**Министерство науки и высшего образования**
**Российской Федерации**
**Федеральное государственное бюджетное образовательное учреждение**
**высшего образования**
**"Московский государственный технический университет**
**имени Н.Э. Баумана**
**(национальный исследовательский университет)"**
**(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ _ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ___СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)_____

# Отчет по лабораторной работе №1

«Создание “истории о данных” (Data Storytelling) »

по курсу «Методы машинного обучения».

ИСПОЛНИТЕЛЬ: Группа ИУ5-24М
Уралова Е.А.
ФИО
_____
подпись

"16" февраля 2024 г.

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю.Е.
ФИО
_____
подпись

"16" февраля 2024 г.

Москва – 2024

**Задание:**

1) Выбрать набор данных (датасет).

2) Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

– История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.

– На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.

– Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.

– Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.

– История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

3) Сформировать отчет и разместить его в своем репозитории на github.

**Выполнение:**

Импортируем библиотеки, загружаем датасет, смотрим данные датасета и количество строк и колонок.

In [5]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [6]:
```
data = pd.read_csv('forestfires.csv', sep=",")
```

In [7]:
```
# Первые 5 строк датасета
data.head()
```

Out[7]:

| | X | Y | month | day | FFMC | DMC | DC | ISI | temp | RH | wind | rain | area |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7 | 5 | mar | fri | 86.2 | 26.2 | 94.3 | 5.1 | 8.2 | 51 | 6.7 | 0.0 | 0.0 |
| **1** | 7 | 4 | oct | tue | 90.6 | 35.4 | 669.1 | 6.7 | 18.0 | 33 | 0.9 | 0.0 | 0.0 |
| **2** | 7 | 4 | oct | sat | 90.6 | 43.7 | 686.9 | 6.7 | 14.6 | 33 | 1.3 | 0.0 | 0.0 |
| **3** | 8 | 6 | mar | fri | 91.7 | 33.3 | 77.5 | 9.0 | 8.3 | 97 | 4.0 | 0.2 | 0.0 |
| **4** | 8 | 6 | mar | sun | 89.3 | 51.3 | 102.2 | 9.6 | 11.4 | 99 | 1.8 | 0.0 | 0.0 |

In [8]:
```
# Размер датасета - 517 строк, 13 колонок
data.shape
```

Out[8]:
```
(517, 13)
```

Посмотрим типы данных колонок и наличие в них пустых значений.

In [9]:
```
# Список колонок
data.columns
```

Out[9]:
```
Index(['X', 'Y', 'month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH',
       'wind', 'rain', 'area'],
      dtype='object')
```

In [10]:
```
# Список колонок с типами данных
data.dtypes
```

Out[10]:
```
X          int64
Y          int64
month     object
day       object
FFMC     float64
DMC      float64
DC       float64
ISI      float64
temp     float64
RH         int64
wind     float64
rain     float64
area     float64
dtype: object
```

In [11]:
```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
X - 0
Y - 0
month - 0
day - 0
FFMC - 0
DMC - 0
DC - 0
ISI - 0
temp - 0
RH - 0
wind - 0
rain - 0
area - 0
```

Основные харатеристики датасета.

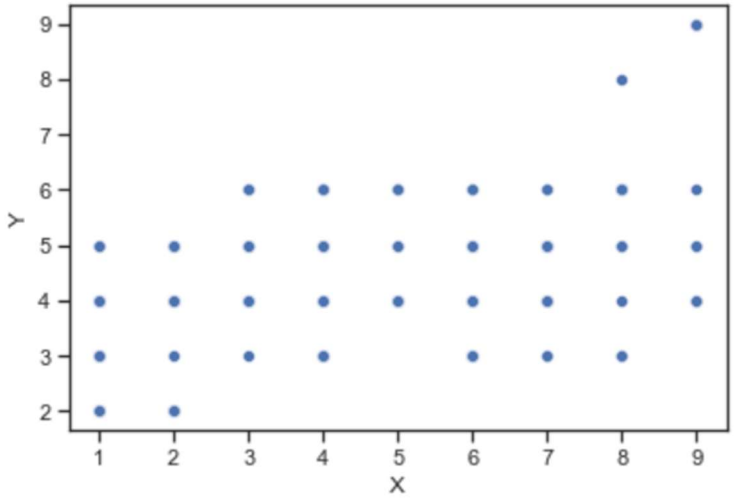
In [12]:
```
# Основные статистические характеристки набора данных
data.describe()
```

