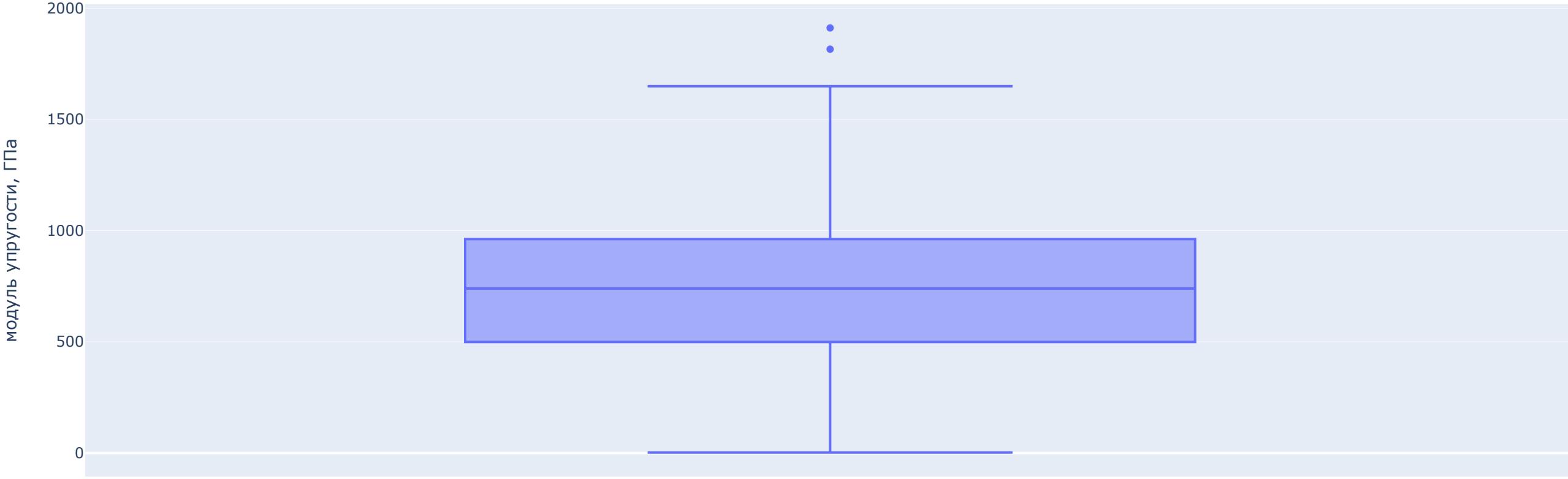
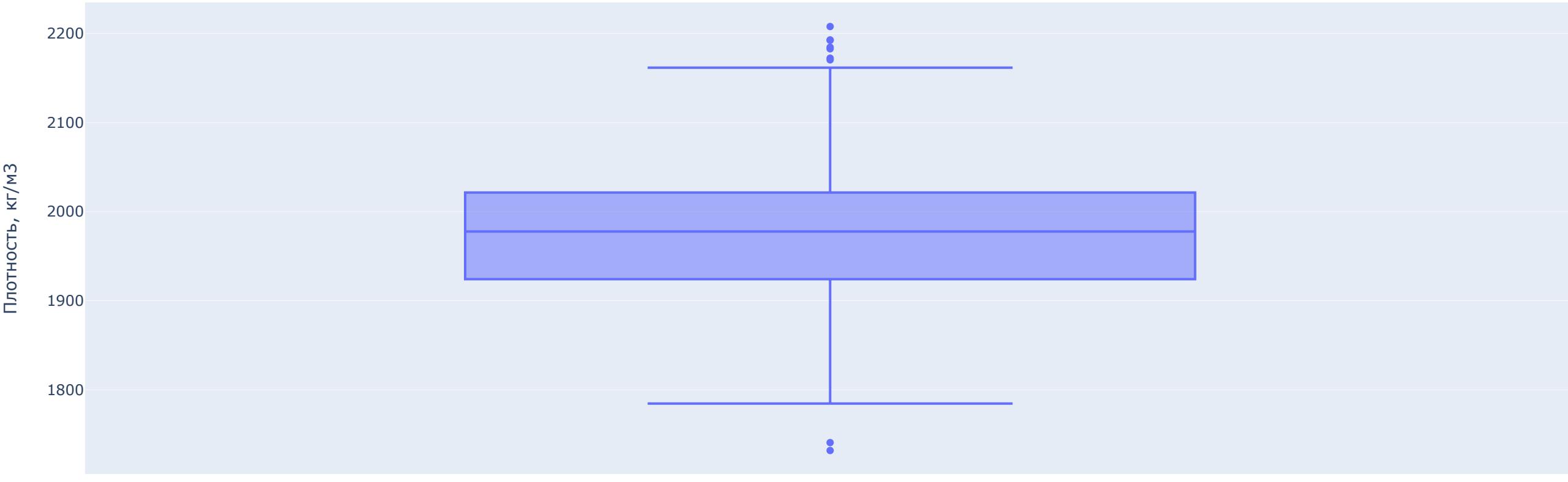
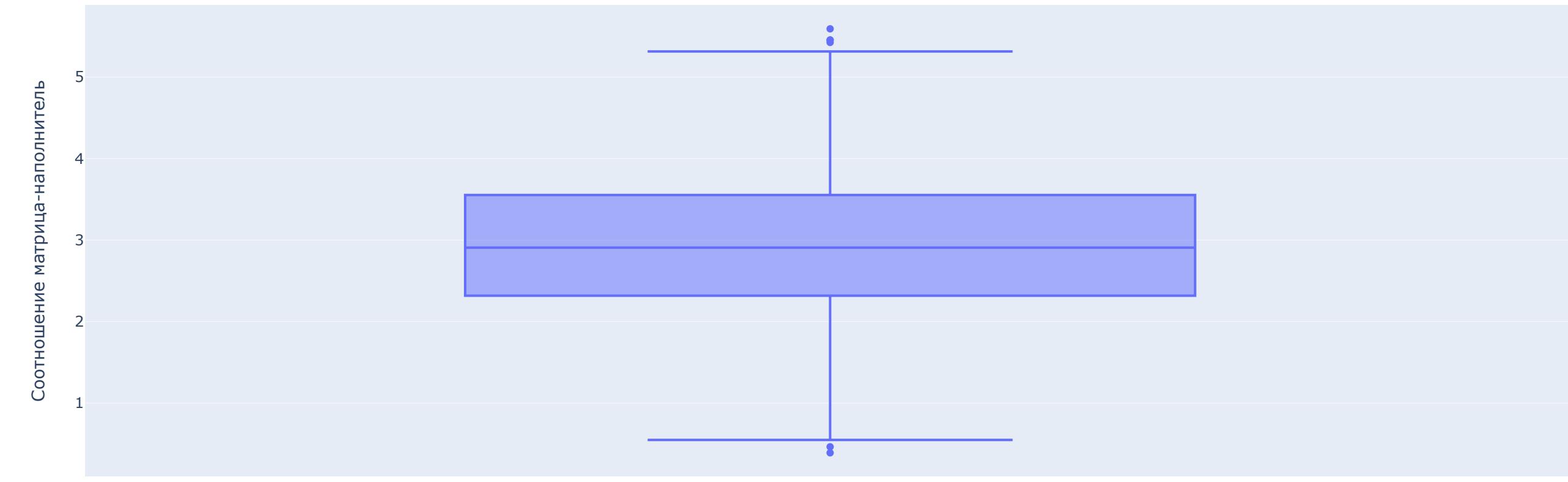


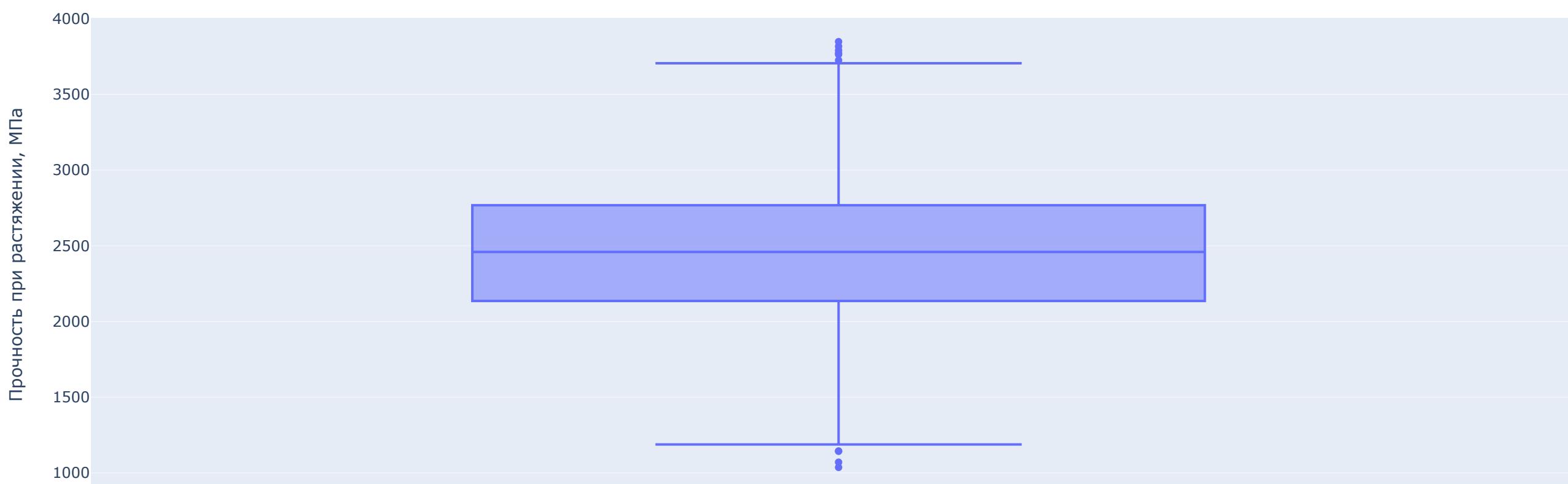
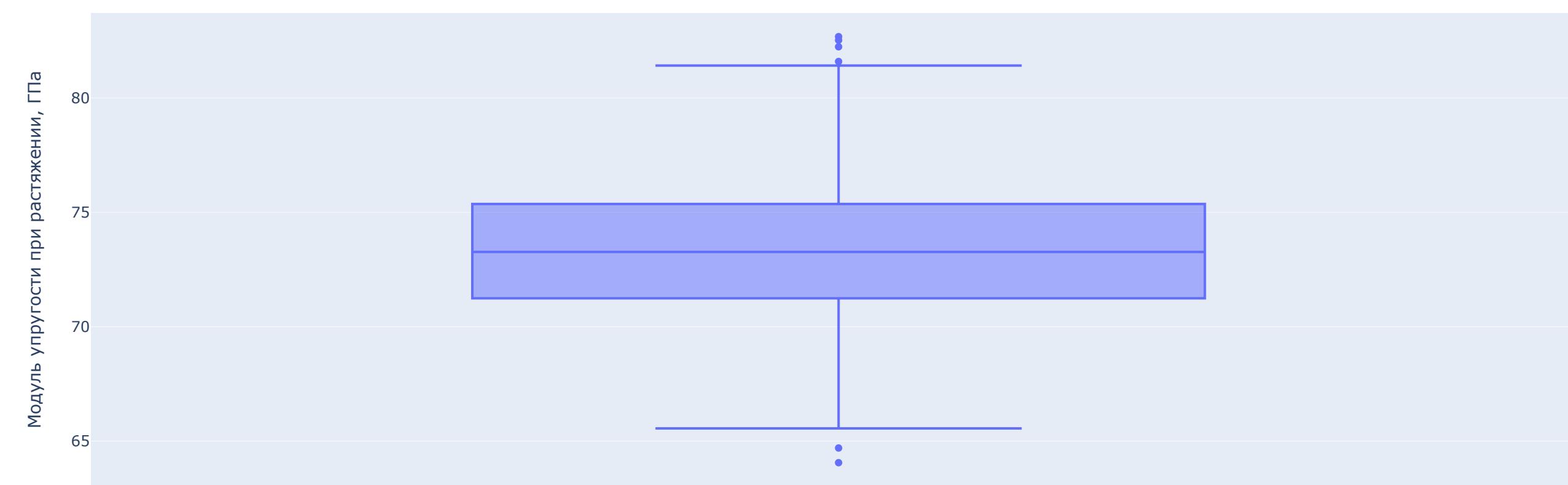
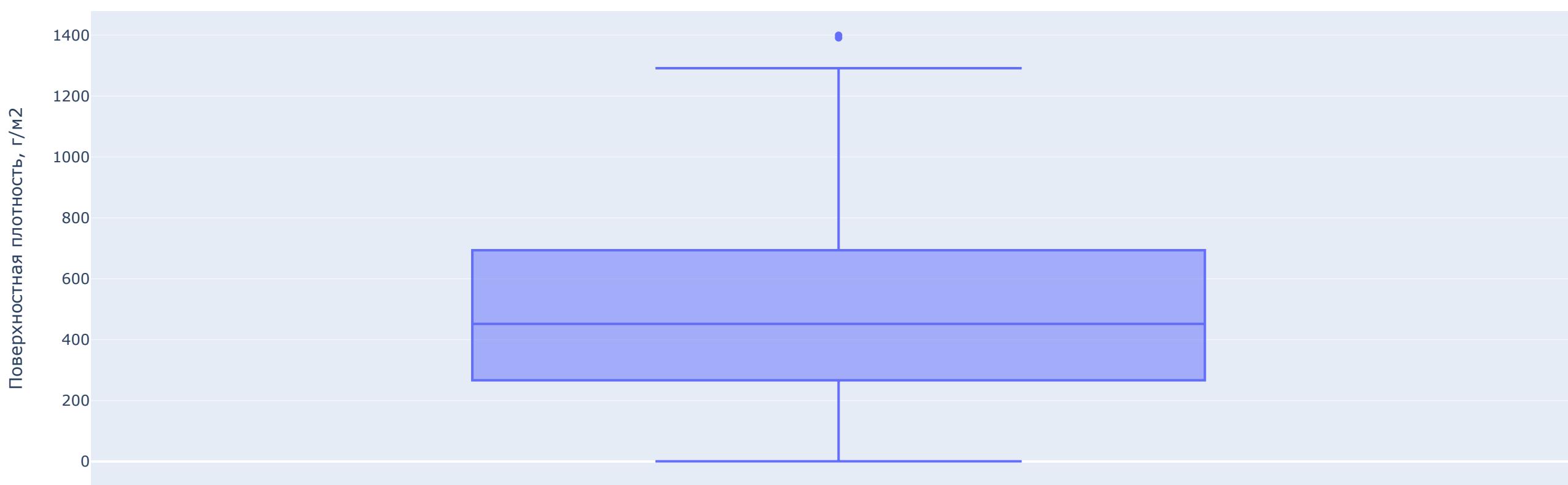
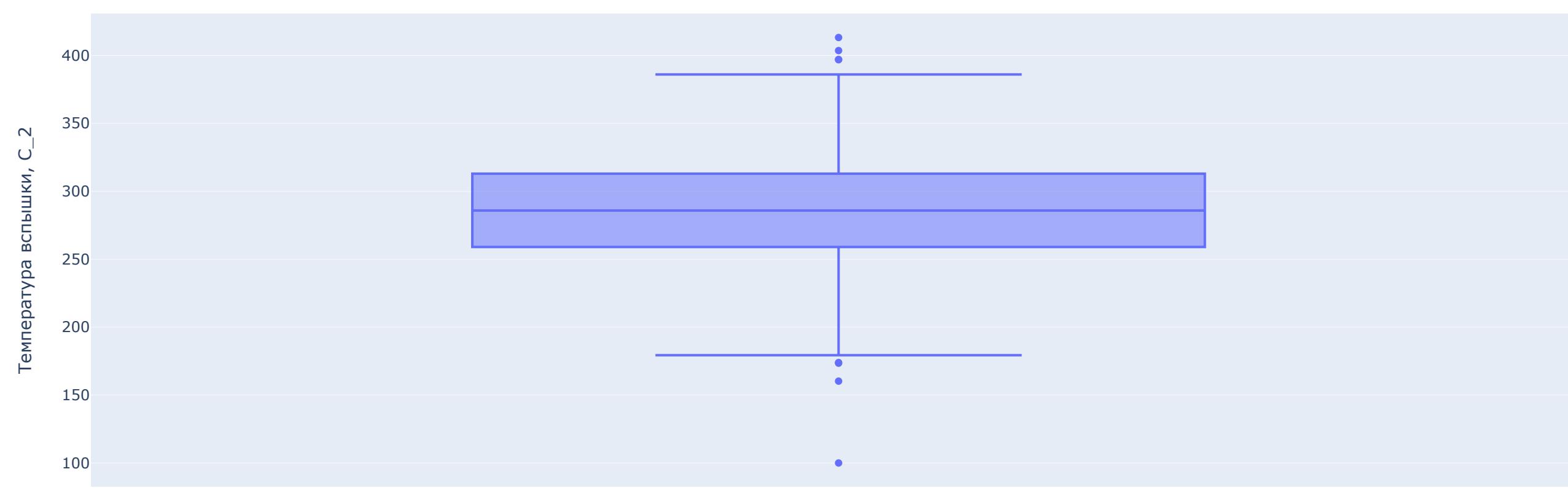
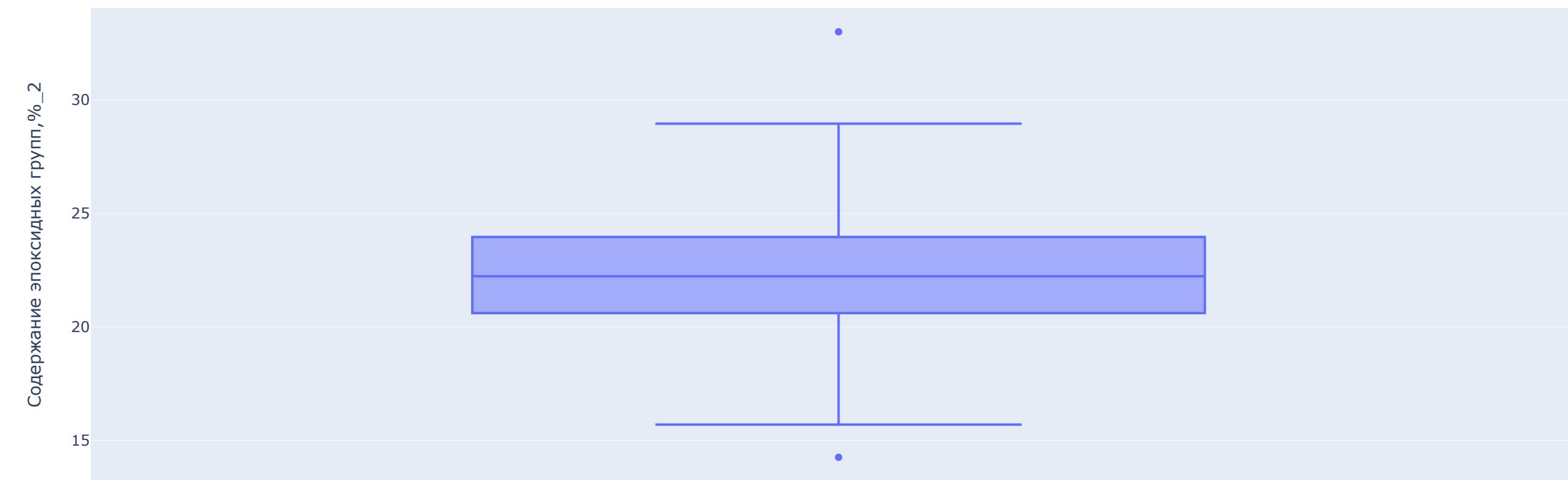


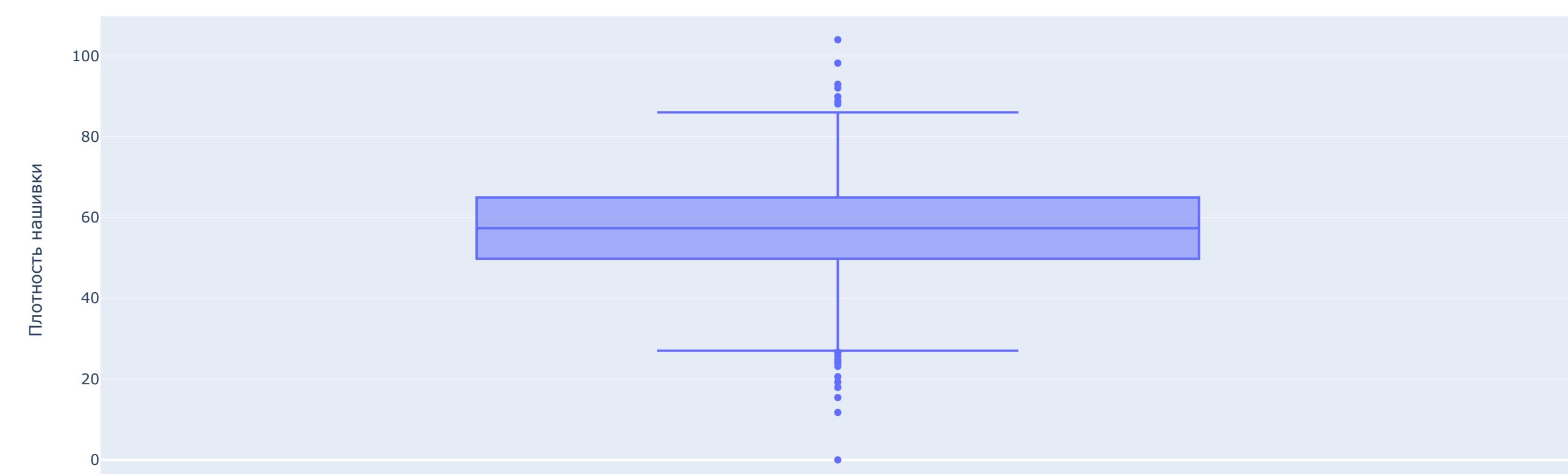
