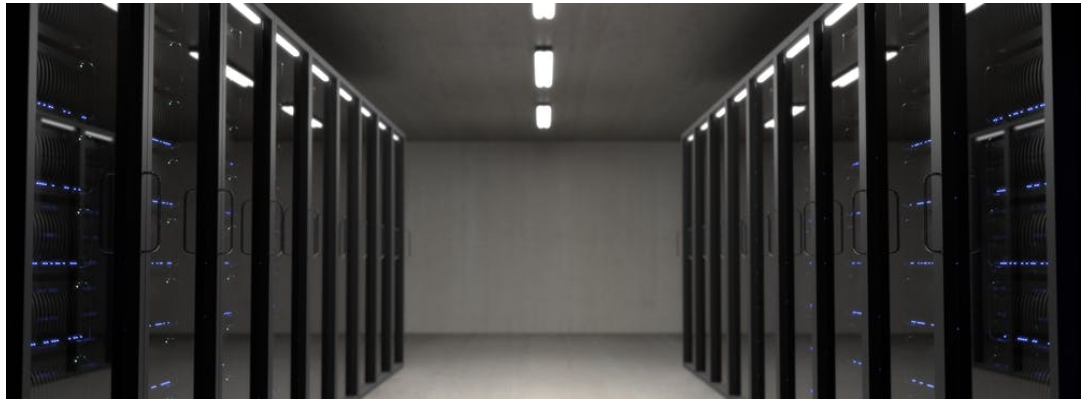
Data Visualization

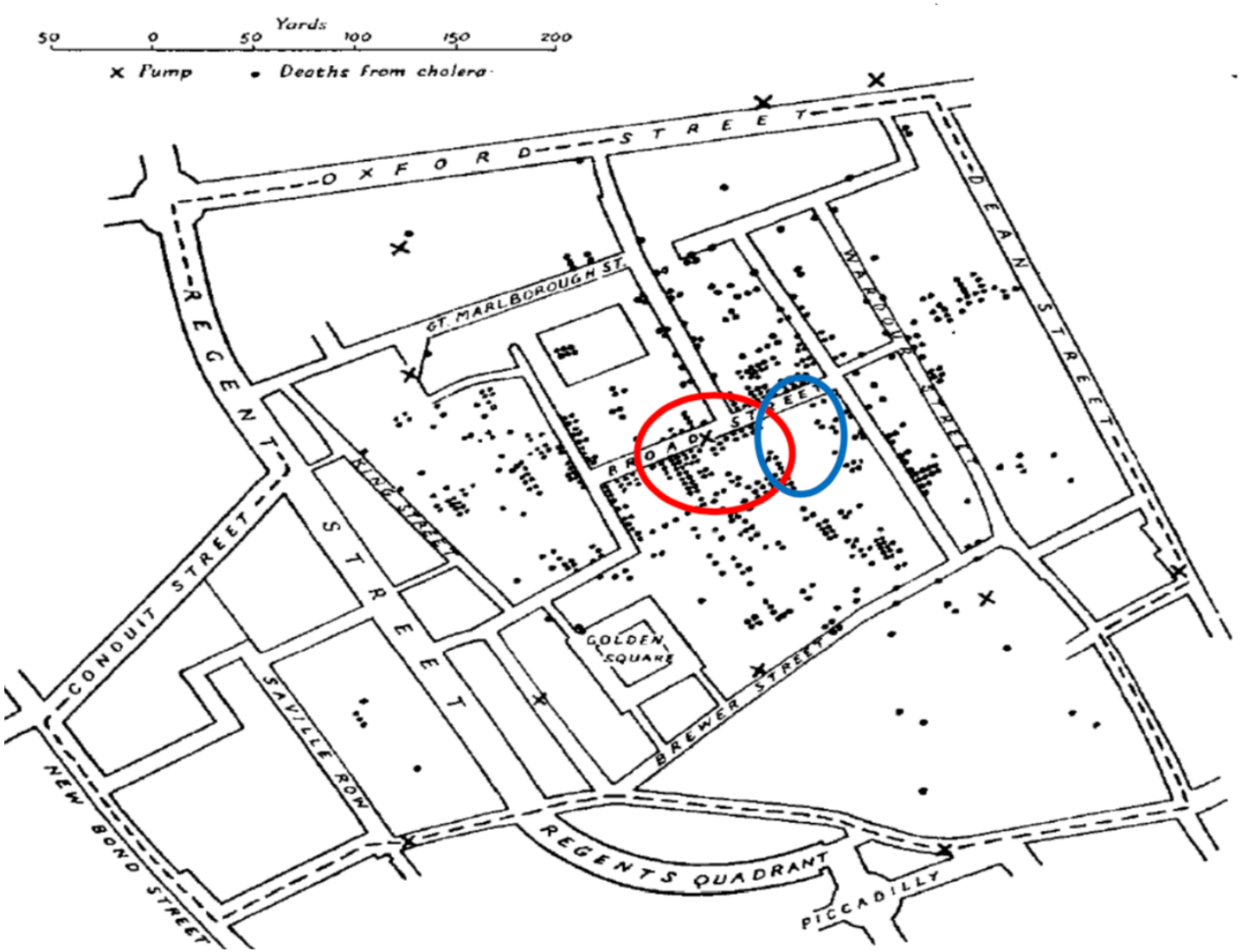
Motivation

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