Out[12]:

| | X | Y | FFMC | DMC | DC | ISI | temp | RH | wind | rain | area |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 | 517.000000 |
| **mean** | 4.669246 | 4.299807 | 90.644681 | 110.872340 | 547.940039 | 9.021663 | 18.889168 | 44.288201 | 4.017602 | 0.021663 | 12.847292 |
| **std** | 2.313778 | 1.229900 | 5.520111 | 64.046482 | 248.066192 | 4.559477 | 5.806625 | 16.317469 | 1.791653 | 0.295959 | 63.655818 |
| **min** | 1.000000 | 2.000000 | 18.700000 | 1.100000 | 7.900000 | 0.000000 | 2.200000 | 15.000000 | 0.400000 | 0.000000 | 0.000000 |
| **25%** | 3.000000 | 4.000000 | 90.200000 | 68.600000 | 437.700000 | 6.500000 | 15.500000 | 33.000000 | 2.700000 | 0.000000 | 0.000000 |
| **50%** | 4.000000 | 4.000000 | 91.600000 | 108.300000 | 664.200000 | 8.400000 | 19.300000 | 42.000000 | 4.000000 | 0.000000 | 0.520000 |
| **75%** | 7.000000 | 5.000000 | 92.900000 | 142.400000 | 713.900000 | 10.800000 | 22.800000 | 53.000000 | 4.900000 | 0.000000 | 6.570000 |
| **max** | 9.000000 | 9.000000 | 96.200000 | 291.300000 | 860.600000 | 56.100000 | 33.300000 | 100.000000 | 9.400000 | 6.400000 | 1090.840000 |

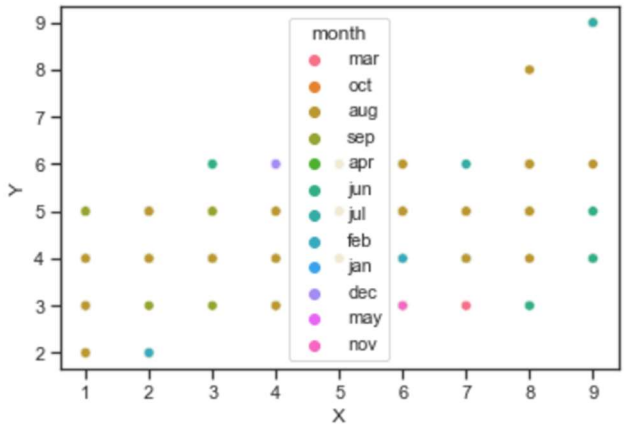
Расположение X и Y:

```
In [13]: sns.scatterplot(data = data, x = 'X', y = 'Y')
Out[13]: <AxesSubplot:xlabel='X', ylabel='Y'>
```




Вырубка леса по месяцам.

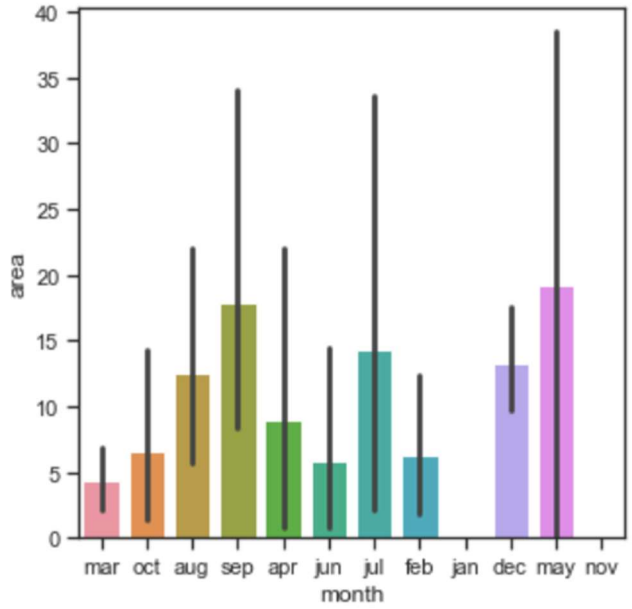
```
In [14]: sns.scatterplot(x = data['X'], y = data['Y'], hue = data['month'])
Out[14]: <AxesSubplot:xlabel='X', ylabel='Y'>
```