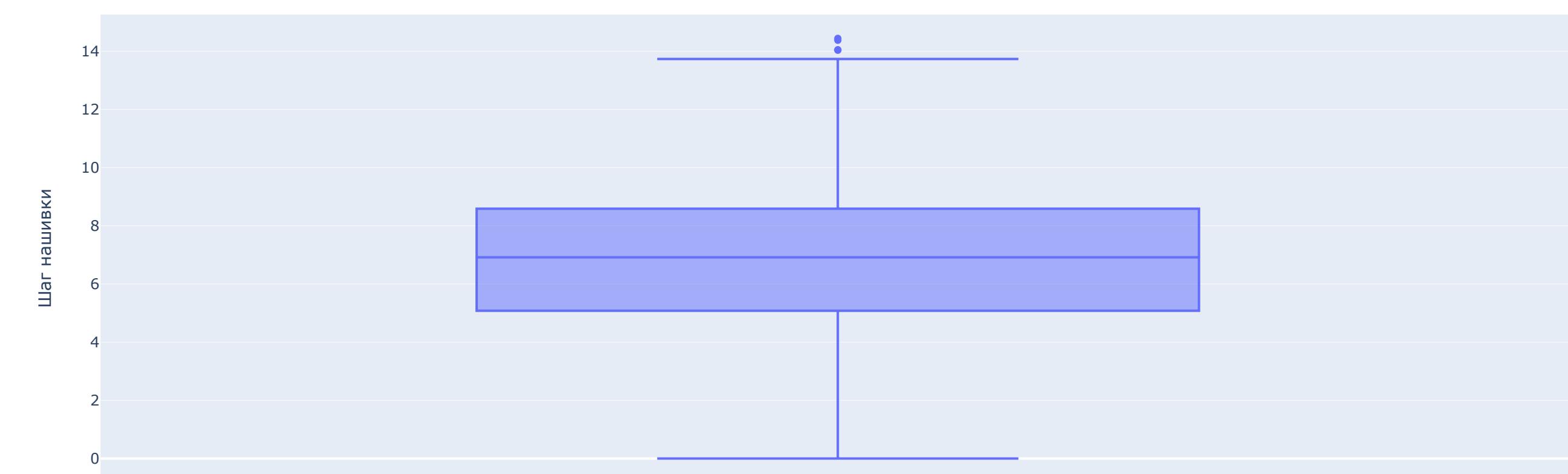
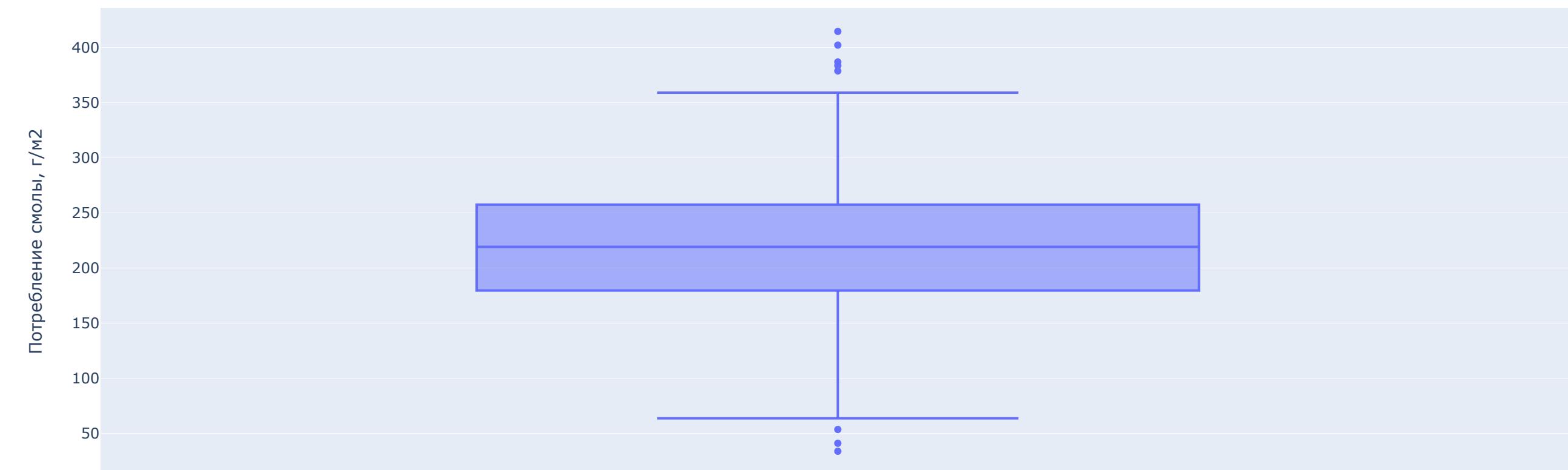




```
In [41]: for column in df.columns:  
    fig = px.box(df, y = column)  
    fig.show()
```

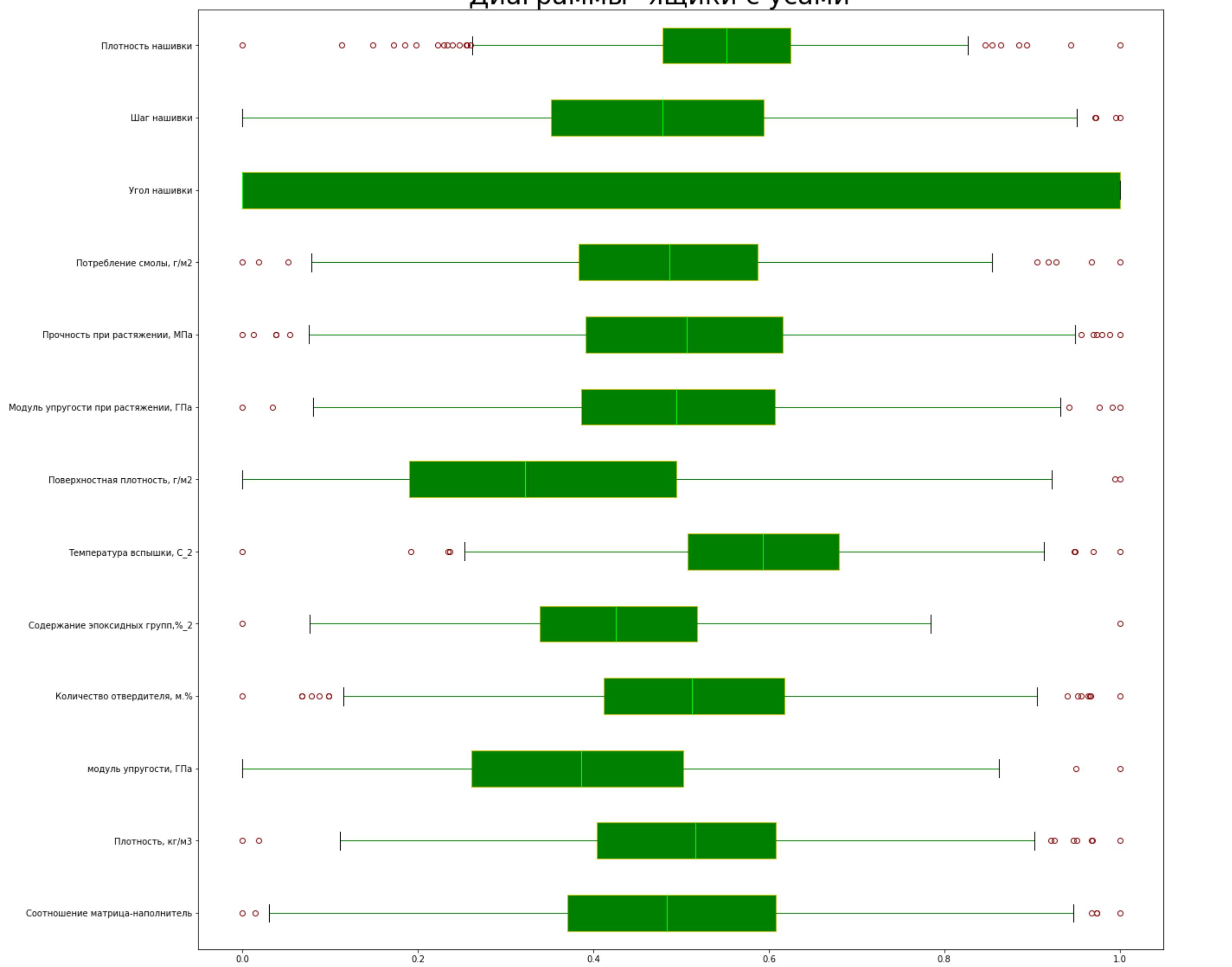






```
In [42]: # "Линии с усами"(боксплоты) (первый вариант)
scaler = StandardScaler()
scaler.fit(df)
plt.figure(figsize = (20, 20))
plt.suptitle('Диаграммы "линии с усами"', y = 0.9,
            fontsize = 30)
plt.boxplot(pd.DataFrame(scaler.transform(df)), labels = df.columns,patch_artist = True, meanline = True, vert = False, boxprops = dict(facecolor = 'g', color = 'y'),medianprops = dict(color = 'lime'), whiskerprops = dict(color = "g"), capprops = dict(color = "black"), flierprops = dict(color = "y", markeredgecolor = "maroon"))
plt.show()
```

Диаграммы "ящики с усами"



Многие алгоритмы машинного обучения чувствительны к разбросу и распределению значений признаков обрабатываемых объектов. Соответственно, выбросы во входных данных могут исказить и ввести в заблуждение процесс обучения алгоритмов машинного обучения, что приводит к увеличению времени обучения, снижению точности моделей и, в конечном итоге, к снижению результатов.

