Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и управления»

Лабораторная работа №1

«Создание "истории о данных" (Data Storytelling)»

ИСПОЛНИТЕЛЬ:

Костян Алина Алексеевна
Группа ИУ5-21М

Москва 2021

Задание:

- Выбрать набор данных (датасет).

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

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. 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.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

- Сформировать отчет и разместить его в своем репозитории на github.

Ход выполнения:

Датасет - Video Game Sales (Продажи видеоигр)

Этот набор данных содержит список видеоигр с продажами более 100 000 копий. Он был создан при сканировании vgchartz.com.

Поля включают:

Рейтинг - Рейтинг общих продаж

Имя - Название игры.

Платформа - Платформа выпуска игр (например, ПК, PS4 и т. Д.).

Год - Год выпуска игры.

Жанр - Жанр игры

Издатель - 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 остальном мире (в миллионах)

Global_Sales - Общий объем продаж по всему миру.

Текст программы:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('vgsales.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

In [4]:
```
data.shape
```

Out[4]:
```
(16598, 11)
```

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          16598 non-null  int64  
 1   Name          16598 non-null  object 
 2   Platform      16598 non-null  object 
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object 
 5   Publisher     16540 non-null  object 
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

In [6]:
```
data.isnull().sum()
```

Out[6]:
```
Rank            0
Name            0
Platform        0
Year          271
Genre           0
Publisher      58
NA_Sales        0
EU_Sales        0
JP_Sales        0
Other_Sales     0
Global_Sales    0
dtype: int64
```

In [7]:
```
data['Genre'].value_counts()
```

Out[7]:
```
Action          3316
Sports          2346
Misc            1739
Role-Playing    1488
Shooter         1310
Adventure       1286
Racing          1249
Platform         886
Simulation       867
Fighting         848
Strategy         681
Puzzle           582
Name: Genre, dtype: int64
```

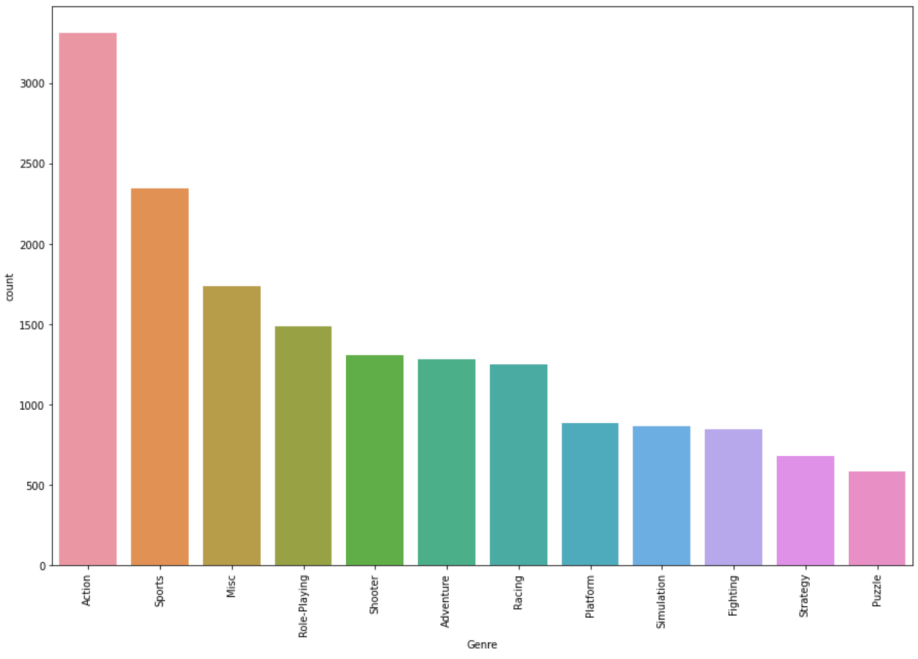
In [8]:
```
plt.figure(figsize=(13,10))
sns.heatmap(data.corr(), cmap = "Blues", annot=True, linewidth=3)
```

Out[8]: <AxesSubplot:>

Из матрицы корреляции видно, что наиболее сильно коррелируют показатели продаж Северной Америки и Европы

In [9]:
```
plt.figure(figsize=(15, 10))
sns.countplot(x="Genre", data=data, order = data['Genre'].value_counts().index)
plt.xticks(rotation=90)
```

