

Lecture 7:

Data Visualization

Introduction to Python
Elf Data Science Course

Dr. Benjamin M. Abdel-Karim

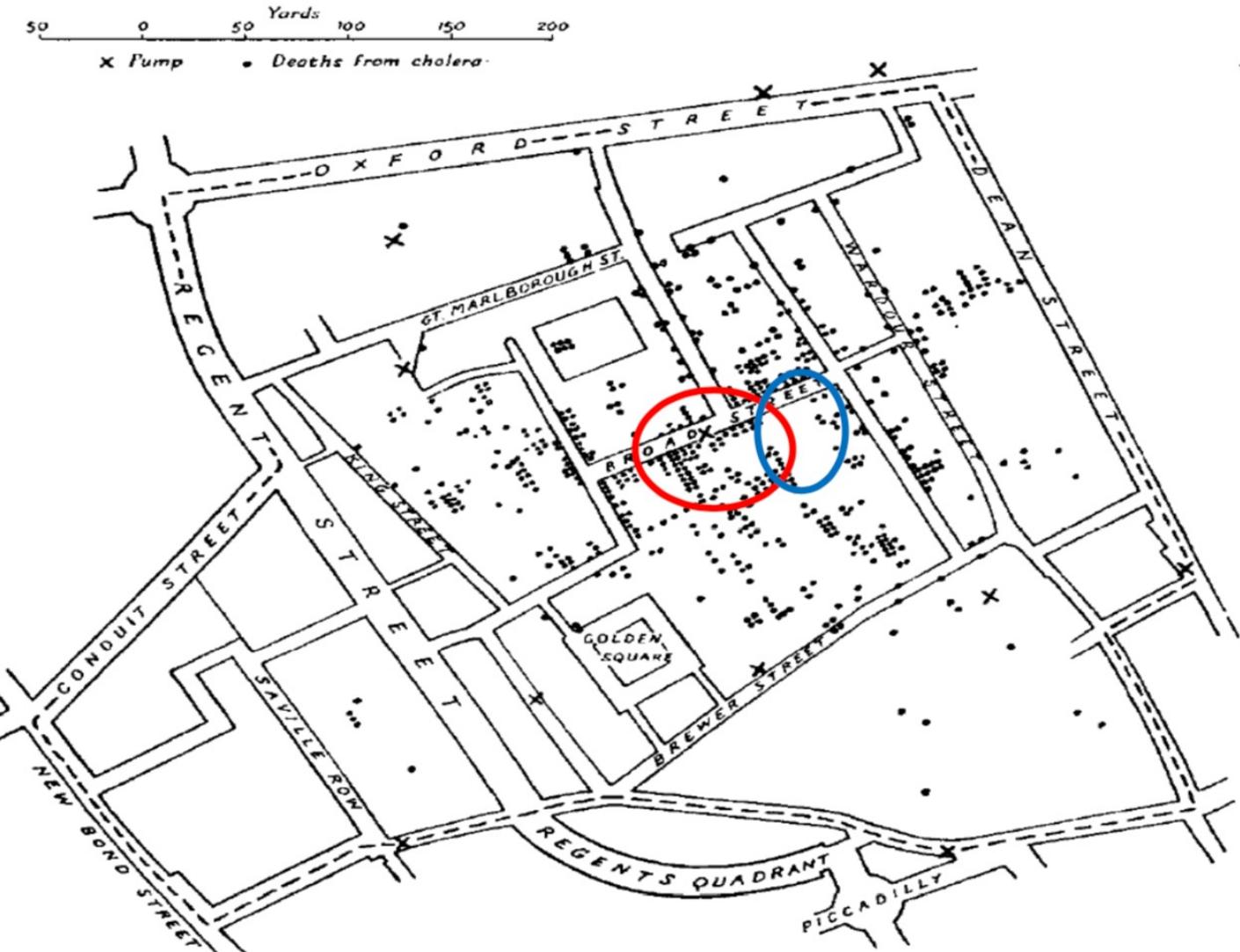
Motivation

- Big Data but Data chaos
- An image says more than 1000 words
- The goal from data to knowledge



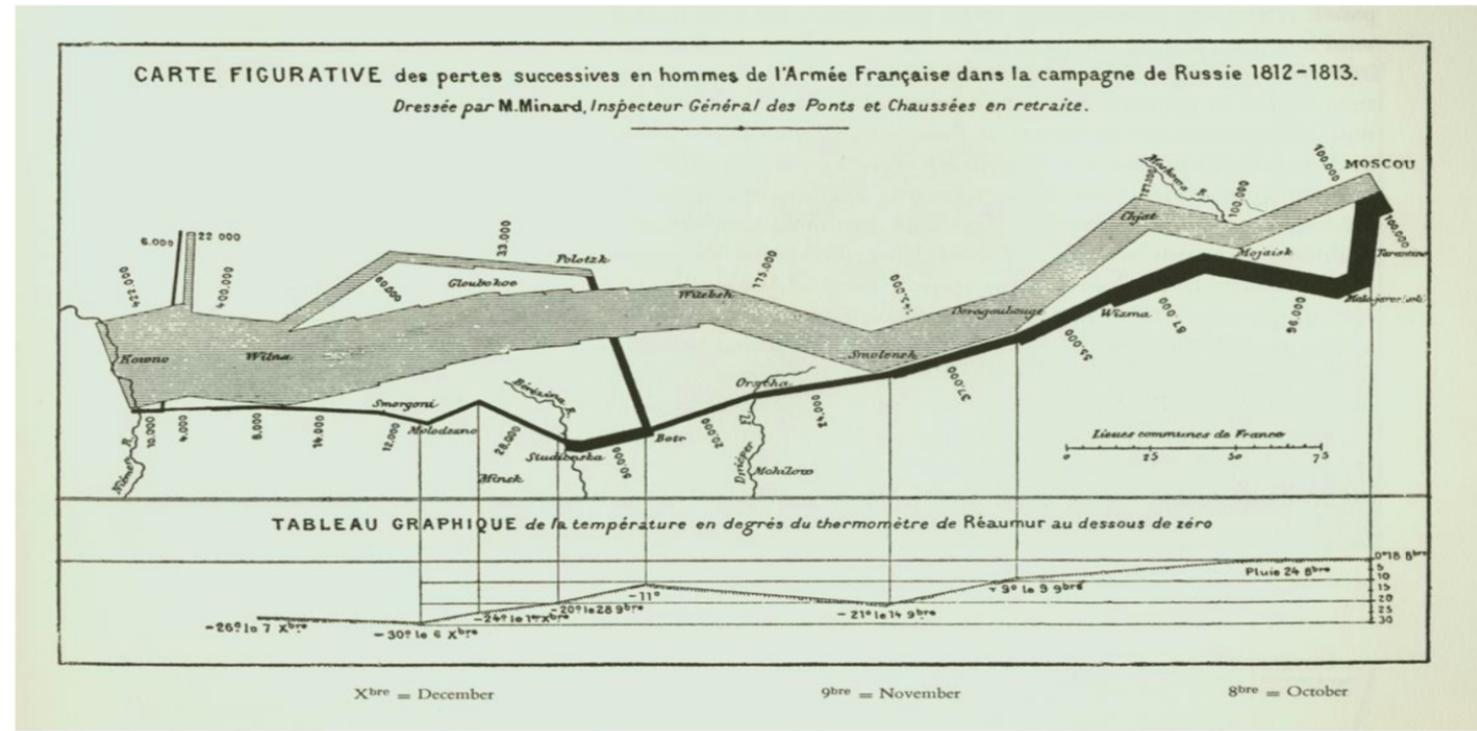
Motivation

- Dr. John Snow (1854):
Map of the Cholera
Epidemic of London
(1853)
- Shows cases of illness
sorted by street districts
- Contaminated pump
might caused the local
cholera cases
- Water pump affected has
been shut down



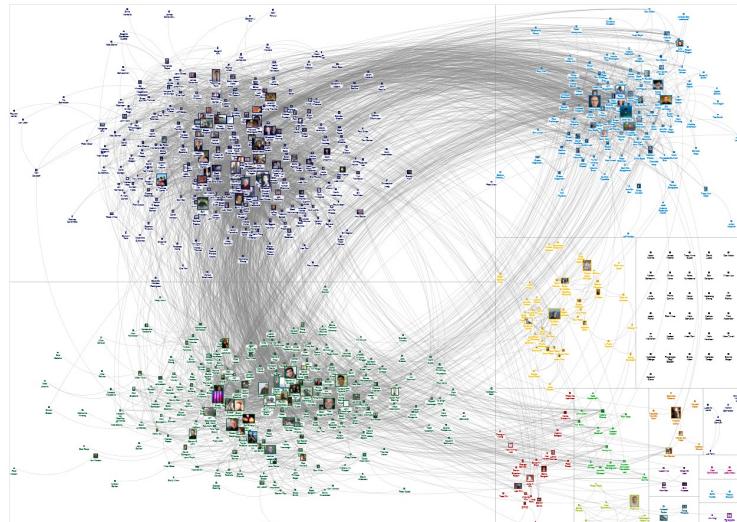
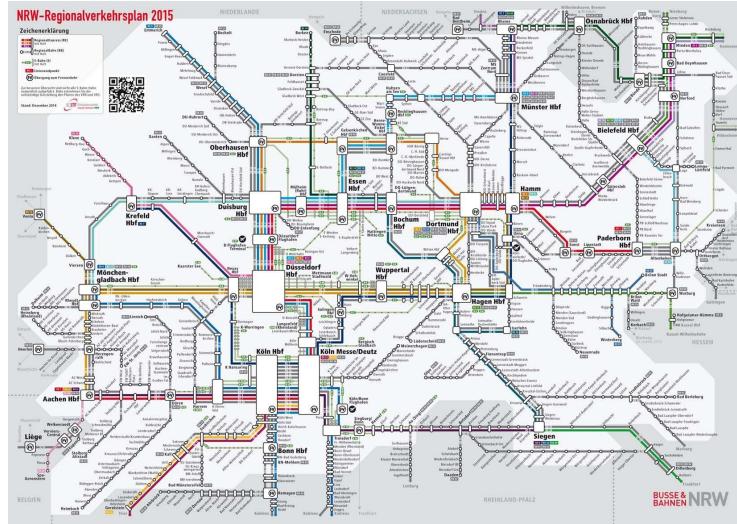
Motivation

- Minard (1861): Map of Napoleon's campaign in Russia (1812/13) "the best statistical drawing ever made..." (Tufts) army strength
- Casualties
- Troop movements
- Temperature during the retreat conditions



Today visualizations are everywhere

- Traffic
- Stock market
- Company
- Gaming
- ...

