

МГТУ им. Н. Э. Баумана, кафедра ИУ5
курс “Методы машинного обучения”

Лабораторная работа №1
«Создание "истории о данных" (Data Storytelling)»

ВЫПОЛНИЛ:

Сергеев М.К.

Группа: ИУ5-22М

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва 2022

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.

Описание набора данных:

Ранг - Рейтинг продаж;

Название - Название игры;

Платформа – Игровая платформа (например, PC, 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.

Выполнение работы:

Импортирование необходимых библиотек

```
In [91]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Исследуем основные характеристики датасета

```
In [92]: data = pd.read_csv("/content/drive/MyDrive/data/Video_Games_Sales.csv")
```

```
In [93]: data.head()
```

```
Out[93]:
```

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3
1	Super Mario	NES	1985.0	Platform	Nintendo	29.08	3.58	6
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3
3	Wii Sports	Wii	2009.0	Sports	Nintendo	15.61	10.93	3
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role- Playing	Nintendo	11.27	8.89	10

```
In [94]: data.shape
```

```
Out[94]: (16719, 16)
```

```
In [95]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16719 entries, 0 to 16718
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
----  -----
0   Name                  16717 non-null  object
1   Platform              16719 non-null  object
2   Year_of_Release       16450 non-null  float64
3   Genre                 16717 non-null  object
4   Publisher             16665 non-null  object
5   NA_Sales              16719 non-null  float64
6   EU_Sales              16719 non-null  float64
7   JP_Sales              16719 non-null  float64
8   Other_Sales           16719 non-null  float64
9   Global_Sales          16719 non-null  float64
10  Critic_Score          8137 non-null   float64
11  Critic_Count          8137 non-null   float64
```

```
12 User_Score      10015 non-null object
13 User_Count      7590 non-null  float64
14 Developer       10096 non-null object
15 Rating          9950 non-null object
dtypes: float64(9), object(7)
memory usage: 2.0+ MB
```

```
In [96]: data.isnull().sum()
```

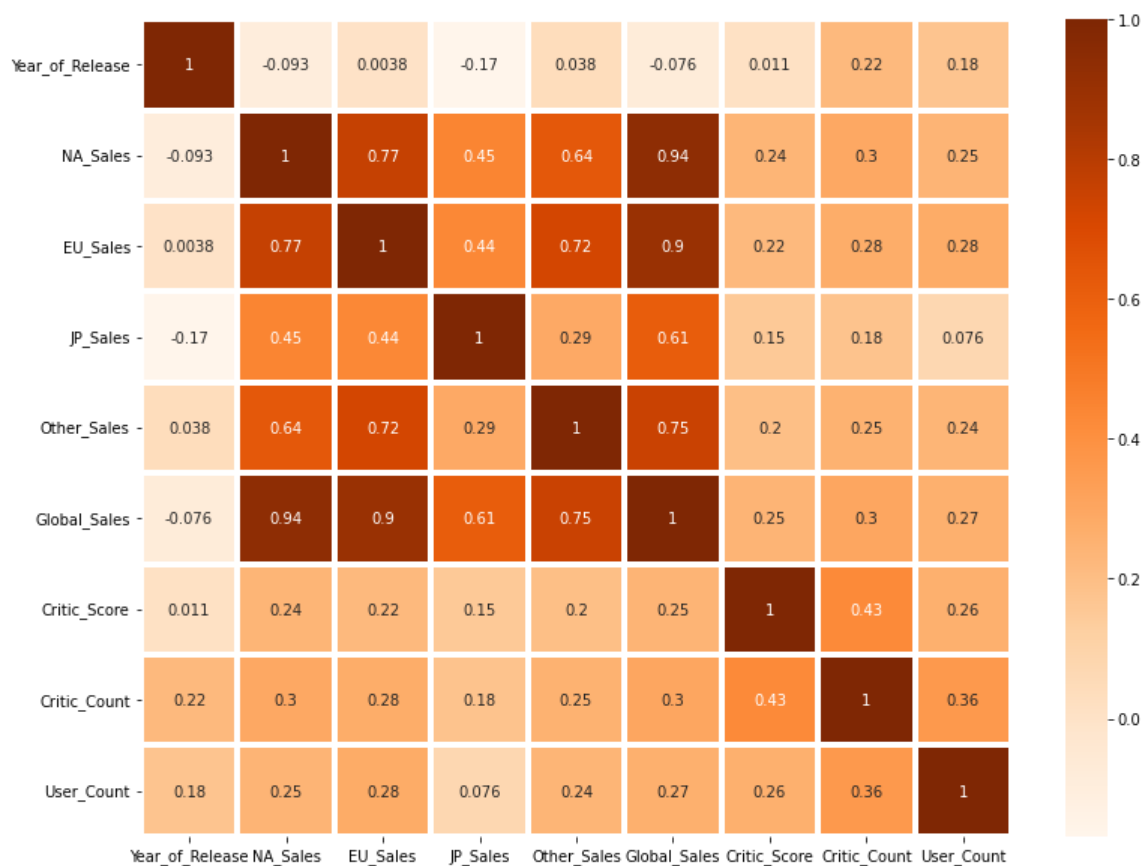
```
Out[96]: Name                2
Platform              0
Year_of_Release      269
Genre                 2
Publisher             54
NA_Sales              0
EU_Sales              0
JP_Sales              0
Other_Sales           0
Global_Sales          0
Critic_Score          8582
Critic_Count          8582
User_Score            6704
User_Count            9129
Developer             6623
Rating                6769
dtype: int64
```