Out[9]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
 [Text(0, 0, 'Action'),
  Text(1, 0, 'Sports'),
  Text(2, 0, 'Misc'),
  Text(3, 0, 'Role-Playing'),
  Text(4, 0, 'Shooter'),
  Text(5, 0, 'Adventure'),
  Text(6, 0, 'Racing'),
  Text(7, 0, 'Platform'),
  Text(8, 0, 'Simulation'),
  Text(9, 0, 'Fighting'),
  Text(10, 0, 'Strategy'),
  Text(11, 0, 'Puzzle')])
```



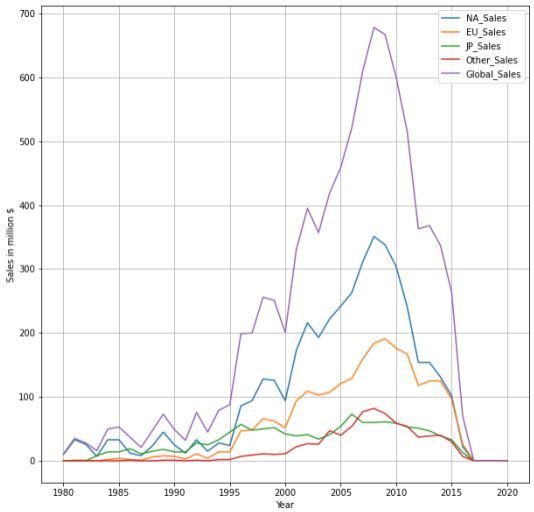
Из графика видно, что количество игр в жанре "Action" наибольшее, дальше идёт жанр "Sports" и так далее.

In [10]:
```
data_by_year = data.groupby(by  = 'Year').sum()
data_by_year.drop(columns="Rank",inplace=True)
data_by_year
```

Out[10]:

Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
1980.0	10.59	0.67	0.00	0.12	11.38
1981.0	33.40	1.96	0.00	0.32	35.77
1982.0	26.92	1.65	0.00	0.31	28.86
1983.0	7.76	0.80	8.10	0.14	16.79
1984.0	33.28	2.10	14.27	0.70	50.36
1985.0	33.73	4.74	14.56	0.92	53.94
1986.0	12.50	2.84	19.81	1.93	37.07
1987.0	8.46	1.41	11.63	0.20	21.74
1988.0	23.87	6.59	15.76	0.99	47.22
1989.0	45.15	8.44	18.36	1.50	73.45
1990.0	25.46	7.63	14.88	1.40	49.39
1991.0	12.76	3.95	14.78	0.74	32.23
1992.0	33.87	11.71	28.91	1.65	76.16
1993.0	15.12	4.65	25.33	0.89	45.98
1994.0	28.15	14.88	33.99	2.20	79.17
1995.0	24.82	14.90	45.75	2.64	88.11
1996.0	86.76	47.26	57.44	7.69	199.15
1997.0	94.75	48.32	48.87	9.13	200.98
1998.0	128.36	66.90	50.04	11.03	256.47
1999.0	126.06	62.67	52.34	10.05	251.27
2000.0	94.49	52.75	42.77	11.62	201.56
2001.0	173.98	94.89	39.86	22.76	331.47
2002.0	216.19	109.74	41.76	27.28	395.52
2003.0	193.59	103.81	34.20	26.01	357.85
2004.0	222.59	107.32	41.65	47.29	419.31
2005.0	242.61	121.94	54.28	40.58	459.94
2006.0	263.12	129.24	73.73	54.43	521.04
2007.0	312.05	160.50	60.29	77.60	611.13
2008.0	351.44	184.40	60.26	82.39	678.90
2009.0	338.85	191.59	61.89	74.77	667.30
2010.0	304.24	176.73	59.49	59.90	600.45
2011.0	241.06	167.44	53.04	54.39	515.99
2012.0	154.96	118.78	51.74	37.82	363.54
2013.0	154.77	125.80	47.59	39.82	368.11
2014.0	131.97	125.65	39.46	40.02	337.05
2015.0	102.82	97.71	33.72	30.01	264.44
2016.0	22.66	26.76	13.70	7.75	70.93
2017.0	0.00	0.00	0.05	0.00	0.05
2020.0	0.27	0.00	0.00	0.02	0.29

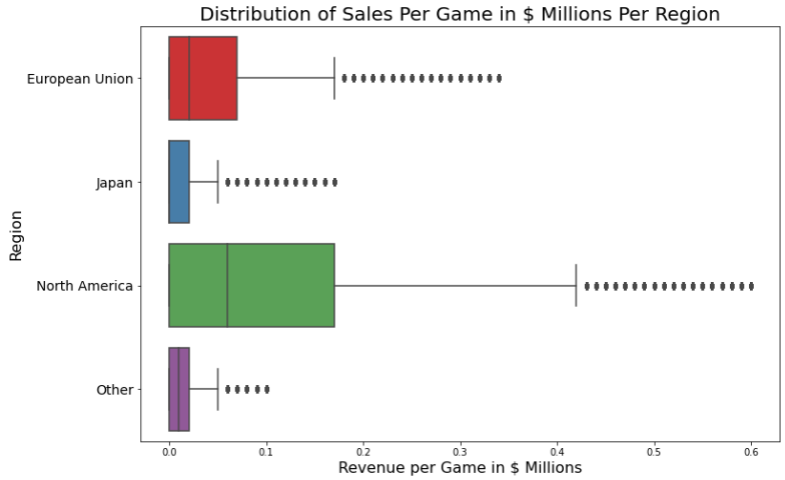
In [11]:
```
data_by_year=data_by_year.apply(lambda x : x.astype("int"))
data_by_year.plot.line(figsize=(10,10), grid="on");
plt.ylabel("Sales in million $");
```



Разбив продажи игр на разные года получим, что приблизительно в 2009 году произошёл скачок продаж и больше всех заработала на продаже Северная Америка. А вот Япония наоборот получила наименьший доход, даже по сравнению с другими странами.

In [12]:
```
data = pd.DataFrame([data['EU_Sales'], data['JP_Sales'], data['NA_Sales'], data['Other_Sales']]).T
regions = ['European Union', 'Japan', 'North America', 'Other']
q = data.quantile(0.90)
data = data[data < q]
plt.figure(figsize=(12,8))