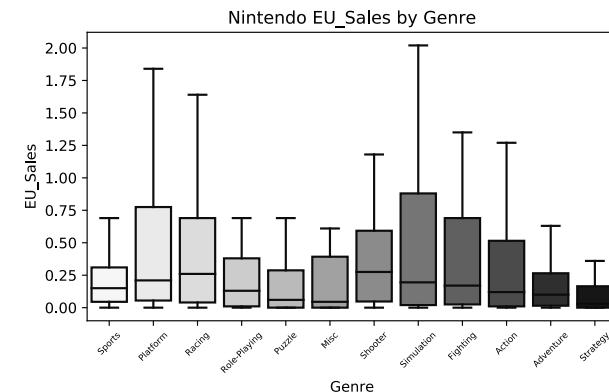


Data/ Information Visualization

- *The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.*
[Card et al 1999]
- *Is the communication of abstract data through the use of interactive visual interfaces.*
[Keim et al 2006]

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	
0	1	Wii Sports	Wii	2006.00000	Sports	Nintendo	41.49000	29.02000	3.77000	8.46000	82.74000
1	2	Super Mario Bros.	NES	1985.00000	Platform	Nintendo	29.08000	3.58000	6.81000	0.77000	40.24000
2	3	Mario Kart Wii	Wii	2008.00000	Racing	Nintendo	15.85000	12.88000	3.79000	3.31000	35.82000
3	4	Wii Sports Resort	Wii	2009.00000	Sports	Nintendo	15.75000	11.01000	3.28000	2.96000	33.00000
4	5	Pokemon Red/Pokemon Blue	GB	1996.00000	Role-Playing	Nintendo	11.27000	8.89000	10.22000	1.00000	31.37000
5	6	Tetris	GB	1989.00000	Puzzle	Nintendo	23.20000	2.26000	4.22000	0.58000	30.26000
6	7	New Super Mario Bros.	DS	2006.00000	Platform	Nintendo	11.38000	9.23000	6.50000	2.90000	30.01000
7	8	Wii Play	Wii	2006.00000	Misc	Nintendo	14.03000	9.20000	2.93000	2.85000	29.02000
8	9	New Super Mario Bros. Wii	Wii	2009.00000	Platform	Nintendo	14.59000	7.06000	4.70000	2.26000	28.62000
9	10	Duck Hunt	NES	1984.00000	Shooter	Nintendo	26.93000	0.63000	0.28000	0.47000	28.31000
10	11	Nintendogs	DS	2005.00000	Simulation	Nintendo	9.07000	11.00000	1.93000	2.75000	24.76000
11	12	Mario Kart DS	DS	2005.00000	Racing	Nintendo	9.81000	7.57000	4.13000	1.92000	23.42000
12	13	Pokemon Gold/Pokemon Silv	GB	1999.00000	Role-Playing	Nintendo	9.00000	6.18000	7.20000	0.71000	23.10000
13	14	Wii Fit	Wii	2007.00000	Sports	Nintendo	8.94000	8.03000	3.60000	2.15000	22.72000
14	15	Wii Fit Plus	Wii	2009.00000	Sports	Nintendo	9.09000	8.59000	2.53000	1.79000	22.00000

...



From Data to Visualization

V_Time	V_Name	V_Size
2019-05-01	Julius	1.85
2019-05-02	Laura	1.55
2019-05-05	Benny	1.87
...		



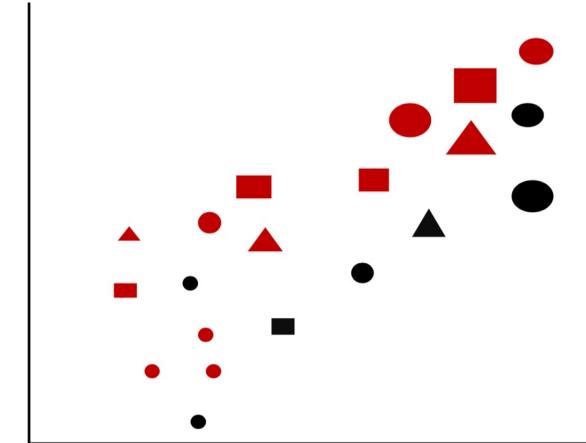
Basic Types of Variables for Visualizations (statistics)

- **Categorical variable** - Disordered set, e.g. names {Ben, Max, Laura} Only defined relation:
 - Equality relation (=)
 - String
- **Ordinal variable** - have natural, ordered categories and the distances between the categories is not known, e.g. Ranking {1,2,3...}
 - Defined order <.
 - Relationen: =, >, <
- **Numerical Variable** - Numeric range, e.g body size [1.85, 1.55, 1.78]
 - Arithmetic operations possible
 - Relationen: =, >, <, und Arithmetische Operationen
 - Discrete variable (Integer)
 - Continuous variable (Float)

Visualization and the Question of right Mapping?

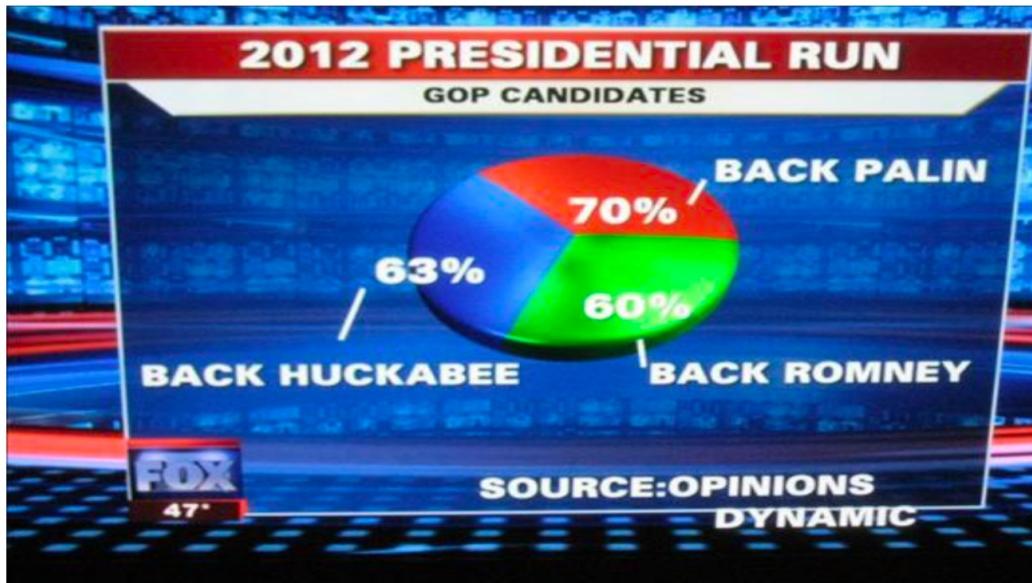
Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of humanity.

Bertin's Original Visual Variables	
Position changes in the x, y location	
Size change in length, area or repetition	
Shape infinite number of shapes	
Value changes from light to dark	
Colour changes in hue at a given value	
Orientation changes in alignment	
Texture variation in 'grain'	



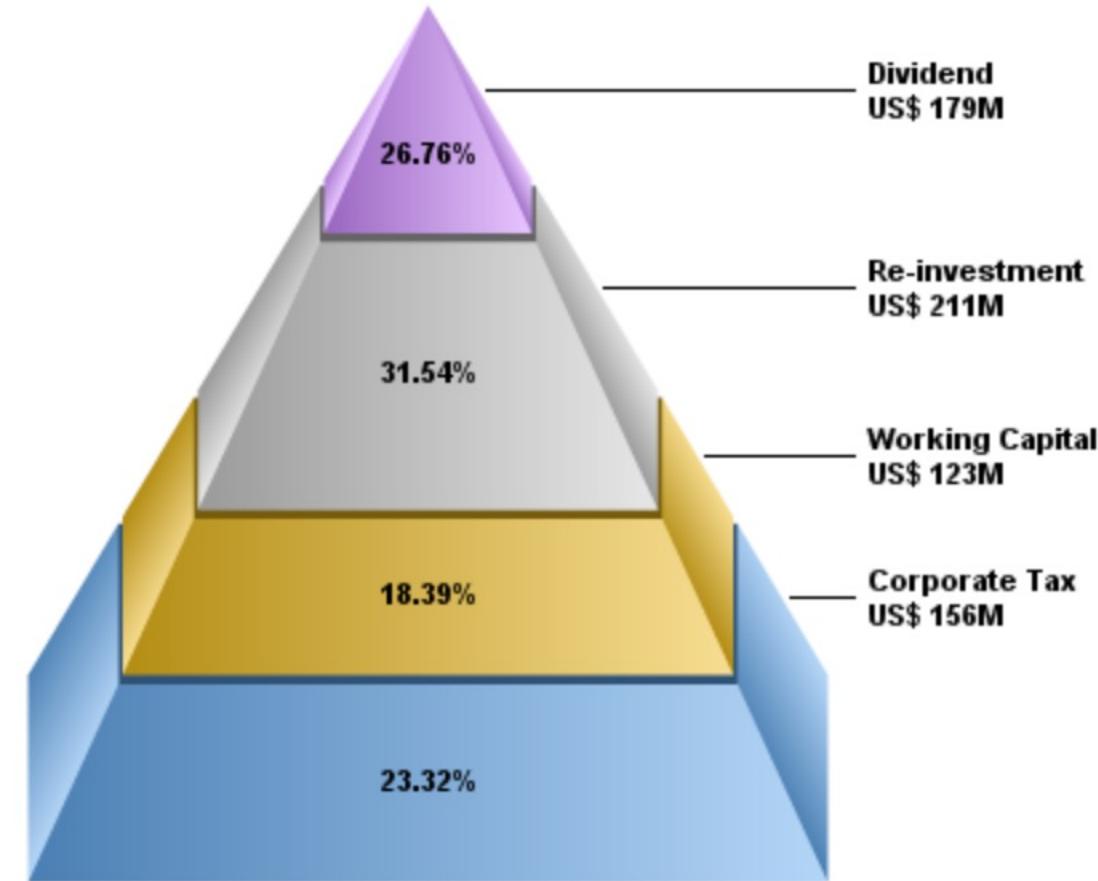
Good Visualizations?

- Let's get some ideas together...