```
In [97]: data['Genre'].value_counts()
```

```
Out[97]: Action            3370
Sports          2348
Misc            1750
Role-Playing    1500
Shooter         1323
Adventure       1303
Racing          1249
Platform        888
Simulation      874
Fighting        849
Strategy        683
Puzzle          580
Name: Genre, dtype: int64
```

```
In [98]: plt.figure(figsize=(13,10))
sns.heatmap(data.corr(), cmap = "Oranges", annot=True, linewidth=3)
```

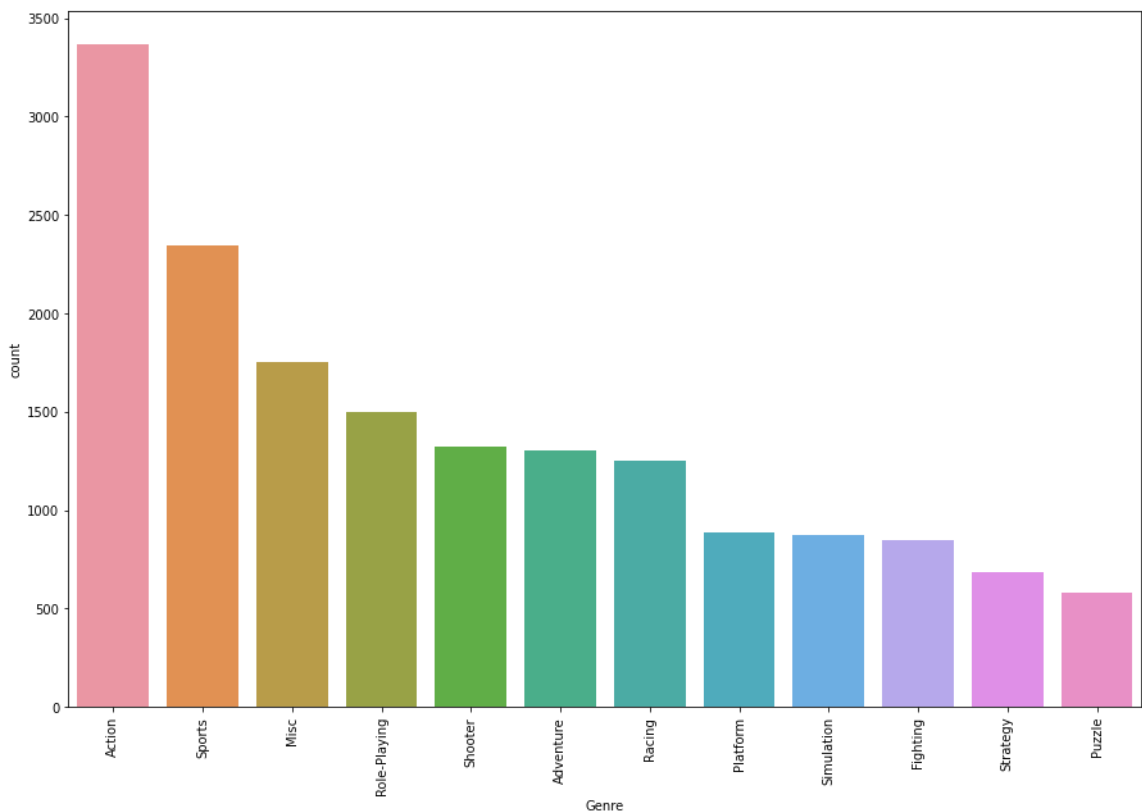
```
Out[98]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8f952c7190>
```



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

```
In [99]: plt.figure(figsize=(15, 10))
sns.countplot(x="Genre", data=data, order = data['Genre'].value_counts())
plt.xticks(rotation=90)
```

```
Out[99]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
<a list of 12 Text major ticklabel objects>)
```



Из гистограммы видно, что больше всего игр в жанре "Action", меньше игра в жанре "Sports" и т.д.

In [100...

