Лабораторная работа №2

В файле «vgsale_1.csv» содержатся данные о видеоиграх, выпущенных с 1980 по 2020 гг. В файле представлены 16598 наблюдений, каждое из которых имеет 10 характеристик:

- **Name** – название игры,
- **Platform** – игровая платформа (PC, PSP, X360 и др.),
- **Year** – год выпуска игры,
- **Genre** – жанр игры,
- **Publisher** – издатель игры,
- **NA_Sales** – продажи в Северной Америке (в миллионах),
- **EU_Sales** – продажи в Европе (в миллионах),
- **JP_Sales** – продажи в Японии (в миллионах),
- **Other_Sales** – продажи в остальных странах мира (в миллионах),
- **Global_Sales** – объем продаж по всему миру.

Загрузите файл «vgsales.csv» в объект *DataFrame*, рассчитайте необходимые показатели и визуализируйте информацию, используя различные инструменты pandas. Проанализируйте полученные графики и сделайте выводы.

1. Игры каких жанров были наиболее популярны до 2000 года, а какие после? Оцените популярность жанров по количеству выпущенных игр и по объему продаж по всему миру. Для визуализации полученных результатов используйте столбиковые диаграммы.

Замечание. Одна и та же игра может встречать в выборке несколько раз, т.к. она может быть выпущена на нескольких платформах.

2. Отобразите на графике общее число видеоигр, выпущенных в каждом году.
3. Определите трех издателей выпустивших наибольшее количество видеоигр. Изобразите количество выпущенных издателями видеоигр для каждой платформы на столбиковой диаграмме (можно использовать диаграмму с накоплением).
4. Отобразите на круговых диаграммах доли суммарного объема продаж с 1980г. до 2000г. и с 2000г. до 2020г. в Северной Америке, Европе, Японии от объема продаж по всему миру.

Полезные функции и методы

1. Выбрать несколько столбцов из DataFrame можно с помощью записи df[[“Имя_столбца_1”, “Имя_столбца_2”]].
2. Метод .drop_duplicates() позволяет удалить дублирующиеся строки.
3. Чтобы выбрать строки, удовлетворяющие условию можете воспользоваться методом .loc(). Например, df.loc[df["Year"] < 2000].
4. Определить уникальные значения в столбце данных и посчитать сколько раз они встречаются можно с помощью метода .value_counts().
5. Метод .sort_values() позволяет отсортировать значения по строкам или столбцам.
6. Создавать графики можно с помощью библиотеки pandas. Сама по себе pandas не выполняет визуализацию данных, но она содержит «функции-обертки» вокруг функций библиотеки matplotlib. Например, метод .plot() pandas является оберткой к функции plot() библиотеки matplotlib. Функции-обертки значительно упрощают построение графиков. Другие виды графиков, часто использующихся при анализе данных, также можно построить с помощью метода .plot() pandas, установив параметр kind или обратиться к соответствующему методу как .plot.имя_метода() (табл.1).

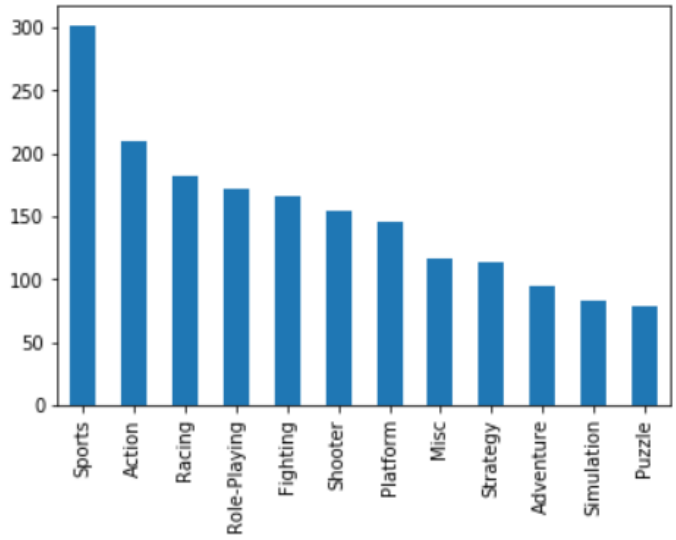
Таблица 1. Виды графиков и соответствующие им значения параметра

Значение параметра	Вид графика
.plot('kind = line') .plot.line()	Линейный график (назначен по умолчанию)
.plot(kind = 'bar') .plot.bar()	Столбиковая диаграмма
kind = 'barh' .plot.barh()	Горизонтальная столбиковая диаграмма
kind = 'hist' .plot.hist()	Гистограмма
kind = 'kde' .plot.kde() kind = 'density' .plot.density()	Графики ядерной оценки плотности
kind = 'pie' .plot.pie()	Круговая диаграмма
kind = 'area' .plot.area()	Площадная диаграмма
kind = 'box' .plot.box()	Диаграмма размаха
kind = 'scatter' .plot.scatter()	Диаграмма рассеяния

Пример. Изобразить на столбиковой диаграмме количество игр каждого жанра выпущенных с 1980 по 2020 гг.

```
df.Genre.value_counts().plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2477b143c18>
```



Узнать подробнее про возможности визуализации с помощью pandas можно по ссылке: https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html.