Fox News 2012 Presidential Run

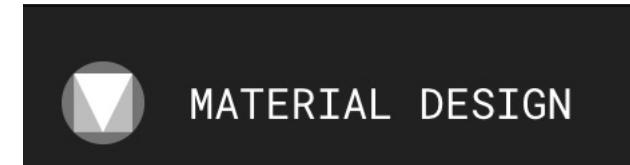
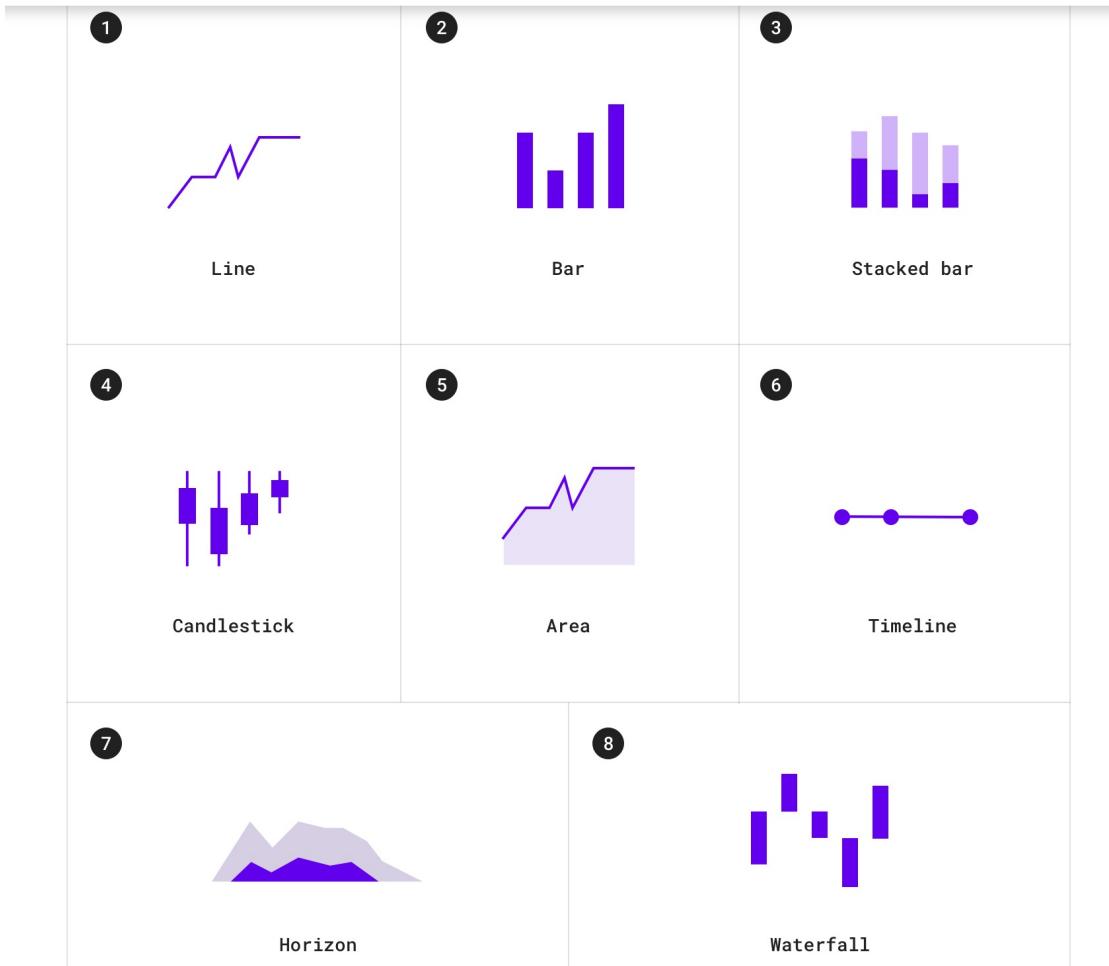
Quelle: wonkette.com



http://www.advsofteng.com/gallery_pyramid.html

Data Visualization Style Guidelines

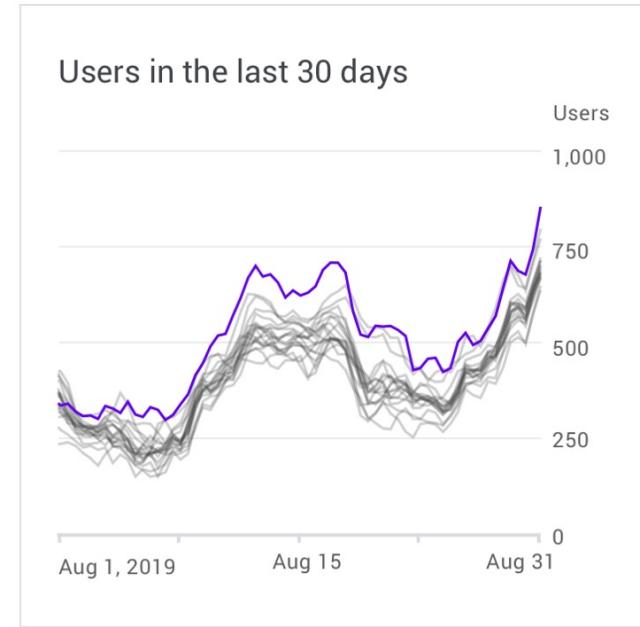
Communication > Data visualization > Types



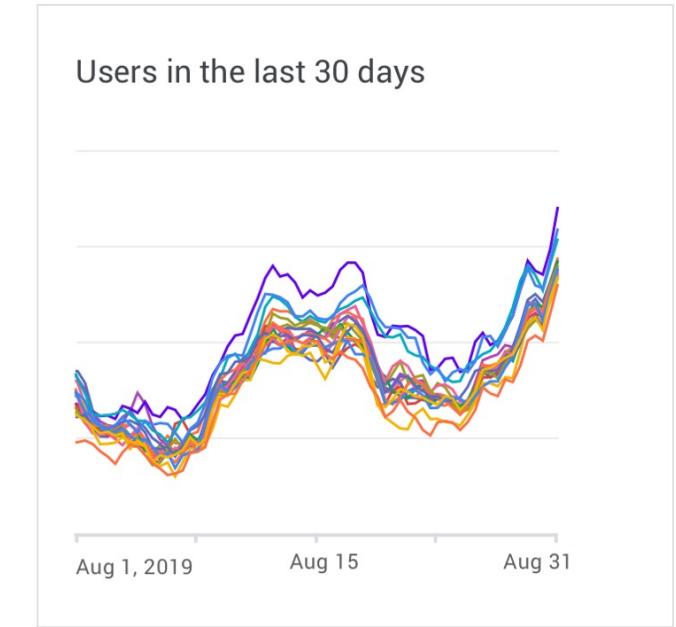
- <https://material.io/design/communication/data-visualization.html#>

Right Guidelines for good visualizations

- Title
- Labels on axis
- Keep it simple
- Only a few colours

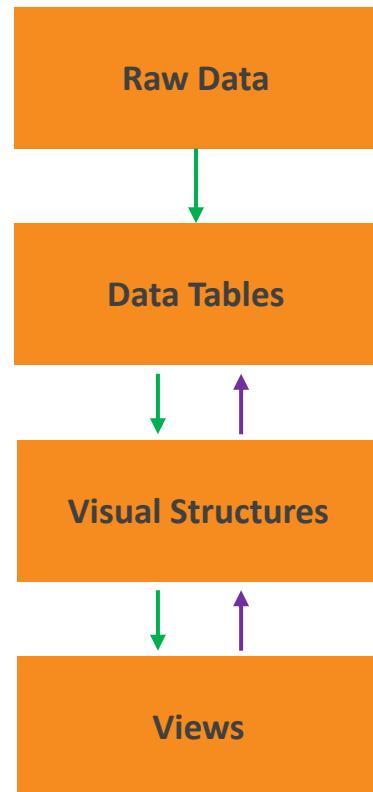


Do



Don't

Visualization Process - The Baseline



What is the **goal** of visualization?
What is the point of interest?
What do you want to **understand**?

How is the data **sampled**?
Which data is **relevant**?

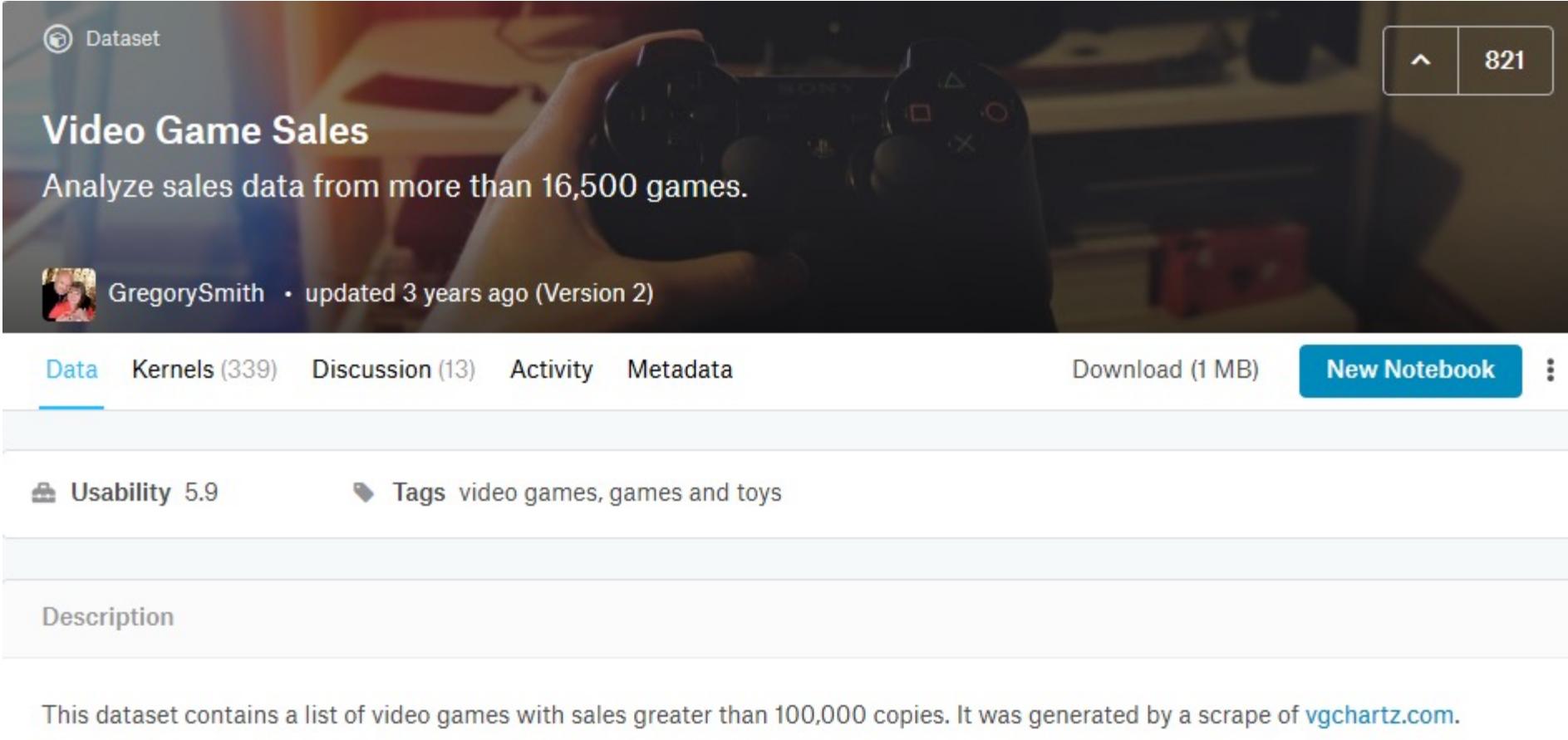
Plot the data.
Are there **anomalies**?
Are there **patterns**?

Build a Visualization.
Do the results make **sense**?

Referenzmodell von Card et al. (1999)

Using large Datasets for Data Analysis

- For this cause, we utilize a dataset on video game sales from kaggle.com



The screenshot shows a dataset page on Kaggle. At the top, it says "Dataset" and "Video Game Sales". Below that, it says "Analyze sales data from more than 16,500 games." A user profile picture for "GregorySmith" is shown, followed by the text "updated 3 years ago (Version 2)". Below the title, there are tabs for "Data" (which is selected), "Kernels (339)", "Discussion (13)", "Activity", and "Metadata". To the right of these tabs are buttons for "Download (1 MB)" and "New Notebook". There is also a three-dot menu icon. Below the tabs, there are sections for "Usability 5.9" and "Tags video games, games and toys". A "Description" section follows, containing the text: "This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of [vgchartz.com](#)".

- Do not worry about the scraping part. For now, the dataset is provided for you in your workspace.