```
In [ ]: # Ящики с усами (первый вариант)
a = 5 # количество строк
b = 5 # количество столбцов
c = 1 # инициализация plt counter
plt.figure(figsize = (35,35))
plt.suptitle("Диаграммы \"ящики с усами\"", y = 0.9 , fontsize = 30)
for col in df.columns:
    plt.subplot(a, b, c)
    plt.figure(figsize=(7,5))
    sns.boxplot(data = df, x = df[col], fliersize = 15, linewidth = 5, boxprops = dict(facecolor = 'y'), color = 'g', medianprops = dict(color='lime'), whiskerprops = dict(color="g"), capprops = dict(color = "yellow"), flierprops = dict(color="y", markeredgecolor = "lime"))
    plt.xlabel(col, size = 20)
    plt.title(col, size = 20)
    plt.show()
    c += 1
```

"ящики с усами" показывают наличие выбросов во всех столбцах, кроме углов нашивки, значит, с ними будем работать

```
In [ ]: # Генерическое распределение и диаграмма "ящик с усами" вместе с данными по каждому столбцу
for column_name in column_names:
    print(column_name)

#Генерическое распределение
gls = df[column_name]
sns.set_style("whitegrid")
sns.kdeplot(data = gls, shade = True, palette = "colorblind", color = "g")
plt.show()

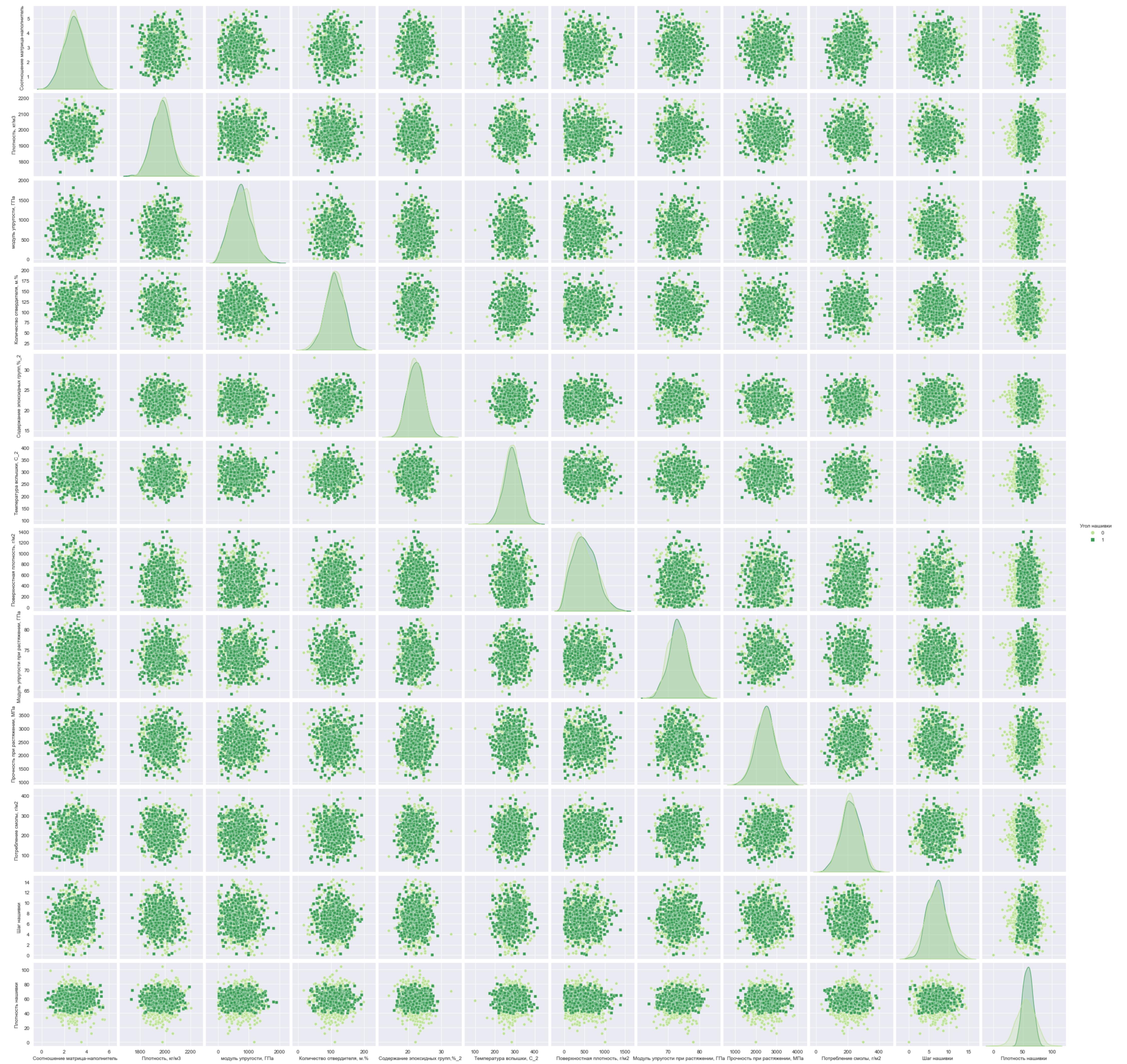
#Диаграмма "Ящик с усами"
sns.boxplot(x=gls, color = "g");
plt.show()

#Значения (или макс)
print("Минимальное значение: ", end = " ")
print(np.min(gls))
print("Первый квартиль: ", end=" ")
print(np.percentile(gls, 25))
print("Второй квартиль: ", end=" ")
print(np.percentile(gls, 50))
print("Среднее значение: ", end = " ")
print(np.mean(gls))

print("Медианное значение: ", end = " ")
print(np.median(gls))
print("\n\n")
```

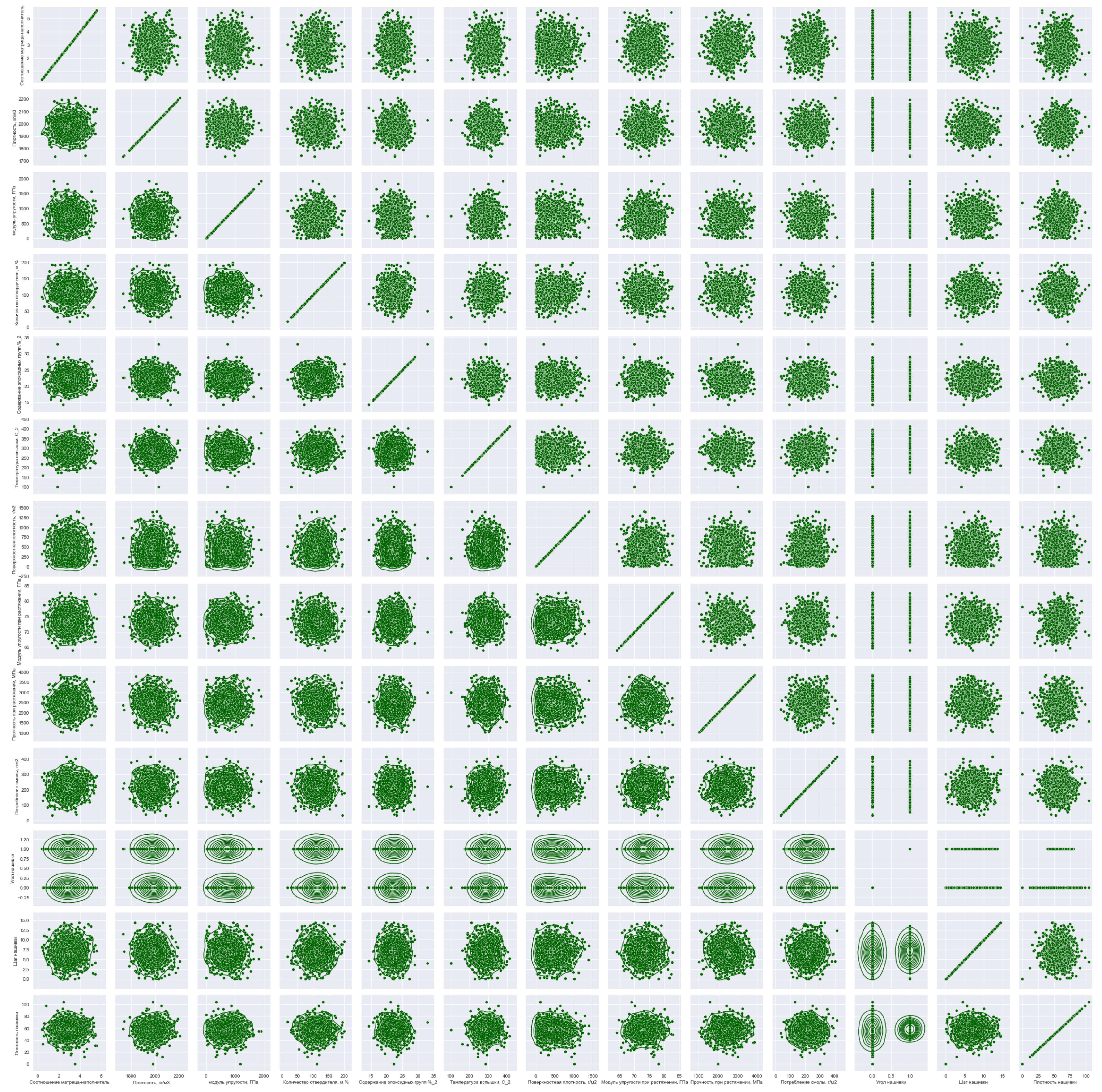
```
#Парные графики распределения точек (матрица диаграмм рассеяния) (первый вариант)
sns.set_style('darkgrid')
sns.pairplot(df, hue = 'Угол нашивки', markers = ["o", "s"], diag_kind = 'auto', palette="YIGn")
# Парные графики распределения точек так же не показывают какой-либо зависимости между данными. Зависимость между показателями не линейная, взаимосвязь отсутствует, необходимо использовать несколько показателей.