colors = sns.color_palette("Set1", len(data))
ax = sns.boxplot(data=data, orient='h', palette=colors)
ax.set_xlabel(xlabel='Revenue per Game in $ Millions', fontsize=16)
ax.set_ylabel(ylabel='Region', fontsize=16)
ax.set_title(label='Distribution of Sales Per Game in $ Millions Per Region', fontsize=20)
ax.set_yticklabels(labels=regions, fontsize=14)
plt.show()
```



Видим, что Северная Америка лидирует по продажам игр как в размахе, так и в значении медианы.

In [13]:
```
top_sale_reg = data[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
# pd.DataFrame(top_sale_reg.sum(), columns=['a', 'b'])
top_sale_reg = top_sale_reg.sum().reset_index()
top_sale_reg = top_sale_reg.rename(columns={"index": "region", 0: "sale"})
top_sale_reg
```

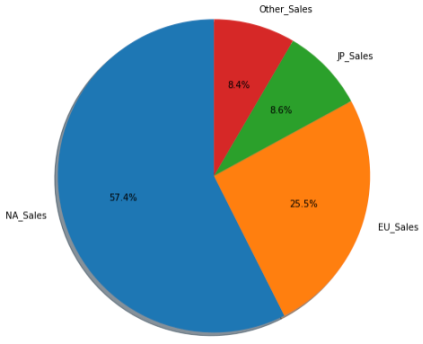
Out[13]:

	region	sale
0	NA_Sales	1675.07
1	EU_Sales	742.82
2	JP_Sales	252.00
3	Other_Sales	246.07

In [14]:
```
labels = top_sale_reg['region']
sizes = top_sale_reg['sale']
```

In [15]:
```
plt.figure(figsize=(10, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
```

Out[15]:
```
([<matplotlib.patches.Wedge at 0x7f9c48999f90>,
  <matplotlib.patches.Wedge at 0x7f9c489a6a10>,
  <matplotlib.patches.Wedge at 0x7f9c489b6510>,
  <matplotlib.patches.Wedge at 0x7f9c489b6fd0>],
 [Text(-1.070049910822957, -0.2549376165805706, 'NA_Sales'),
  Text(1.0499814336634996, -0.3279313784344864, 'EU_Sales'),
  Text(0.7904091122375416, 0.765018584932328, 'JP_Sales'),
  Text(0.28821760581657324, 1.0615698807414247, 'Other_Sales')],
 [Text(-0.5836635877216129, -0.1390568817712203, '57.4%'),
  Text(0.572717145634636, -0.17887166096426527, '25.5%'),
  Text(0.431132243038659, 0.4172828645085425, '8.6%'),
  Text(0.1572096031726763, 0.5790381167680497, '8.4%')])
```



Аналогично видим, что Северная Америка имеет большую долю в продаже игр.

Исходя из проведённого анализа, получаем, что наибольшую прибыль от продажи видеоигр получают в Северной Америке. В 2009 году произошёл скачок продаж, в котором Япония не проявила себя. А также самым популярным жанром является "Action"