Import the data

- The data frame object has some powerful operations
- This makes our life easier to reach the goals

```
# Now create a pandas dataframe.  
df = pd.read_csv(...)  
df = df.dropna(...)
```

General Syntax

```
# Access to the library
import matplotlib.pyplot as plt

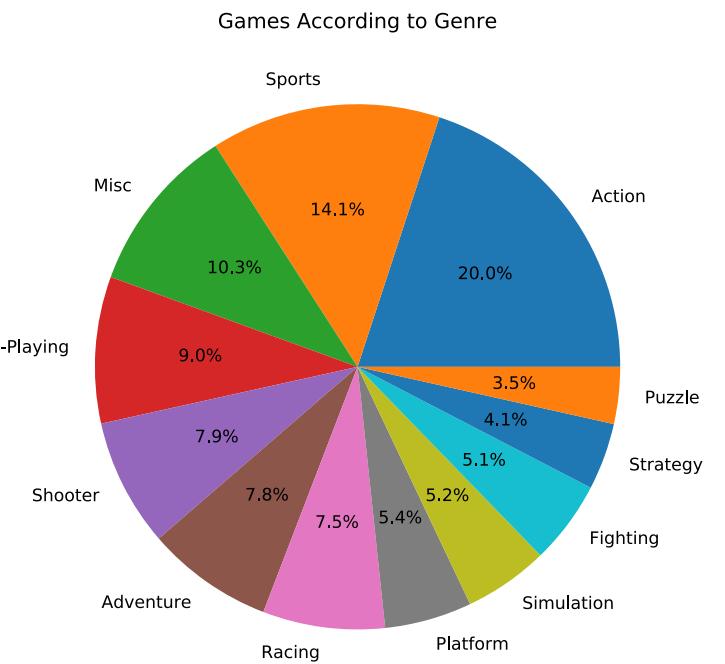
plt.figure()
plt.plotname(parameters)
plt. . .
plt.savefig()
plt.close()
```

Create a new figure
Choose plot
Additional adjustments
Save plot as .png or .pdf
Close plot

General Syntax for Data Access

```
Variable = dataframe[‘Column name’].method()  
  
Create a new variable  
Data frame object with column referencing  
Useful method
```

Pie Chart

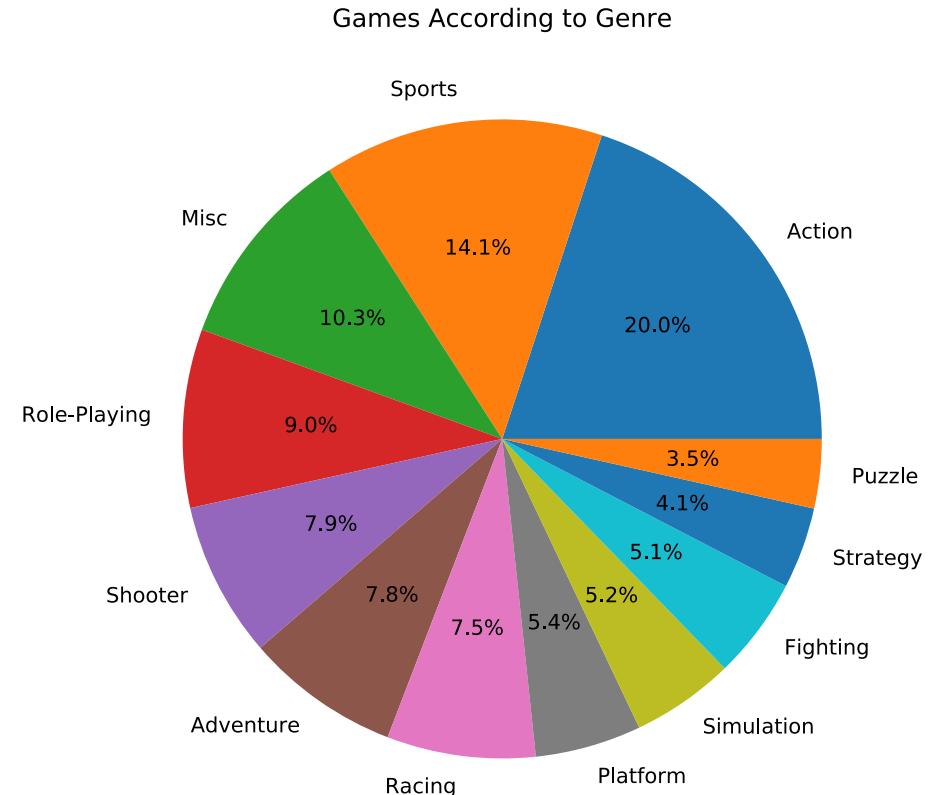


Games According to Genre

- Try to great a scatter plot games according to genre

Pie plot:

- Pie charts express portions of a whole, using arcs or angles within a circle.
- Make a pie chart of array x . The fractional area of each wedge is given by $x/\text{sum}(x)$. If $\text{sum}(x) < 1$, then the values of x give the fractional area directly.



Data Access for Pie Plot

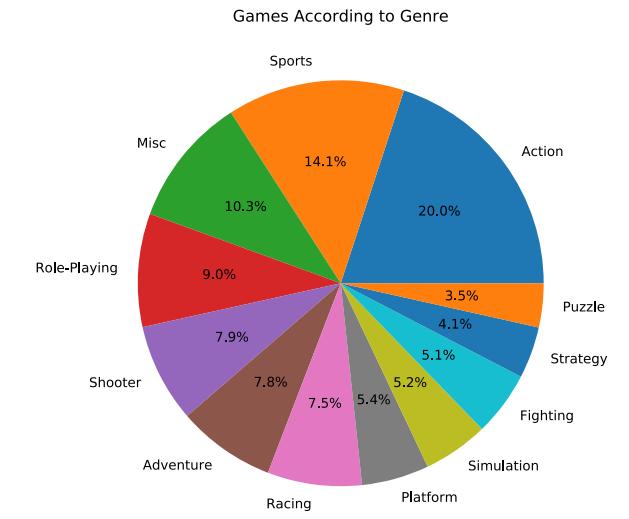
Return a series containing counts of unique values

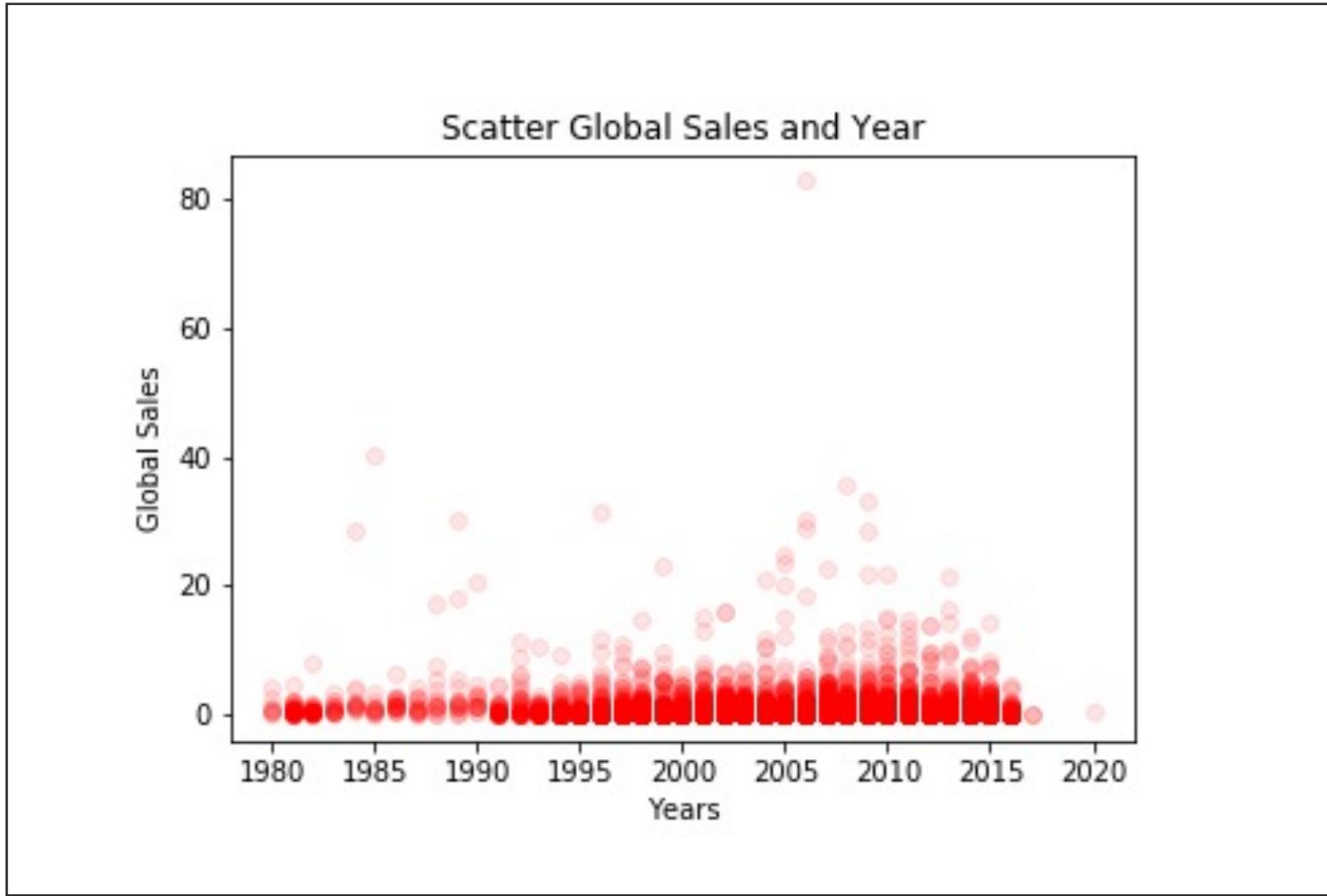
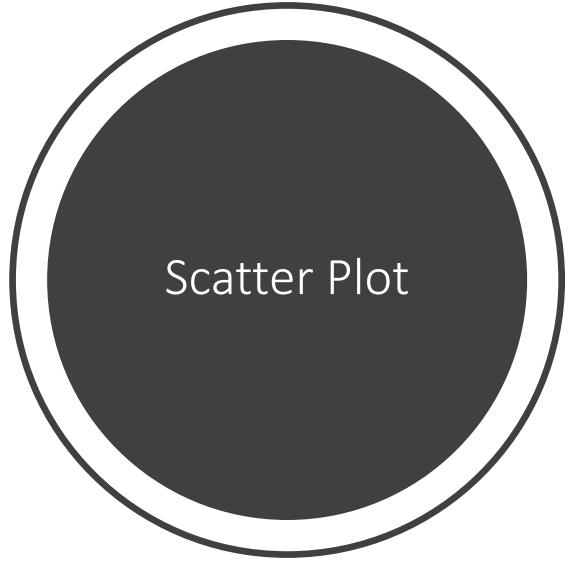
```
LSizes = df['Genre'].value_counts()  
LLabels = df['Genre'].value_counts().index
```

Return unique index like labels

Solution: Scatter Plot

```
plt.figure(figsize = (7,7))
plt.pie(LSizes, labels=LLabels,
autopct='%.1f%%')
plt.title('Games According to Genre')
plt.savefig('PieGamesGenre.pdf')
```