# из графиков можно наблюдать выбросы, потому что некоторые точки расположены далеко от общего облака
# Описанные линейной корреляции наборомика подтверждаются при построении регрессии?
```

<seaborn.axisgrid.PairGrid at 0x1c3db5688e0>

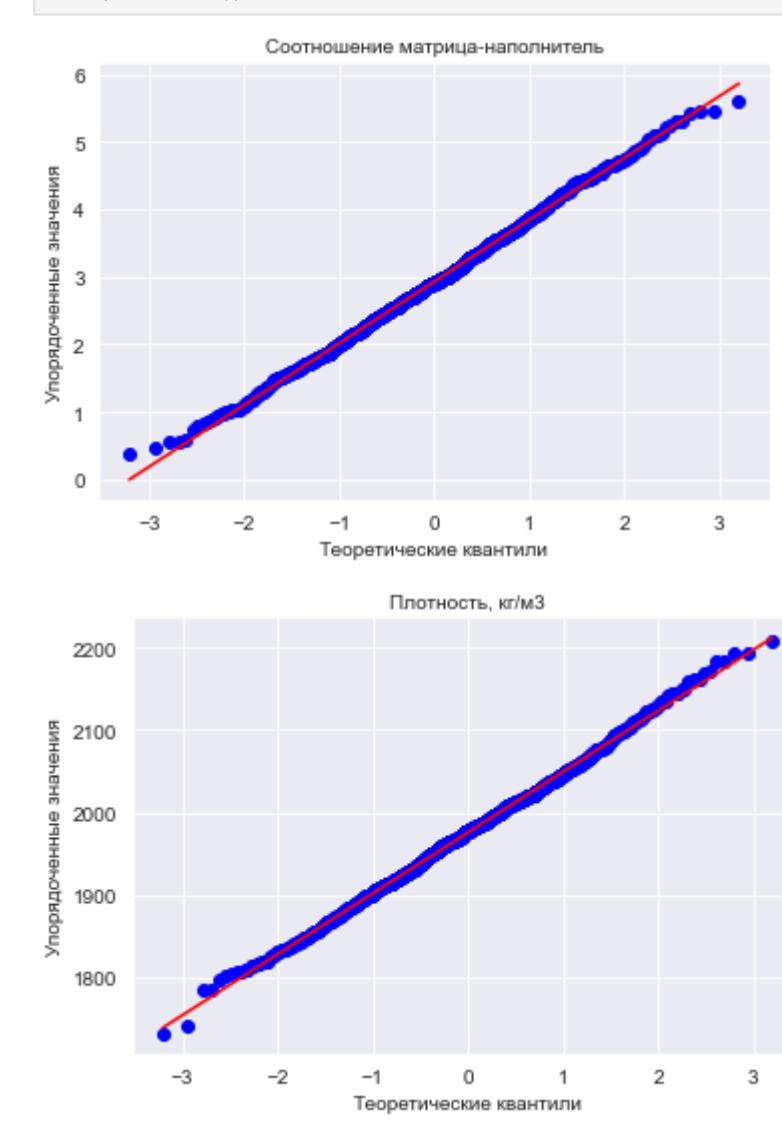


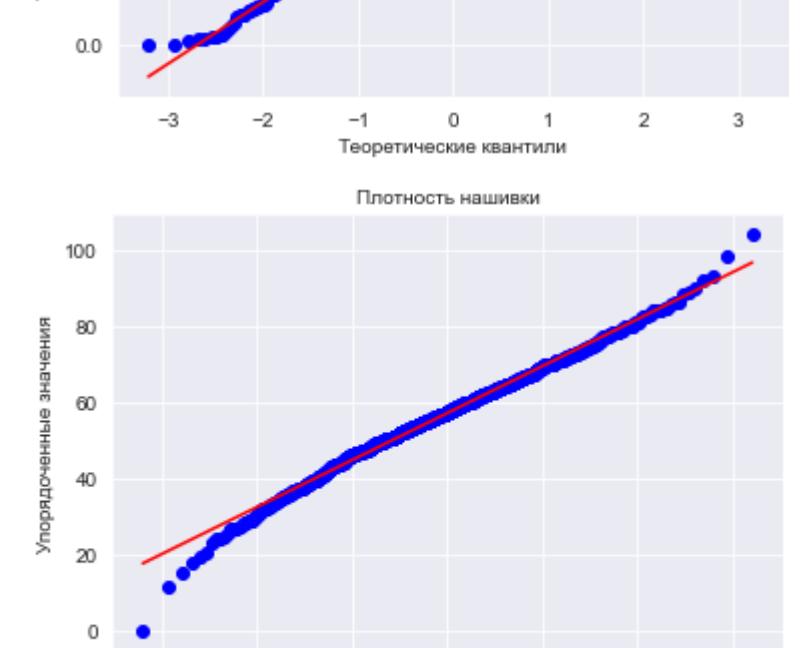
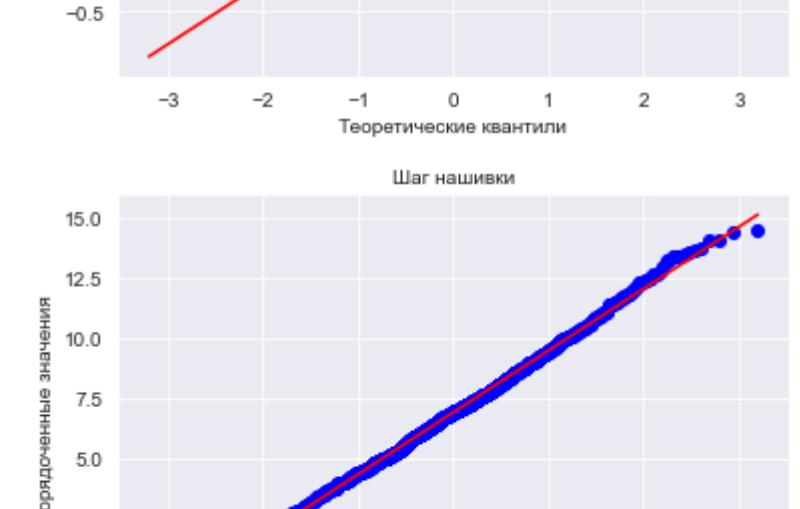
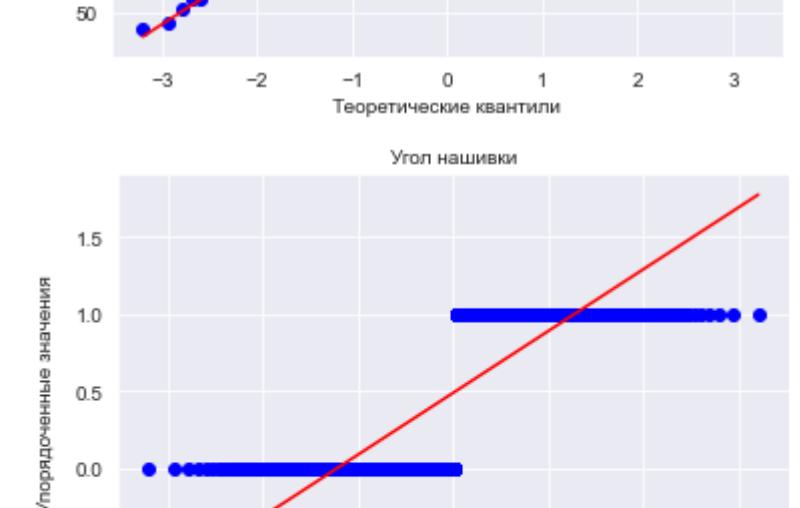
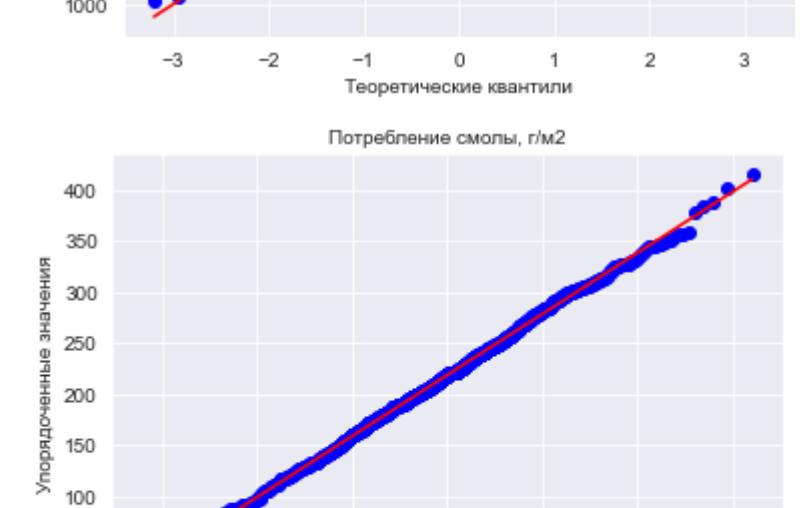
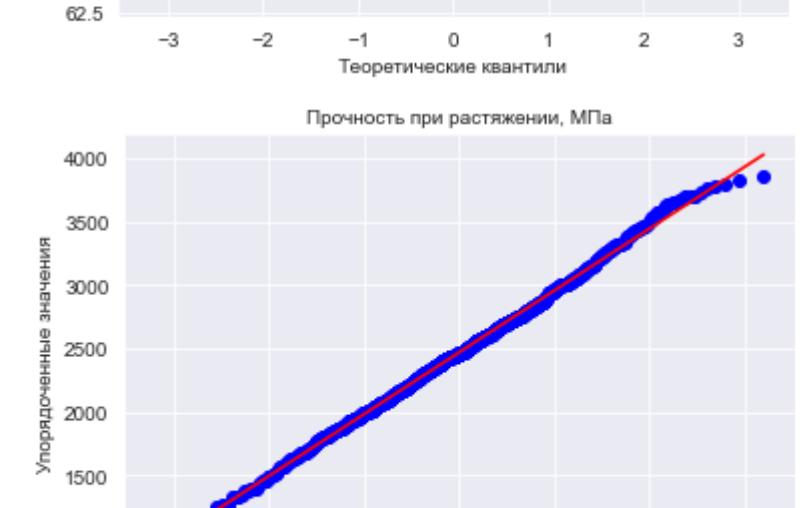
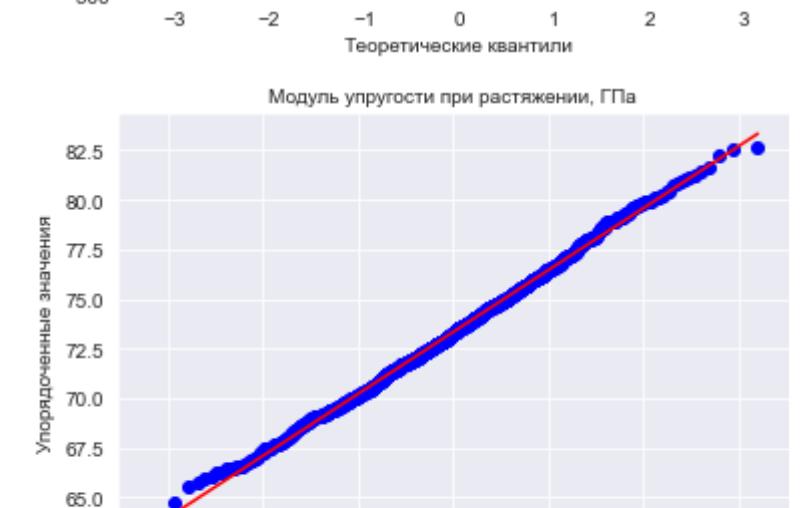
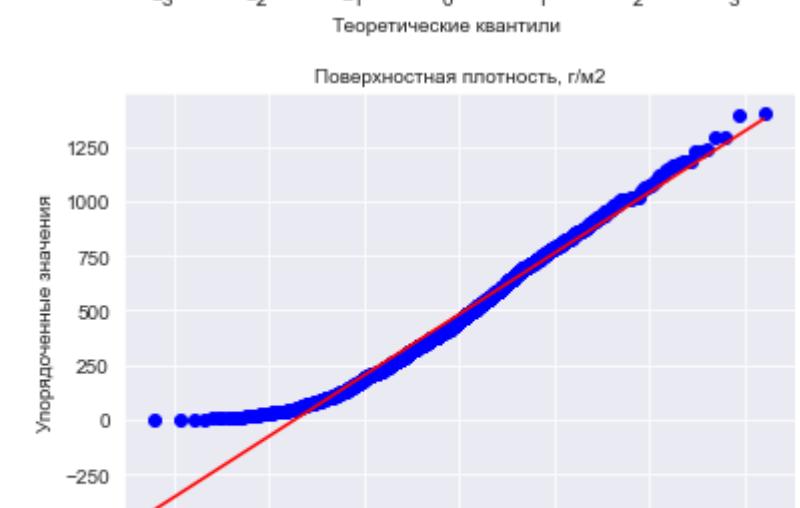
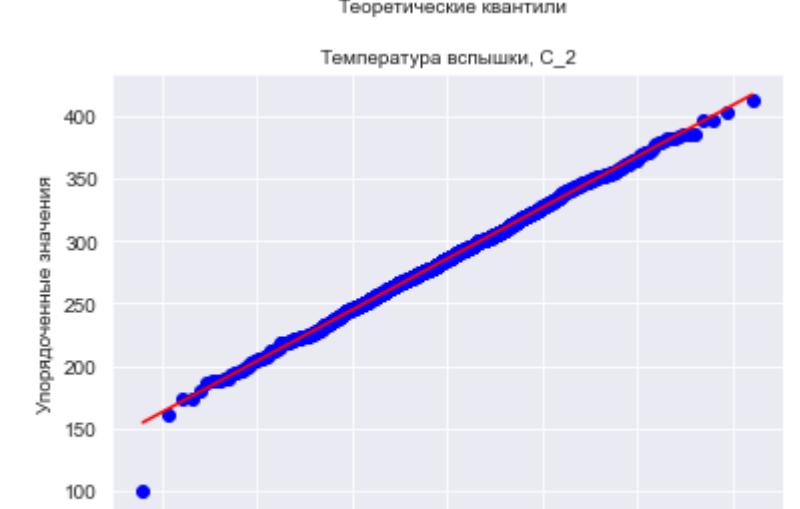
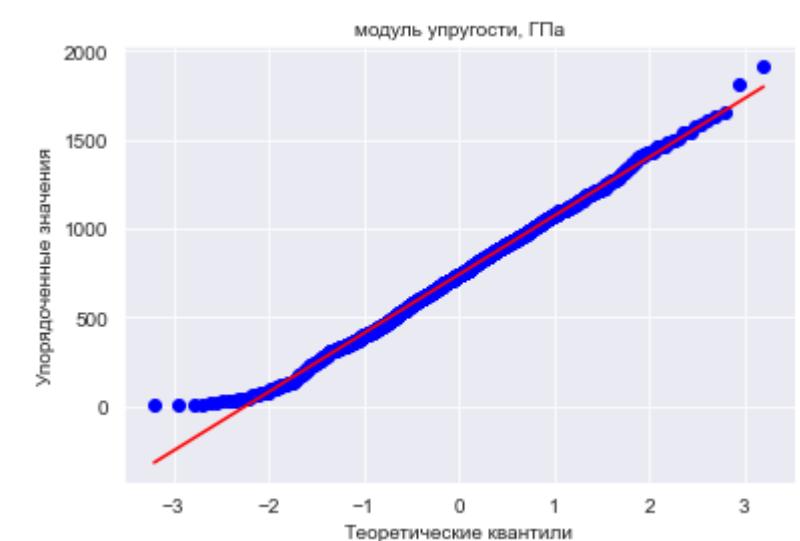
```
In [46]: # Проверим, где наши расстояния между - скatterплоты (блочный барометр)
g = sns.PairGrid(df[df.columns])
g.map_upper(sns.scatterplot, color = 'darkgreen')
g.map_upper(sns.kdeplot, color = 'darkgreen')