Количество вырубленного леса по месяцам в определенной области.

```
In [16]: fig, ax = plt.subplots(figsize=(5,5))
         sns.barplot(x = data['month'], y = data['area'])
```

```
Out[16]: <AxesSubplot:xlabel='month', ylabel='area'>
```

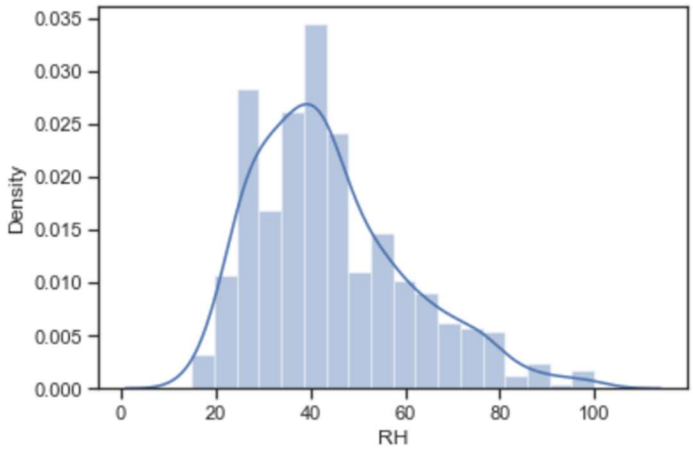



Распределение RH.

```
In [29]: sns.distplot(data['RH'])
```

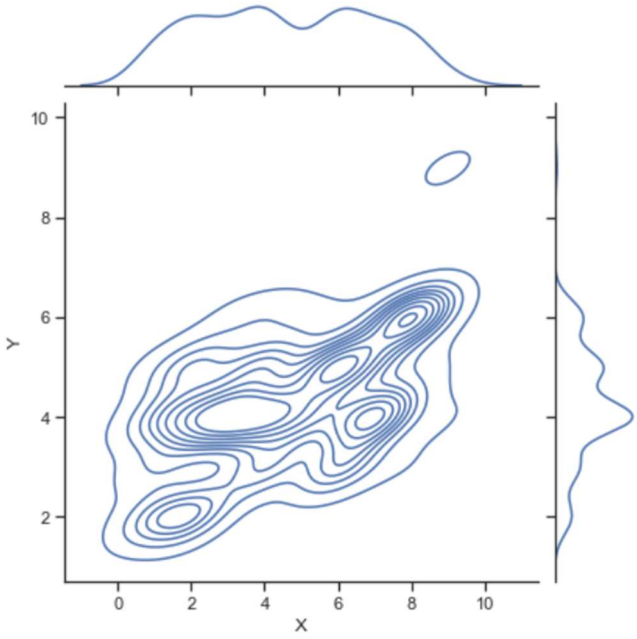
```
C:\Users\Asus\anaconda3\lib\site-packages\seaborn\distributions
will be removed in a future version. Please adapt your code to
bility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[29]: <AxesSubplot:xlabel='RH', ylabel='Density'>
```