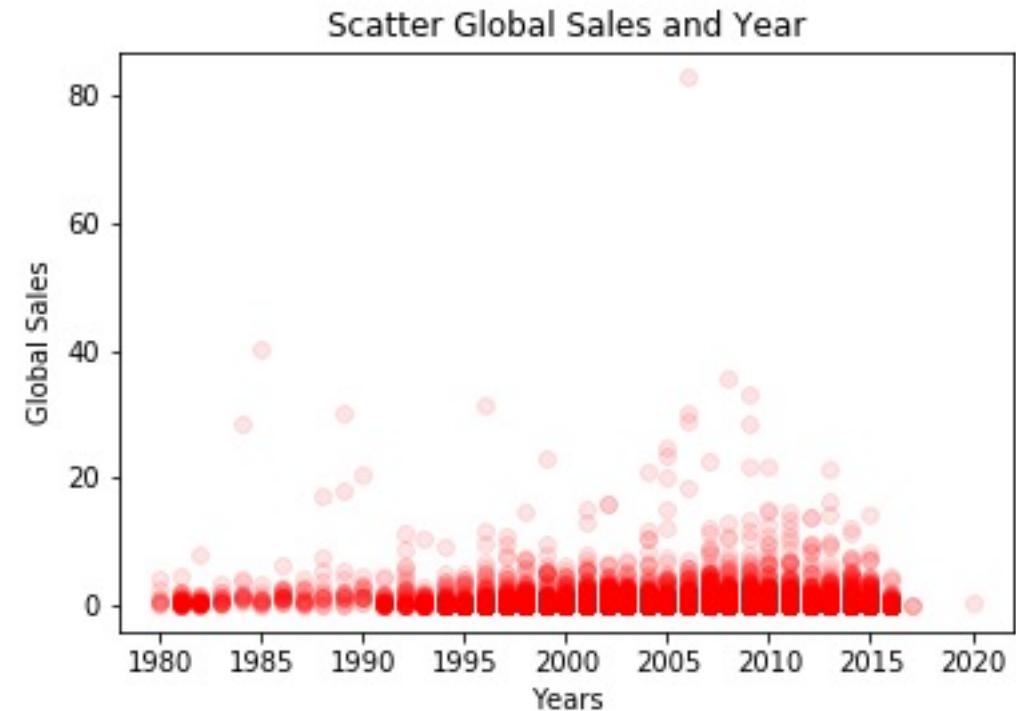


Video Games Global Sales by Years

- Try to great a scatter plot for “Global Sales” and “years”.

Scatter plot:

- Scatter plot is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for set of data.
- There are 2 variables for each object
- Variables mapped to position
- Scatterplot works well to recognize patterns (e.g. correlations of two variables, groups)
- Problems: Must be well designed to avoid problems “Overplotting”



Data Access for Scatter

```
LGlobal_Sales = df['Global_Sales'].values  
LYears = df['Year'].values()
```

Matplotlib Options

- Matplotlib plots has numerous options.
- See documentation is a good starting point
- https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.scatter.html
- Plots are functions with function parameters

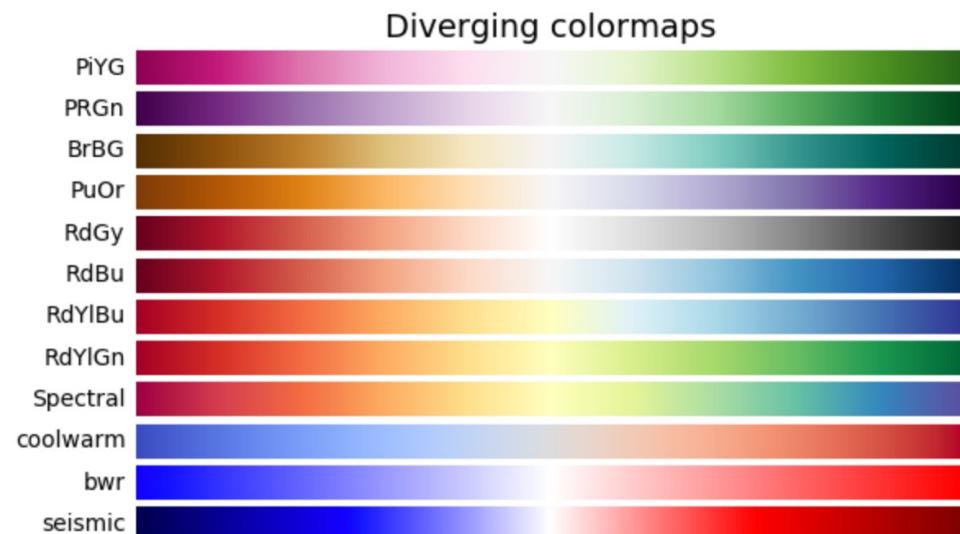
```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None,  
    cmap=None, norm=None, vmin=None, vmax=None, alpha=None, lin  
    ewidths=None, verts=None, edgecolors=None, *, plotnonfinite=  
    False, data=None, **kwargs) [source]
```

Matplotlib – Chance the colours

```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None,  
cmap=None, norm=None, vmin=None, vmax=None, alpha=None, lin  
ewidths=None, verts=None, edgecolors=None, *, plotnonfinite=  
False, data=None, **kwargs) [source]
```

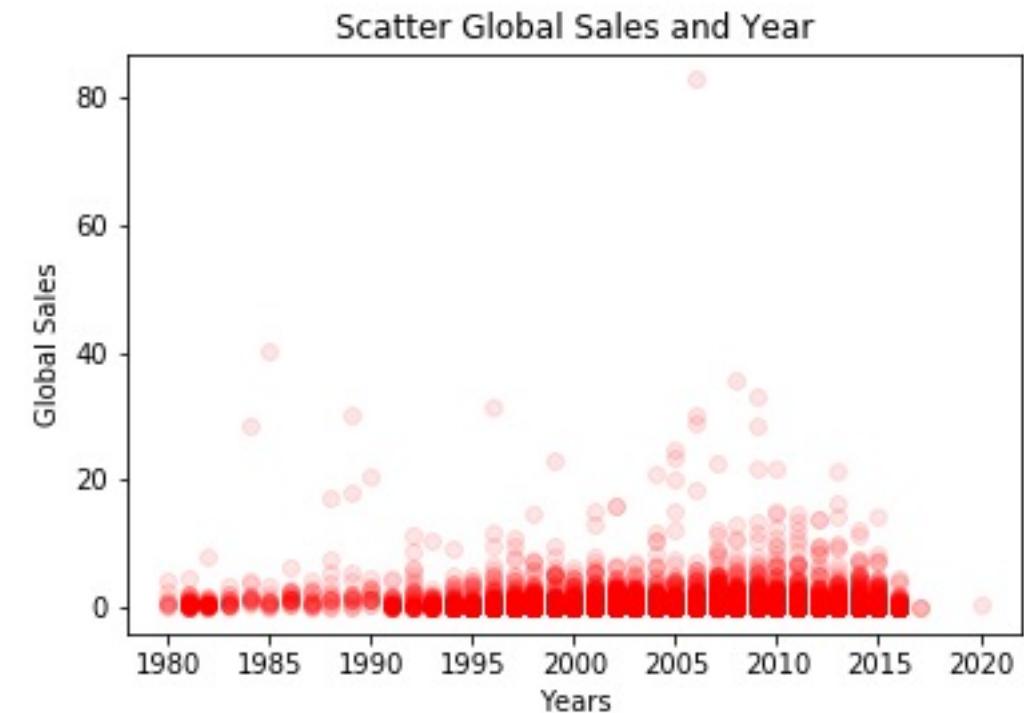
There are hundreds of colours options:

<https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html>



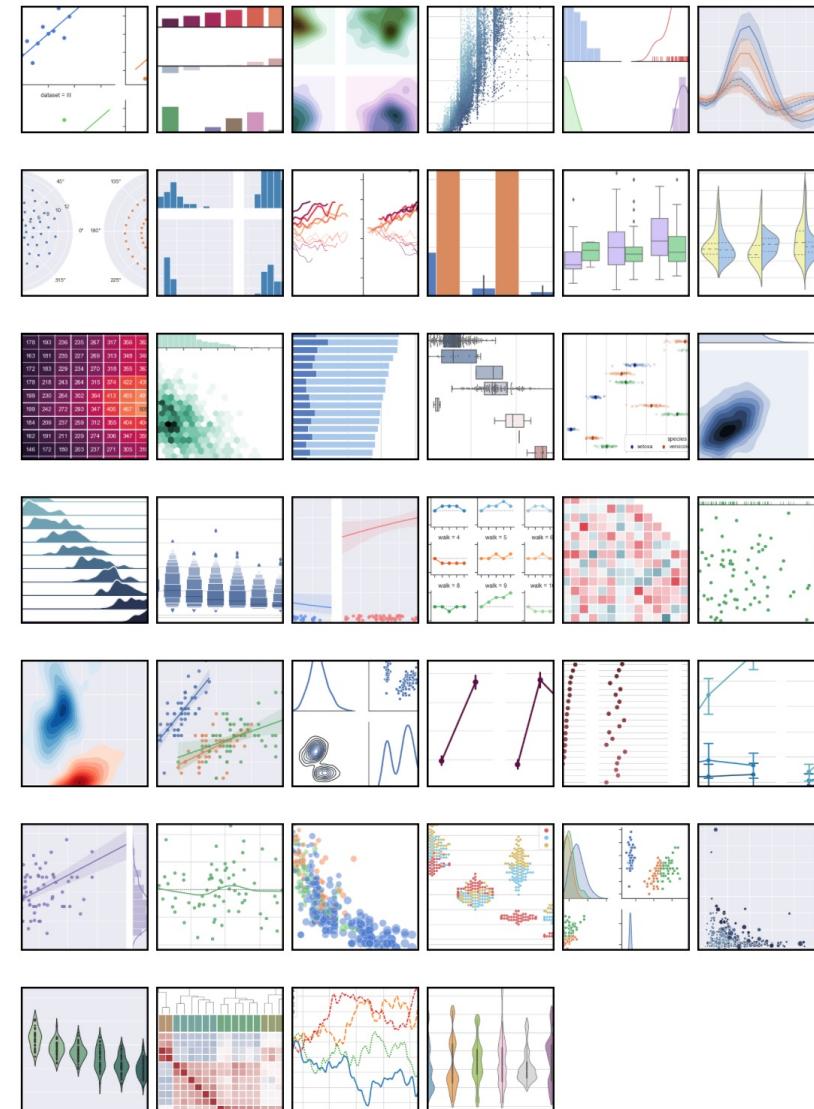
Solution: Scatter Plot

```
plt.figure()
plt.scatter(LYears,
LGlobal_Sales, color='red',
alpha=0.1)
plt.title('Scatter Global Sales
and Year')
plt.ylabel('Global Sales')
plt.xlabel('Years')
plt.savefig('ScatterUpdate.png')
plt.show()
```



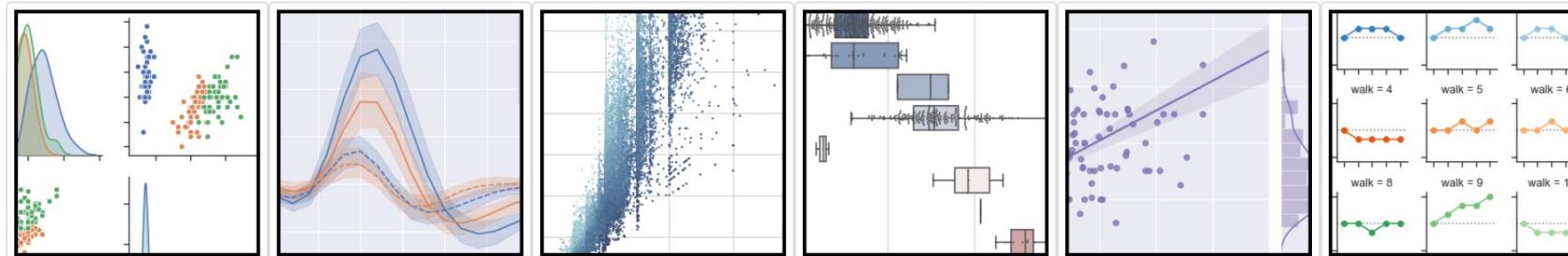
Seaborn

Example gallery



Combination of Power

- Dealing with matplotlib and Pandas data frame architecture can be awkward
- An extension of the matplotlib is Seaborn.
- This library can operate directly with data frame objects. For data science, a useful tool!
- seaborn: statistical data visualization
- Seaborn is a Python data visualization library based on matplotlib.
- It provides a high-level interface for drawing attractive and informative statistical graphics.