```
data_by_year = data.groupby(by = 'Year_of_Release').sum()
data_by_year.drop(columns=["Critic_Count", "User_Count", "Critic_Score"])
data_by_year
```

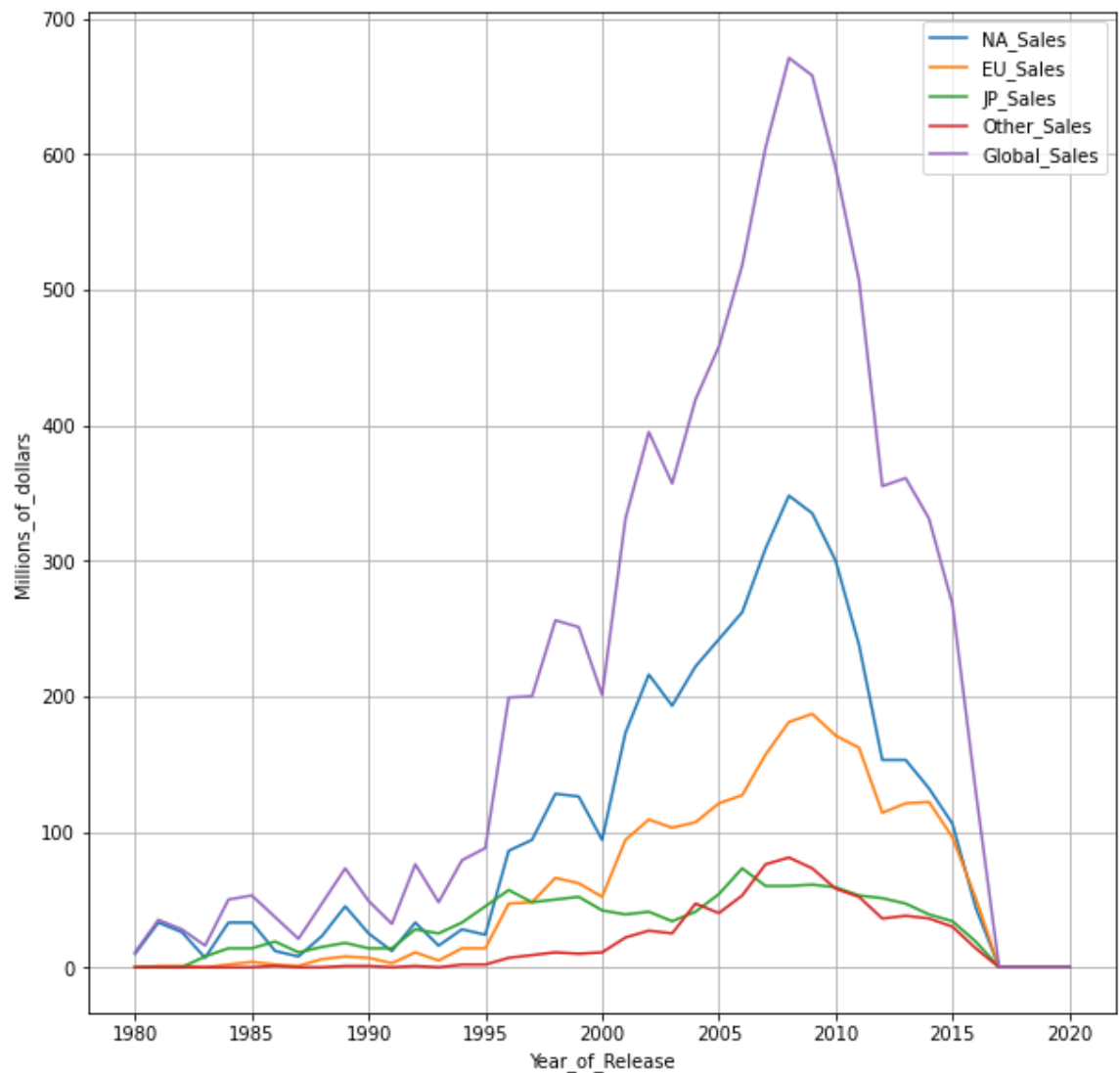
Out[100...

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Year_of_Release					
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	33.89	11.71	28.91	1.65	76.17
1993.0	16.90	5.18	25.36	0.97	48.40
1994.0	28.16	14.88	33.99	2.20	79.18
1995.0	24.83	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	11.01	256.45
1999.0	126.06	62.67	52.34	10.04	251.25
2000.0	94.50	52.77	42.77	11.62	201.58
2001.0	173.98	94.89	39.86	22.73	331.47
2002.0	216.19	109.75	41.76	27.27	395.51
2003.0	193.61	103.81	34.20	25.92	357.80
2004.0	222.51	107.28	41.65	47.24	419.05
2005.0	242.15	121.11	54.27	40.29	458.31
2006.0	262.13	127.89	73.74	53.95	518.22
2007.0	309.89	157.82	60.29	76.75	605.37
2008.0	348.69	181.14	60.25	81.42	671.79
2009.0	335.55	187.94	61.89	73.44	658.88
2010.0	300.65	171.42	59.49	58.57	590.59
2011.0	238.79	162.97	53.07	52.75	507.79
2012.0	153.26	114.59	51.80	36.19	355.84
2013.0	153.65	121.55	47.69	38.35	361.24
2014.0	132.27	122.74	39.69	36.83	331.51
2015.0	106.86	96.72	34.09	30.31	268.05
2016.0	44.93	51.22	19.31	14.48	130.10
2017.0	0.00	0.00	0.06	0.00	0.06
2020.0	0.27	0.00	0.00	0.02	0.29

In [101]...

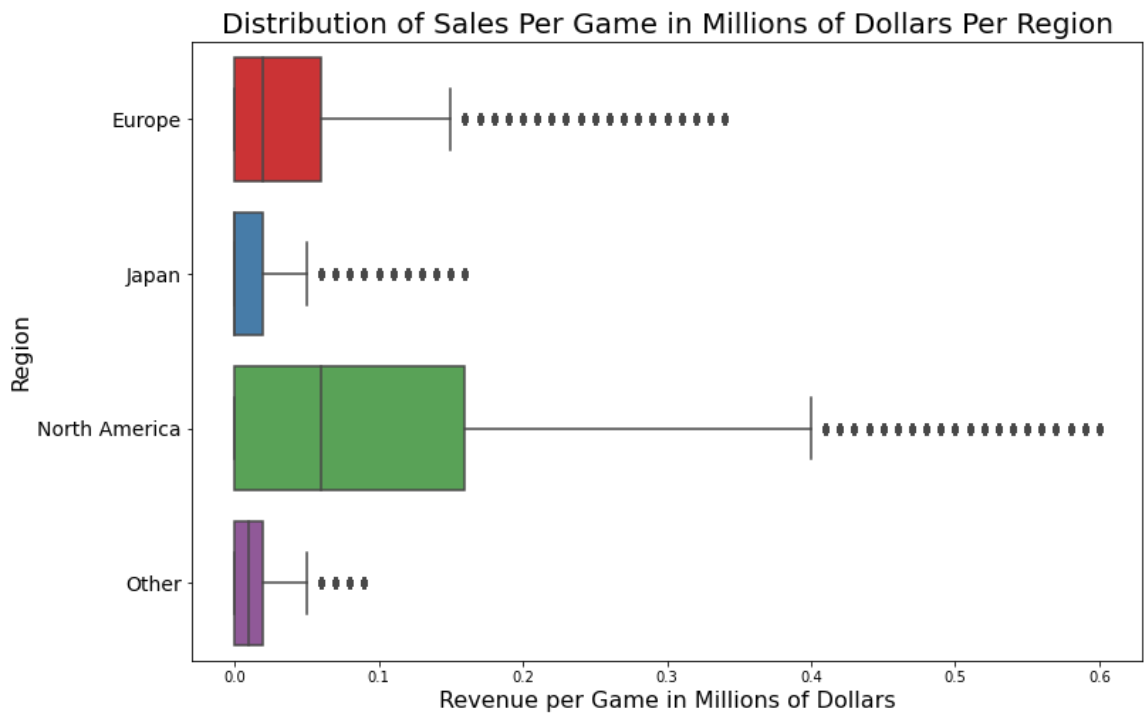
```
data_by_year=data_by_year.apply(lambda x : x.astype("int"))
data_by_year.plot.line(figsize=(10,10), grid="on");
plt.ylabel("Millions_of_dollars");
```

Разбив игры по продажам в разные года по разным регионам, можно заметить, что наибольшие продажи игр по всему миру пришли на 2009 год. При этом, среди регионов больше всего игр было продано в Северной Америке, а меньше всего в Японии