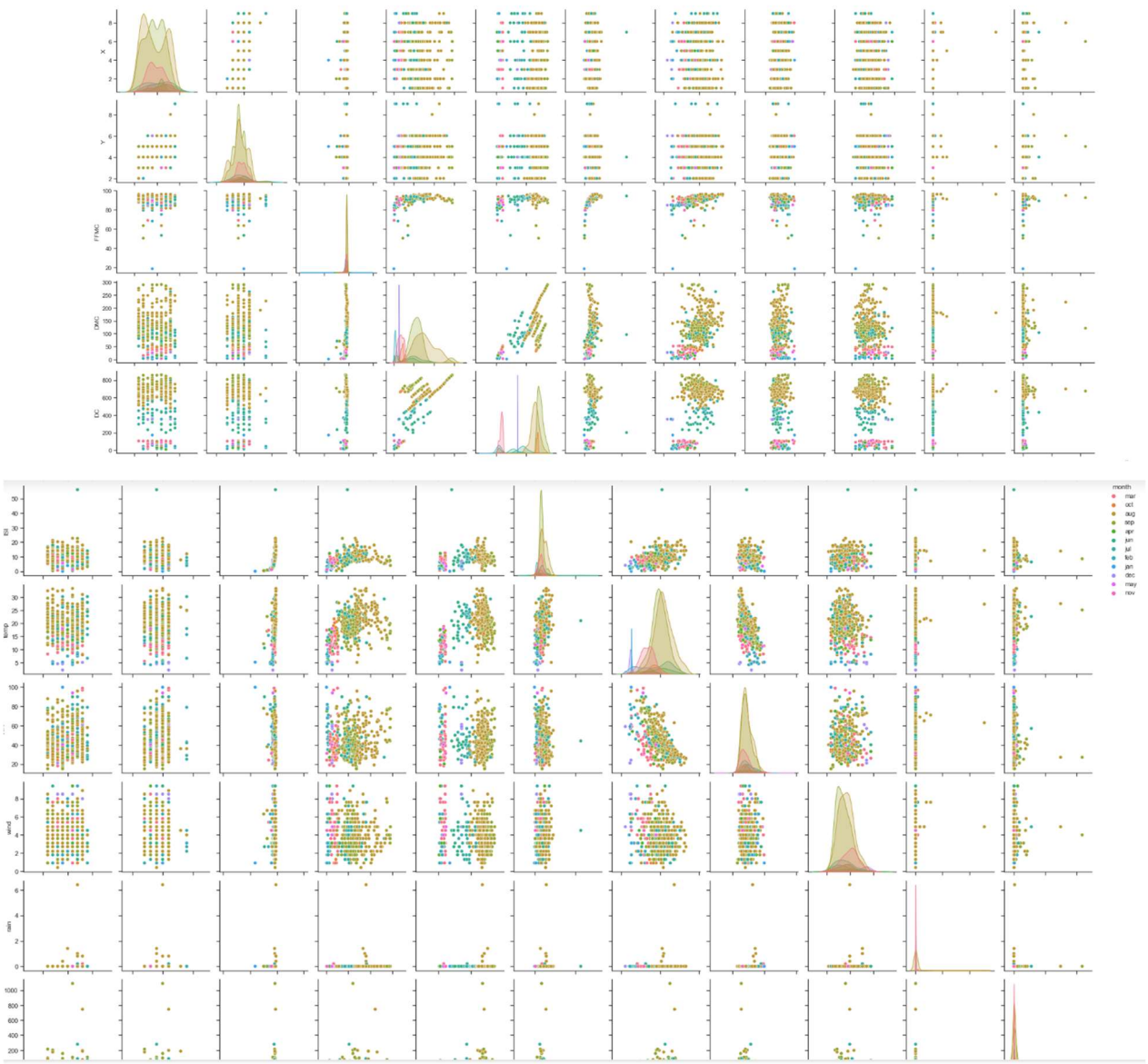


Наглядное изображение областей вырубки.

```
In [30]: sns.jointplot(x='X', y='Y', data=data, kind="kde")
Out[30]: <seaborn.axisgrid.JointGrid at 0x1a41ec6f100>
```

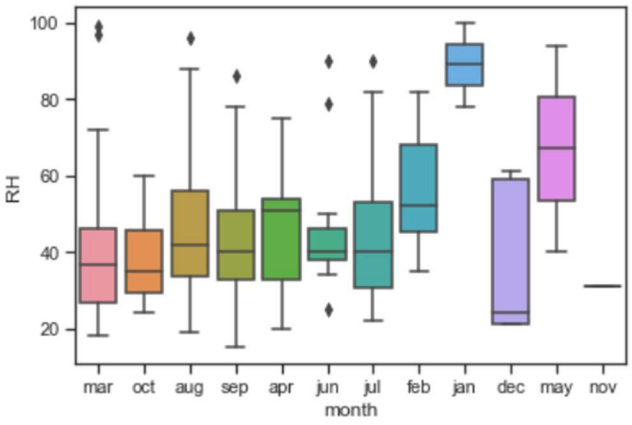


```
In [31]: sns.pairplot(data, hue="month")
Out[31]: <seaborn.axisgrid.PairGrid at 0x1a41ed684c0>
```