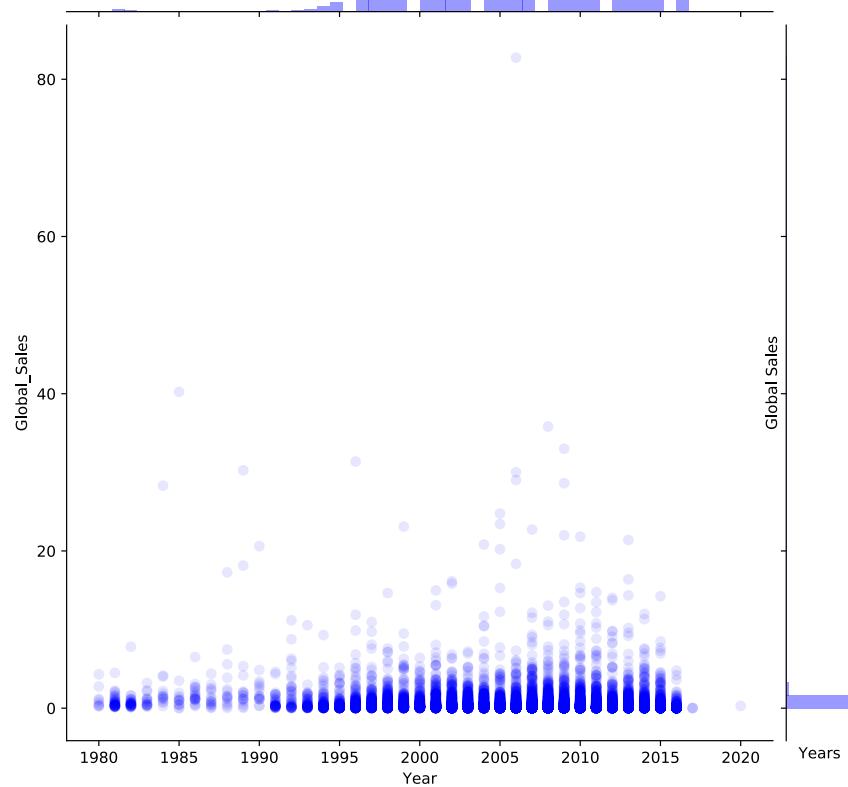


Joint Plot

- Try to create a Joint plot for “Global Sales” and “years”.

Scatter plot:

- Draw a scatterplot with marginal histograms



Access Data for Joint Plot

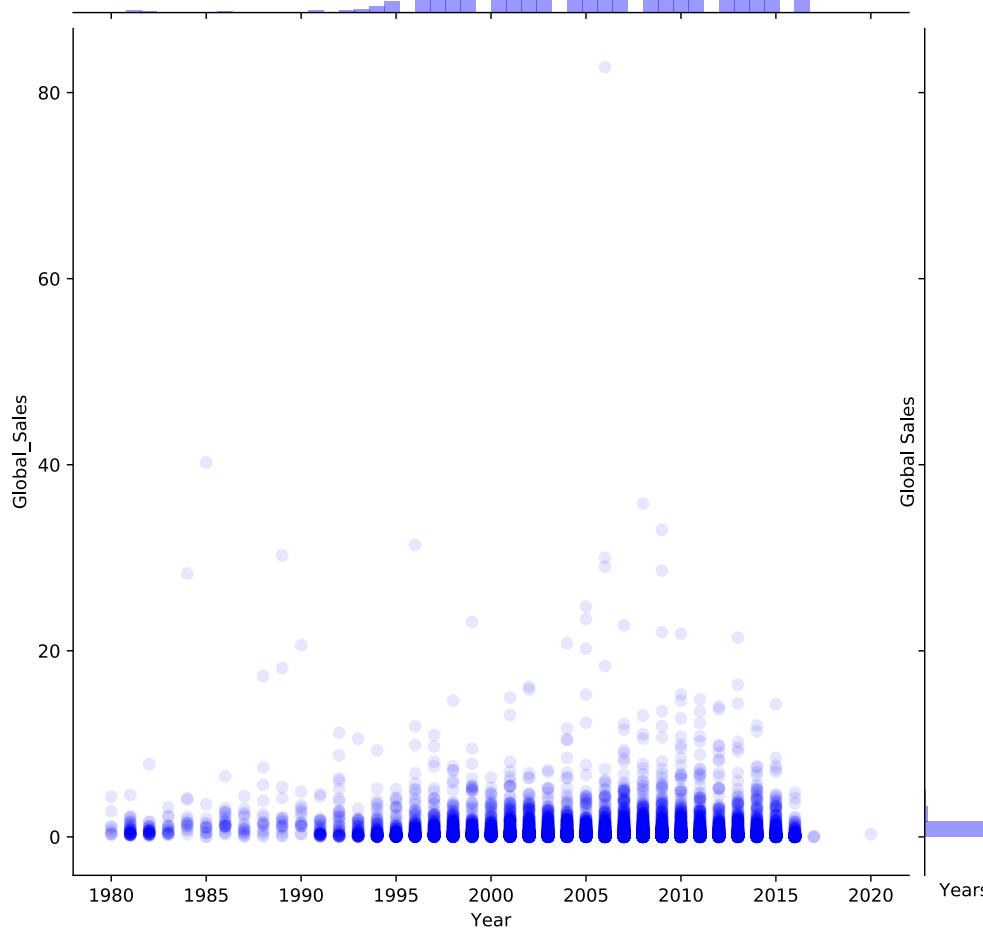
- The data frame object has some powerful operations

```
# Import the matplotlib library.  
import matplotlib.pyplot as plt  
  
# Extract the data for the plot, for example:  
LGlobal_Sales = df['Global_Sales'].get_values()  
LYears = df['Year'].get_values()
```

Solution: Joint Plot

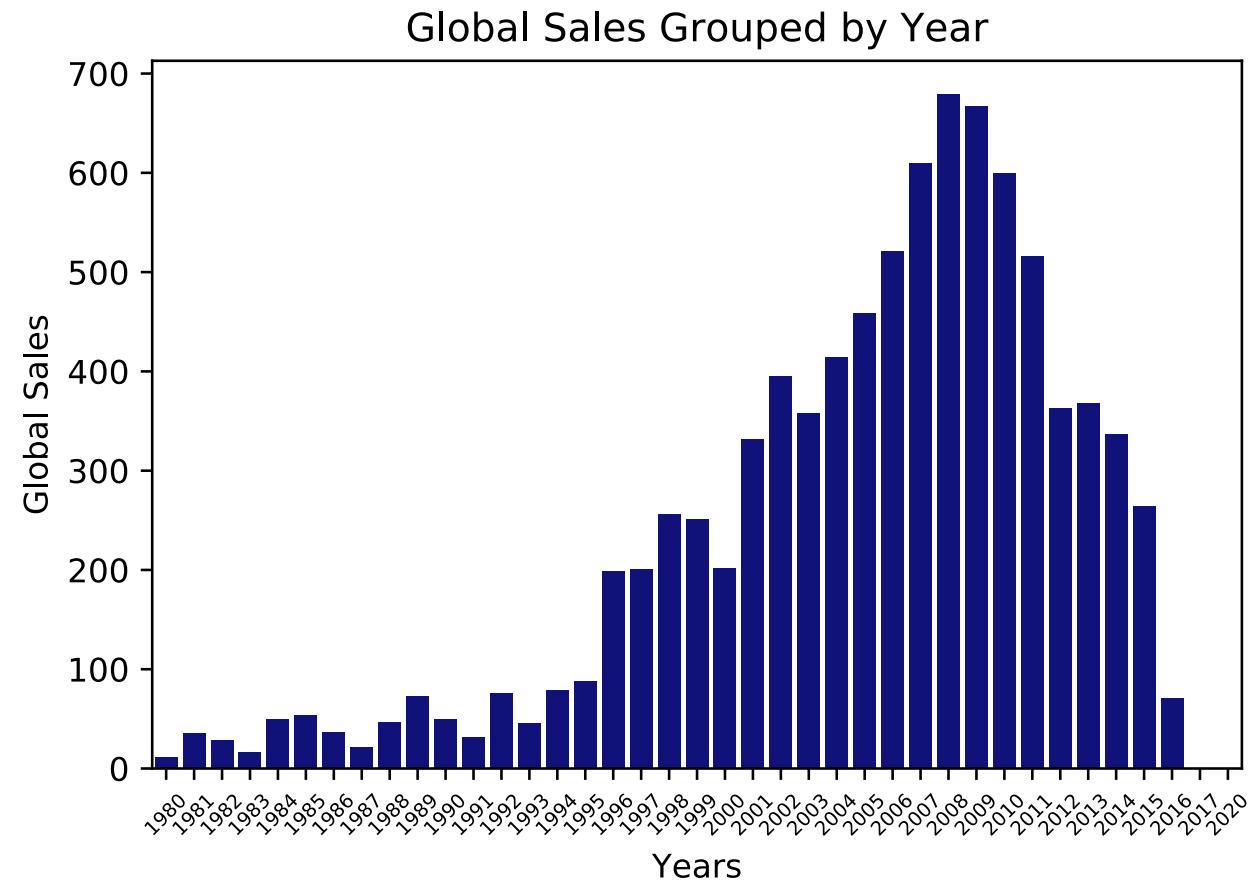
```
# For using seaborn we need to use the bib.  
import seaborn as sns  
  
# Extended  
plt.figure()  
sns.jointplot(df.Year, df.Global_Sales, size=8, ratio=9,  
color="blue", alpha=0.1)  
# plt.title('Global Sales Over the Years')  
plt.ylabel('Global Sales')  
plt.xlabel('Years')  
plt.savefig('JoinPlot.pdf')  
plt.tight_layout()  
plt.show()
```