```
In [102... data = pd.DataFrame([data['EU_Sales'], data['JP_Sales'], data['NA_Sales']
regions = ['Europe', '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 of Dollars', fontsize
ax.set_ylabel(ylabel='Region', fontsize=16)
ax.set_title(label='Distribution of Sales Per Game in Millions of Dollar
ax.set_yticklabels(labels=regions, fontsize=14)
plt.show()
```



Из диаграммы "Ящик с усами" видно, что Северная Америка лидирует по продажам игр как в размахе, так и по медианному значению

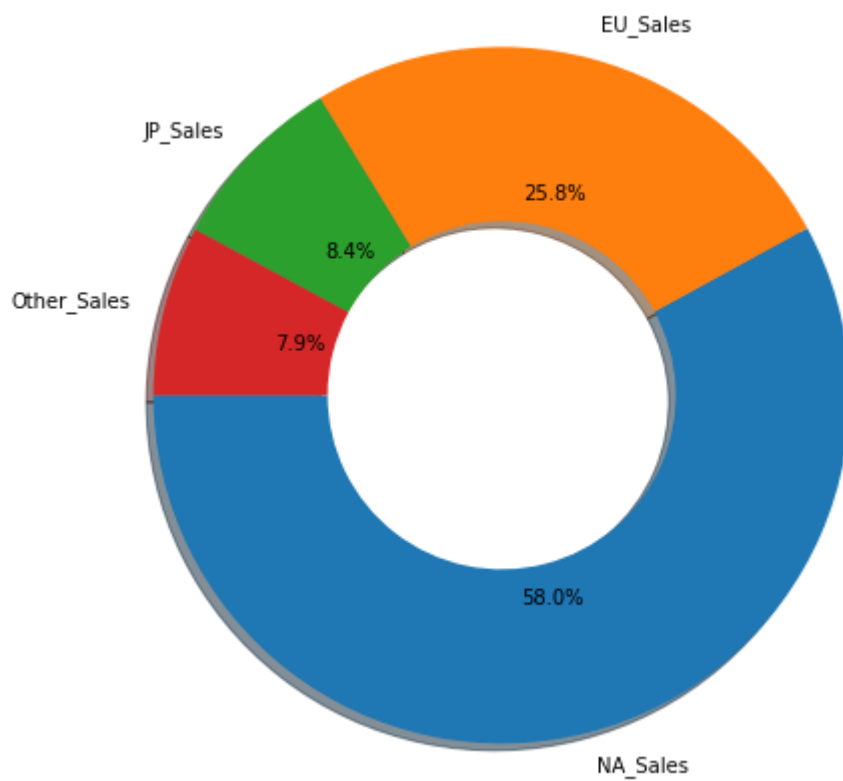
```
In [103... top_sale_reg = data[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
top_sale_reg = top_sale_reg.sum().reset_index()
top_sale_reg = top_sale_reg.rename(columns={"index": "Region", 0: "Sales"})
top_sale_reg
```