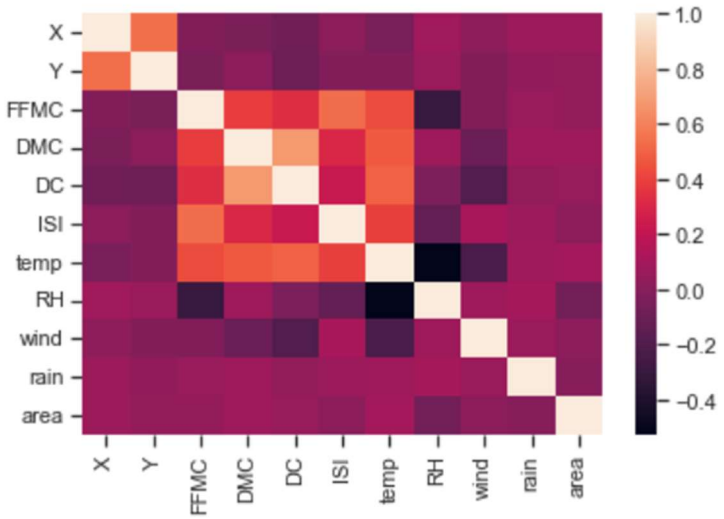
Распределение параметра RH, сгруппированных по месяцам.

```
In [32]: # Распределение параметра RH сгруппированные по month.
         sns.boxplot(x='month', y='RH', data=data)
Out[32]: <AxesSubplot:xlabel='month', ylabel='RH'>
```



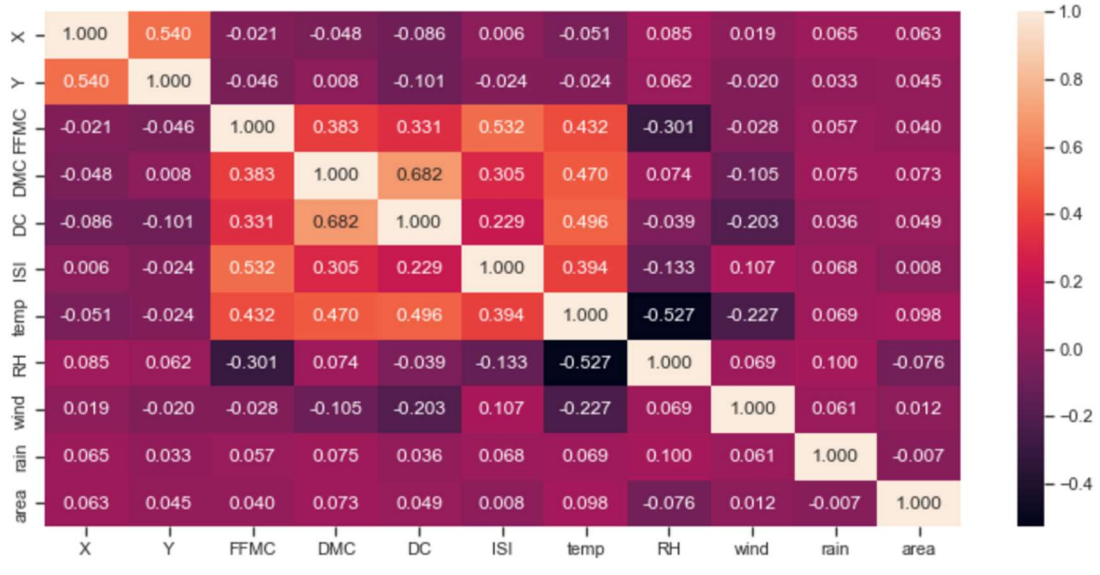
Корреляционная матрица.

```
In [33]: sns.heatmap(data.corr())
Out[33]: <AxesSubplot:>
```




Корреляционная матрица со значениями. Самая большая корреляция между DC и DMC.

```
In [37]: fig, ax = plt.subplots(figsize=(13,6))
         # Вывод значений в ячейках
         sns.heatmap(data.corr(), annot=True, fmt='.3f')
Out[37]: <AxesSubplot:>
```



**Вывод:**