More than One Plot



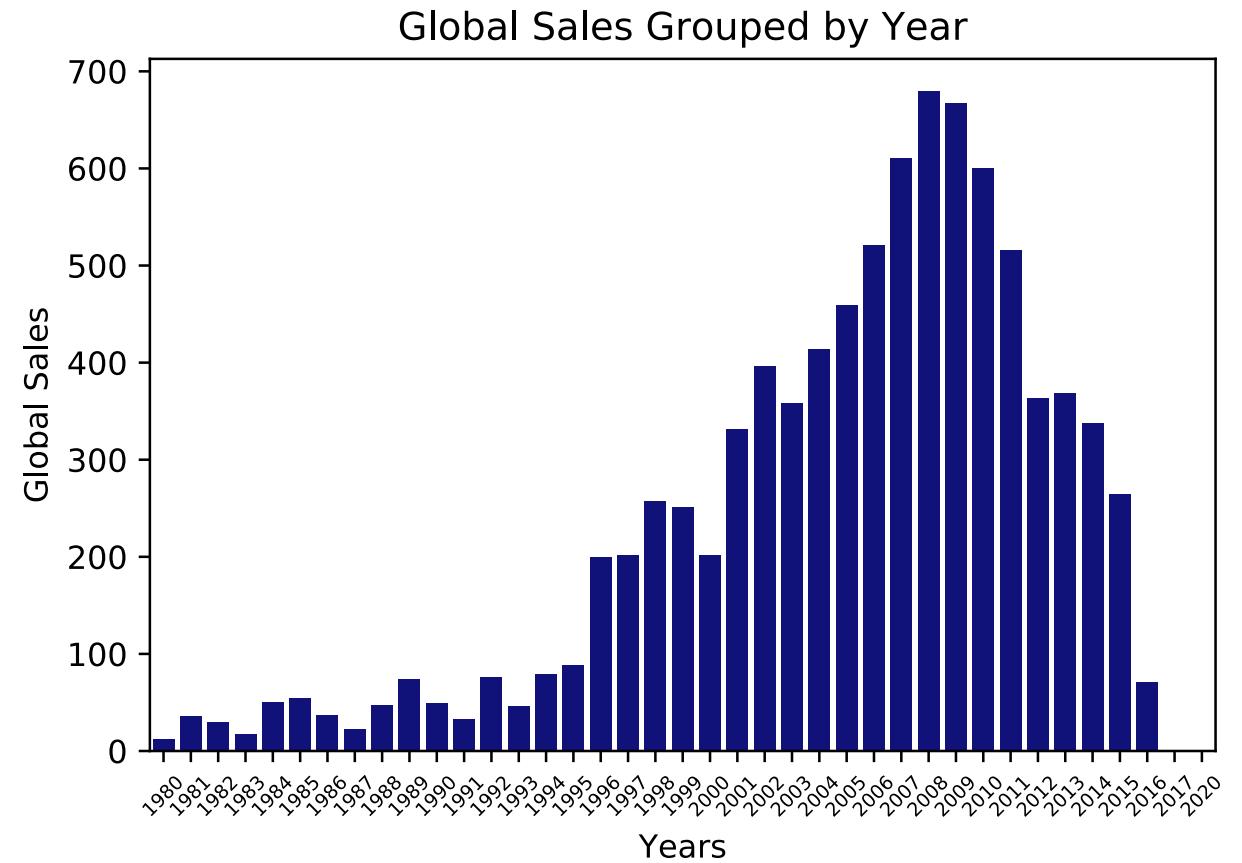
<http://seaborn.pydata.org/generated/seaborn.jointplot.html?highlight=join%20plot>

Bar Chart



Bar Chart

- Simple Chart
- Mapping value to position
- A bar plot is a plot that presents categorical data with rectangular bars
- Problems: Data with high range! Sorted vs. unsorted



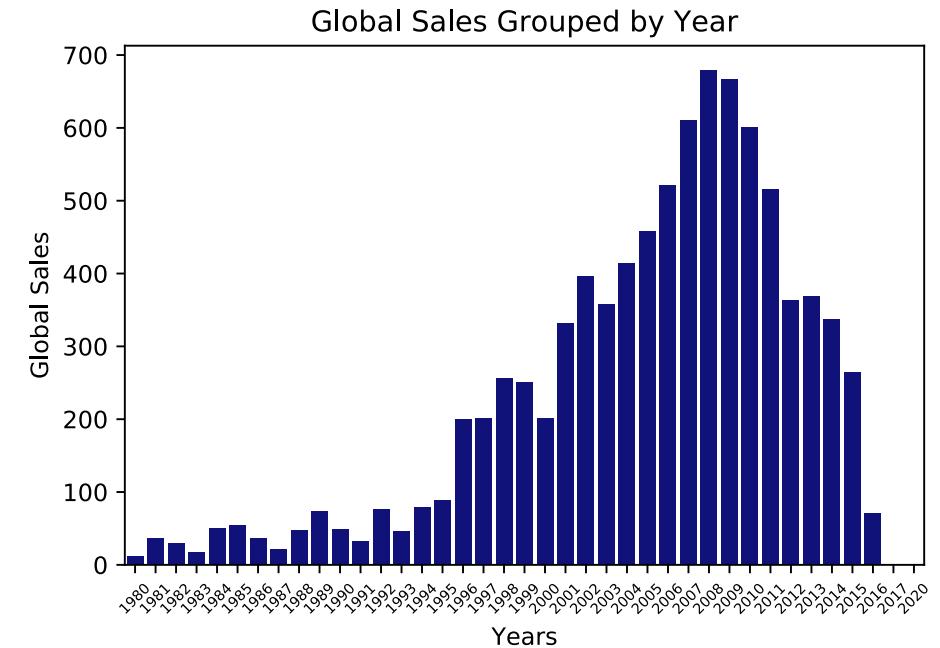
Access Data for Bar Char

- The data frame object has some powerful operations

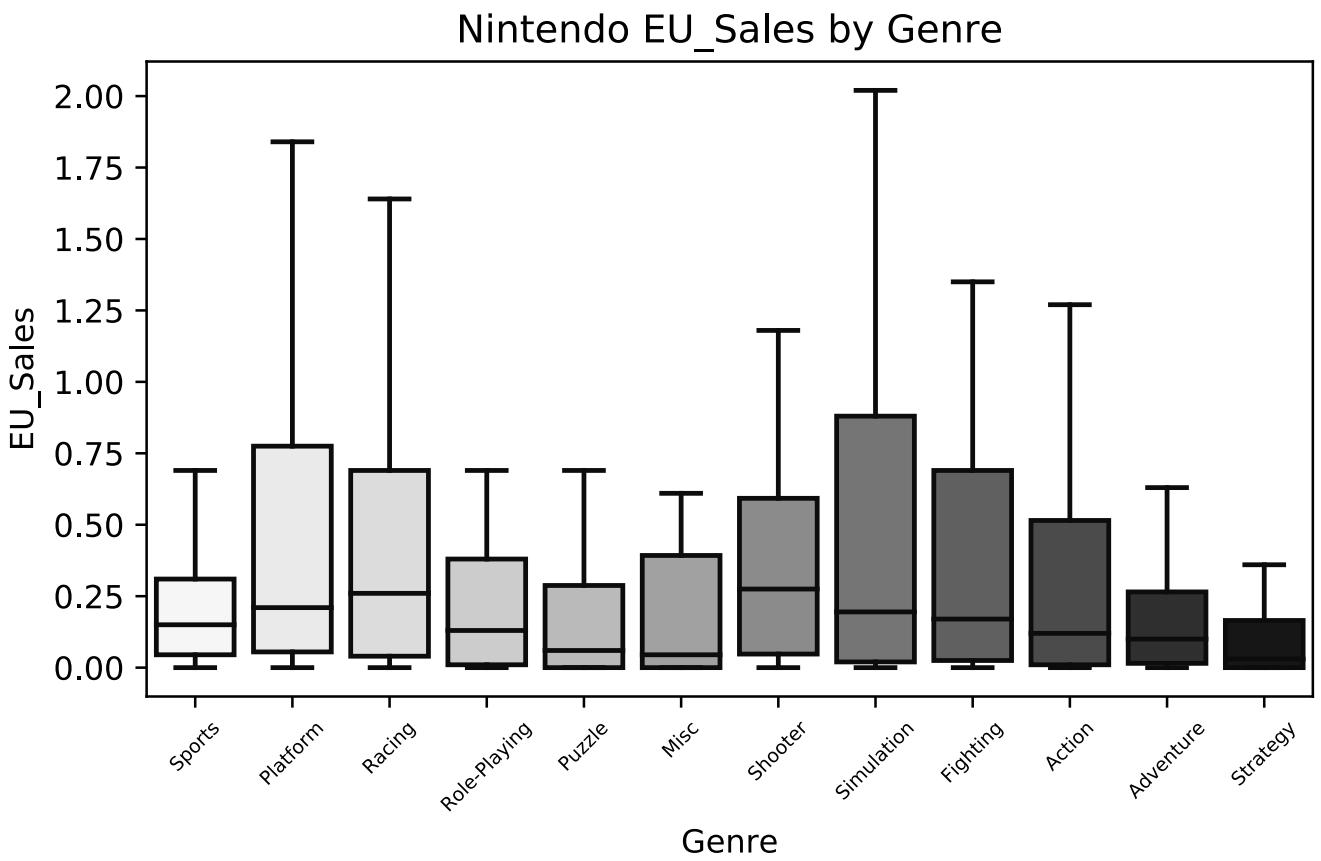
```
# Access data
# In our case sum data for each year.
# @code: df_groupData Grouped data by years and summed up.
# @code: LIndexesOfGroupData years converted as int!
df_groupData = df.groupby(['Year']).sum()
LdfGroupSales = df_groupData['Global_Sales']
LIndexesOfGroupData = df_groupData.index.astype(int)
```