g.map_lower(sns.kdeplot, color = 'darkgreen')
plt.show()
# Корреляции нам
```

```
Out[46]: <function matplotlib.pyplot.show(close=None, block=None)>
```

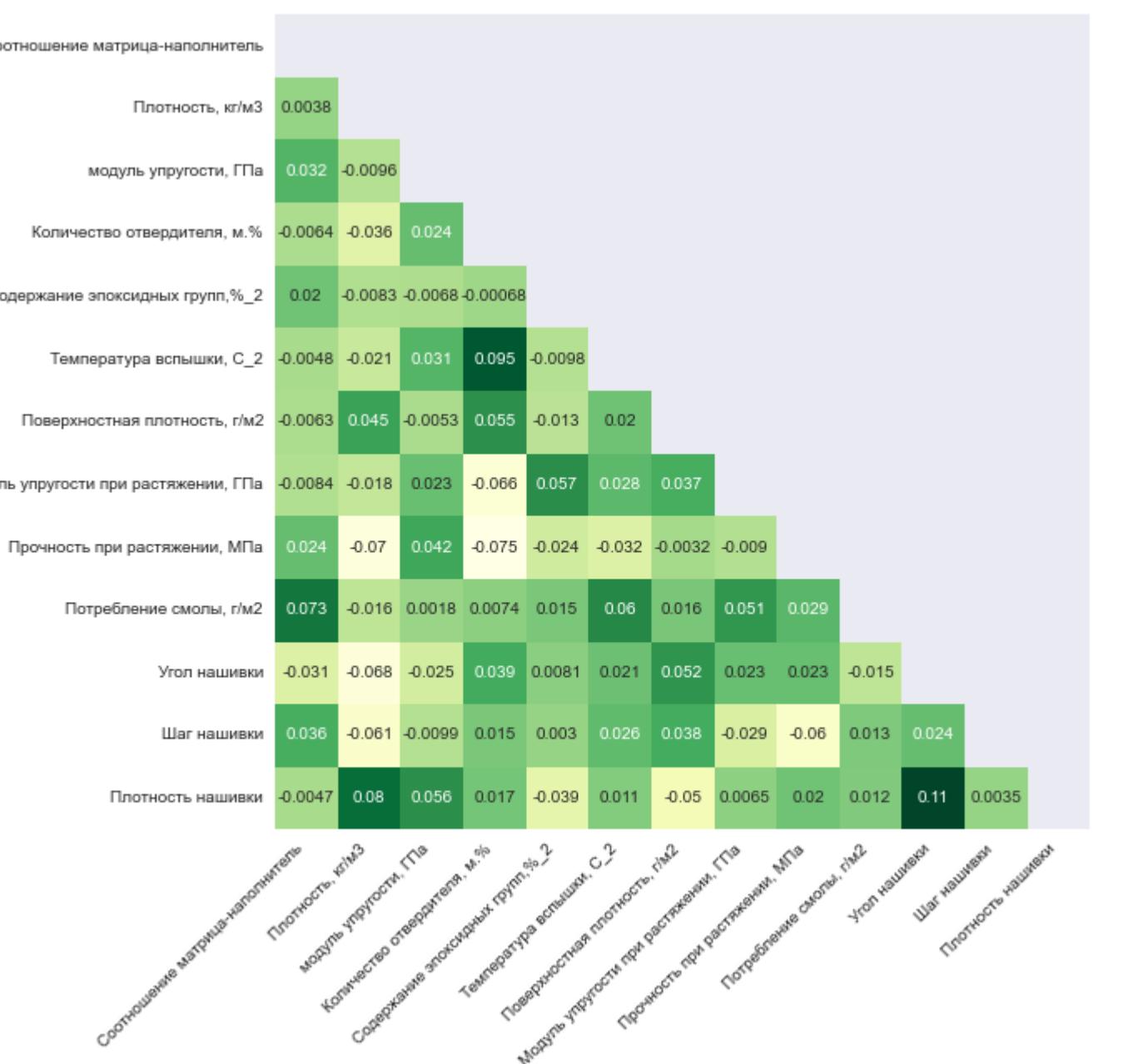


```
In [47]: # график qq
for i in df.columns:
    plt.figure(figsize = (6, 4))
    res = stats.probplot(df[i], plot = plt)
    plt.title(i, fontsize = 10)
    plt.xlabel("Теоретические квантили")
    plt.ylabel("Упорядоченные значения")
    plt.show()
```





```
# Создаю пустое для отображения большого графика
f, ax = plt.subplots(figsize = (11, 9))
# Визуализирую данные корреляции и создаем левобокую панель
sns.heatmap(df.corr(), mask = mask, annot = True, square = True, cmap = 'YlGn')
plt.xticks(rotation = 45, ha="right")
plt.show()
# Максимальная корреляция между Плотностью нашибки и углом нашибки составляет 0.11, что говорит об отсутствии зависимости между этими данными.
# Корреляция между всеми параметрами очень близка к 0, что говорит об отсутствии корреляционных связей между переменными.
```



In [49]: # График корреляции подтверждает данные теории композитных материалов. Мы видим, что на качество материала влияет температура фасонки и количество отвердителя из-за взаимодействия отвердителя с матрицей и наполнителем под влиянием температуры. Угол нашибки и плотность нашибки несомненно оказывают влияние на свойства материала. А потребление смолы и соотношение матрицы-наполнителя, плотности и плотности нашибки, модуль упругости и плотности нашибки имеют не особенно выраженную корреляцию.

Вывод на данном этапе работы: На наших "сырых" данных мы наблюдаем выбросы в каждом столбце, кроме столбца "Угол нашибки" и корреляция входных переменных очень слабая.