```
Out[103...      Region  Sales
0  NA_Sales  1674.46
1  EU_Sales   744.53
2  JP_Sales   242.07
3  Other_Sales  227.81
```

```
In [104... labels = top_sale_reg['Region']
sizes = top_sale_reg['Sales']
```

```
In [105... plt.figure(figsize=(10, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', wedgeprops=dict(width=0.5))
```

```
Out[105... ([<matplotlib.patches.Wedge at 0x7f8f9382a7d0>,
<matplotlib.patches.Wedge at 0x7f8f937b3190>,
<matplotlib.patches.Wedge at 0x7f8f937b3d10>,
<matplotlib.patches.Wedge at 0x7f8f937ba850>],
[Text(0.2723019312452782, -1.0657634156979174, 'NA_Sales'),
Text(0.2836793891660941, 1.062791609000726, 'EU_Sales'),
Text(-0.7982850337767683, 0.7567965412500403, 'JP_Sales'),
Text(-1.0664161445551974, 0.2697343260173396, 'Other_Sales')],
[Text(0.14852832613378808, -0.5813254994715913, '58.0%'),
Text(0.15473421227241493, 0.5797045140003959, '25.8%'),
Text(-0.4354282002418736, 0.4127981134091128, '8.4%'),
Text(-0.581681533393744, 0.14712781419127613, '7.9%')])
```



Из кольцевой диаграммы также видно, что Северная Америка имеет наибольшую долю продаж во всем мире

На основании проведенного анализа можно сделать следующий вывод:

- Наиболее популярным жанром игр во всем мире является "Action"; Самую большую долю продаж в мире
- имеет Северная Америка;
- В 2009 году произошел скачок продаж видеоигр по всему миру, кроме Японии.