Solution: Bar Plot

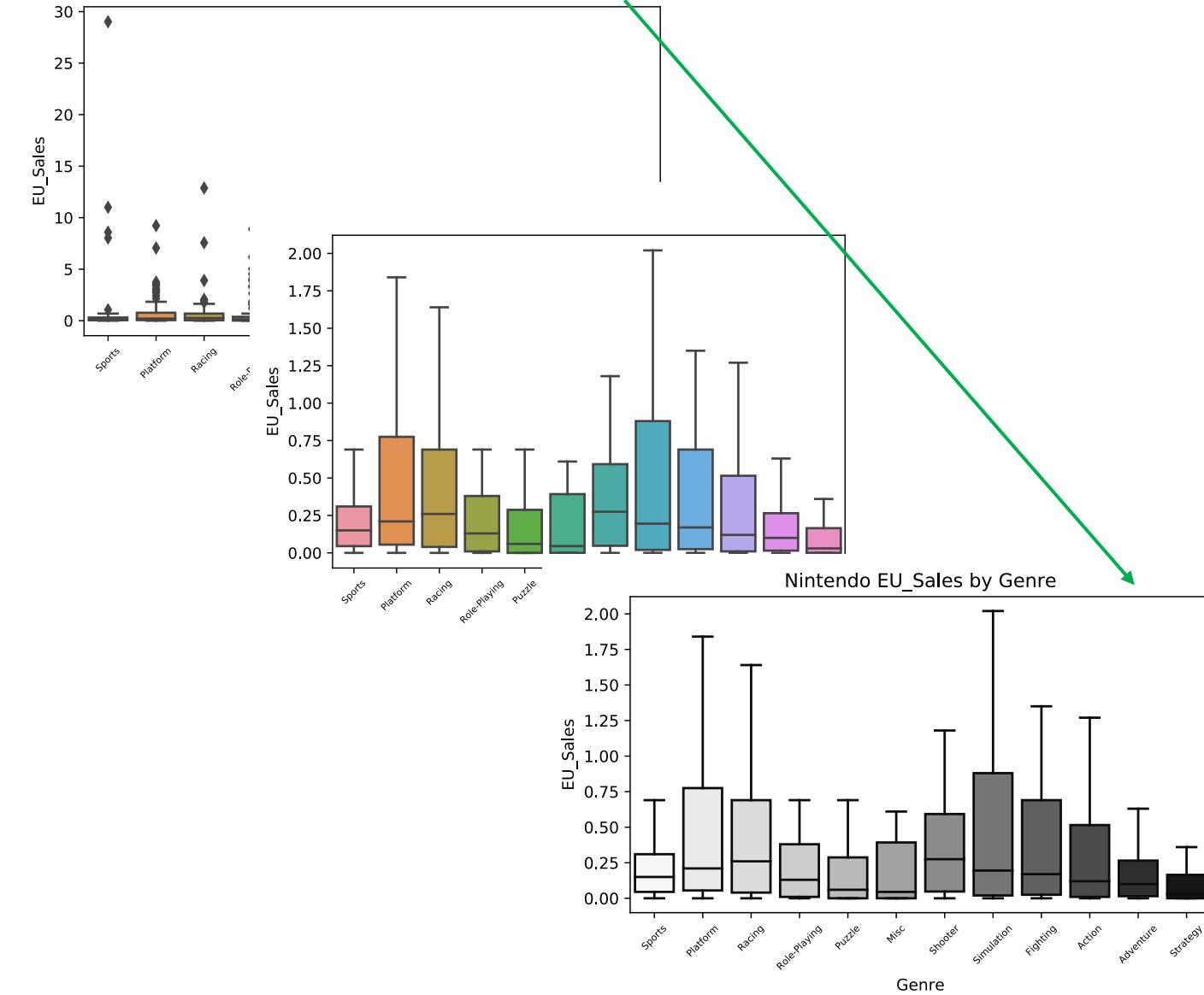
```
plt.figure()  
sns.barplot(y=LdfGroupSales,  
x=LIIndexesOfGroupData, color='darkblue')  
plt.title('Global Sales Grouped by Year')  
plt.xticks(rotation=45, fontsize=6)  
plt.ylabel('Global Sales')  
plt.xlabel('Years')  
plt.savefig('BarChart.pdf')  
plt.show()
```



Box Plot



Visualization Process - A deeper Look



What is the **goal** of visualization?
What is the point of interest?
What do you want to
understand?

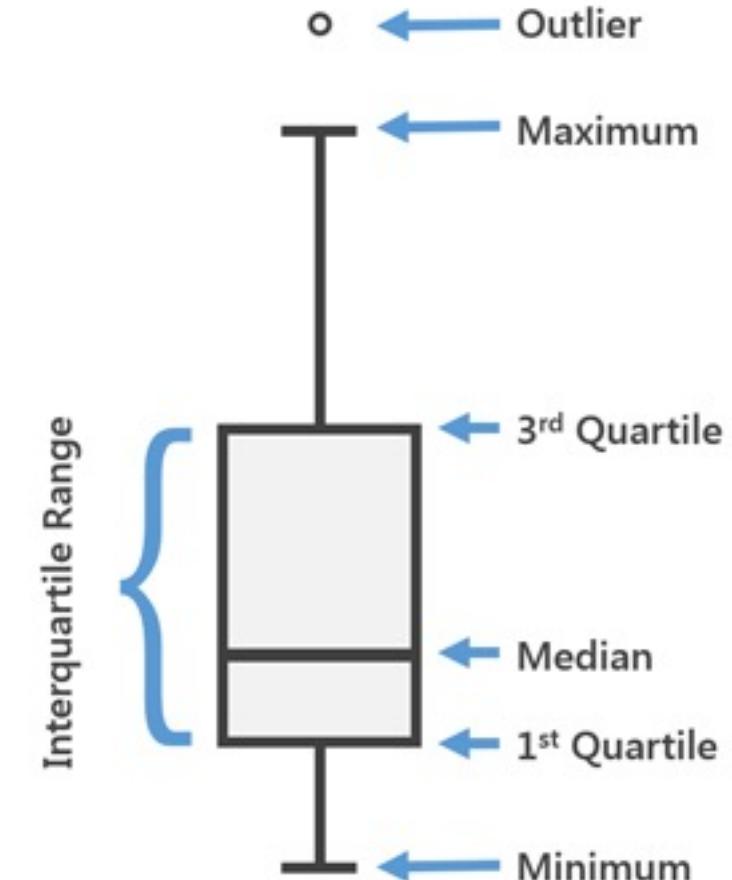
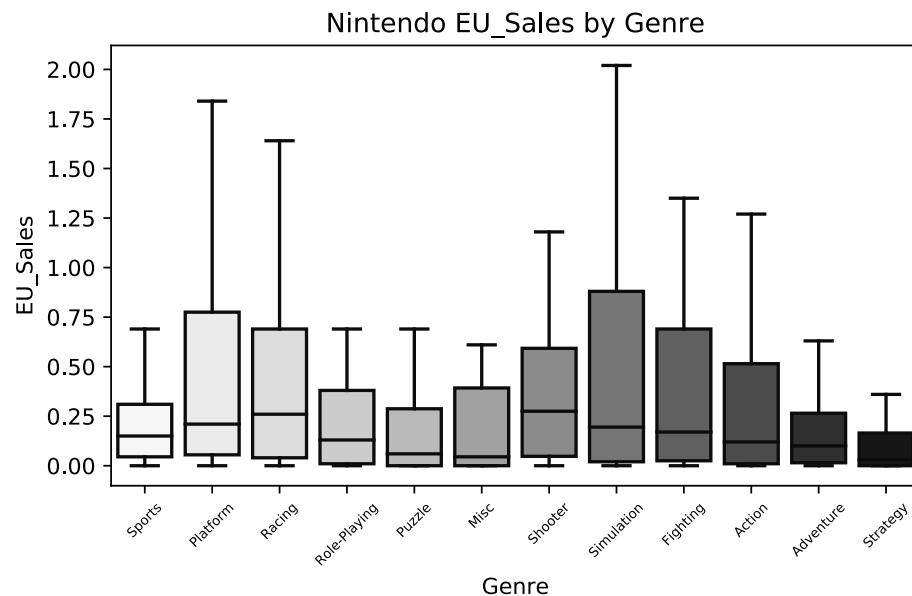
How is the data **sampled**?
Which data is **relevant**?

Plot the data.
Are there **anomalies**?
Are there **patterns**?

Build a Visualization.
Do the results make **sense**?

Box Plot

- Visualizes statistical properties
- Uses different forms for mapping
- Problems: Numbers and range of outliers



https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwicyf_y0vPkAhVQYVAKHSSWAQQjRx6BAgBEAQ&url=https%3A%2F%2Fpro.arcgis.com%2Fde%2Fopro-app%2Fhelp%2Fanalysis%2Fgeoprocessing%2Fcharts%2Fbox-plot.htm&psig=AOfVaw3Q-zP6jy0RYzKLHM60kVF&ust=1569764533344246

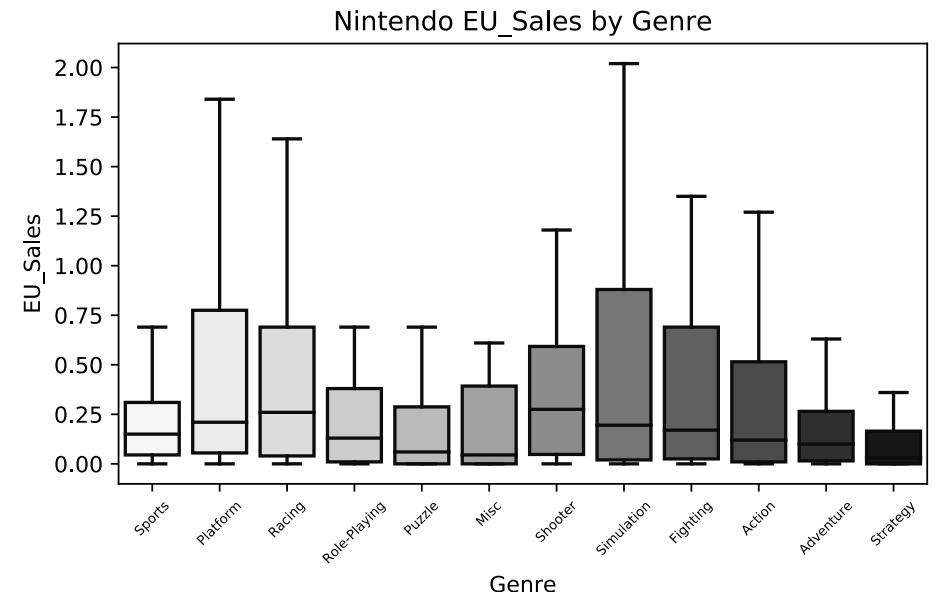
Access Data for Bar Char

- The data frame object has some powerful operations

```
dfNintendo = df[df.Publisher == 'Nintendo']
```

Solution: Bar Plot

```
plt.figure()  
sns.boxplot(x='Genre', y='EU_Sales',  
data=dfNintendo, showfliers=False,  
            palette='Greys')  
plt.title('Nintendo EU_Sales by Genre')  
plt.xticks(rotation=45, fontsize=6)  
plt.tight_layout()  
plt.savefig('BoxPlotUpdat3.pdf')  
plt.show()
```



More Information

Choosing color palettes:

https://seaborn.pydata.org/tutorial/color_palettes.html

Example gallery

<http://seaborn.pydata.org/examples/index.html>

Data Frame own Plots methods

pandas.DataFrame.plot

```
df.plot(x =df['A'], y= x =df['B'], kind = 'scatter')
```

```
df.plot(x =df['A'], y= x =df['B'], kind = 'line')
```

```
df.plot(x =df['A'], y= x =df['B'], kind = 'bar')
```

Visualization: Lessons learned

- There is a large number of visualizations.
- However, creating a good visualization is not a trivial undertaking.
- Concentration on the central message is important.

Kontakt



Dr. Benjamin M. Abdel-Karim

YouTube



<https://www.youtube.com/channel/UC51s9CqrZ466xw4vSizkHmw>

LinkedIn



<https://www.linkedin.com/in/benjamin-m-abdel-karim-b48948180/>

References

- <https://www.datacamp.com/community/blog/seaborn-cheat-sheet-python>
- <https://material.io/design/communication/data-visualization.html#>

Cheat Sheets:

- [https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas Cheat Sheet.pdf](https://github.com/pandas-dev/pandas/blob/master/doc/cheatsheet/Pandas_Cheat_Sheet.pdf)
- <https://drive.google.com/drive/folders/0BylrJAE4KMTtaGhRcXkxNHhmY2M>
- <https://python-graph-gallery.com/cheat-sheets/>