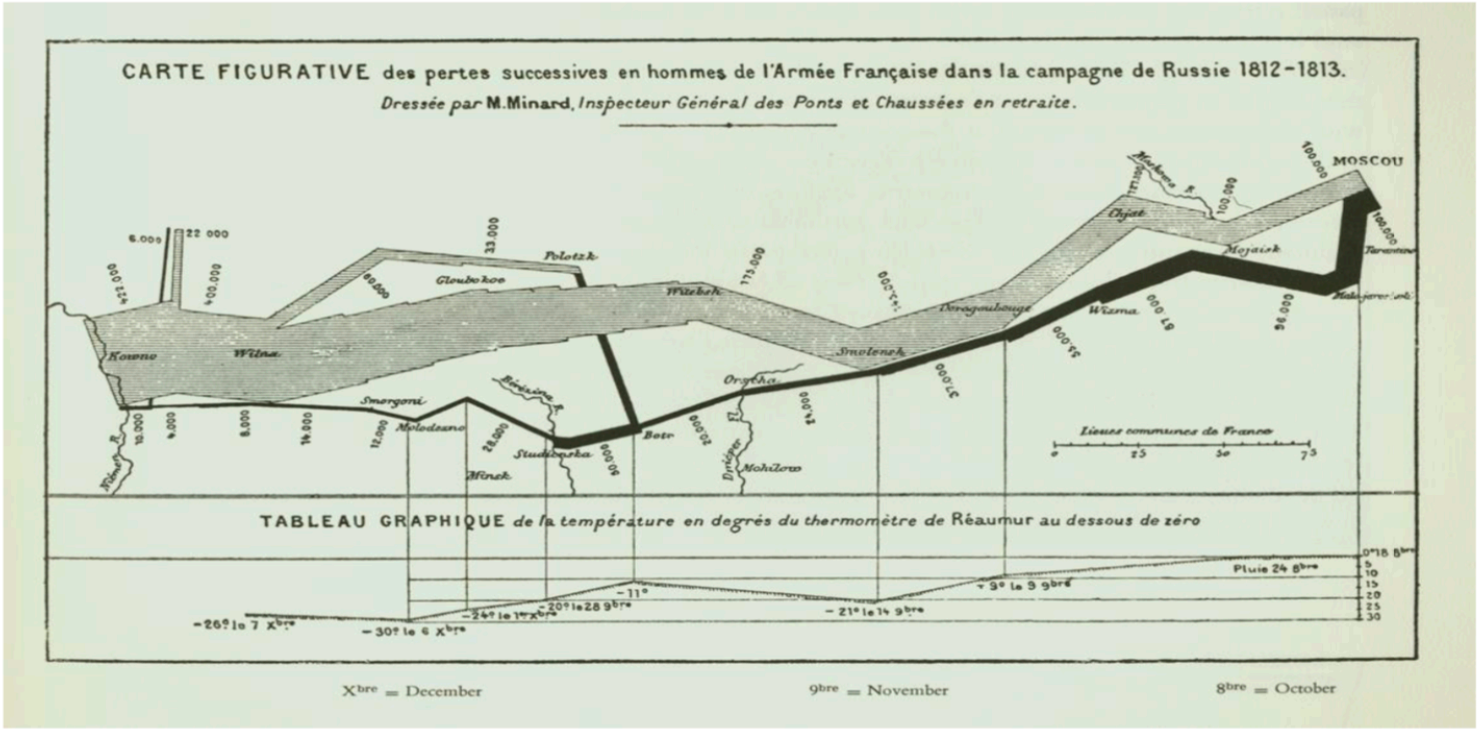
Motivation

- 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



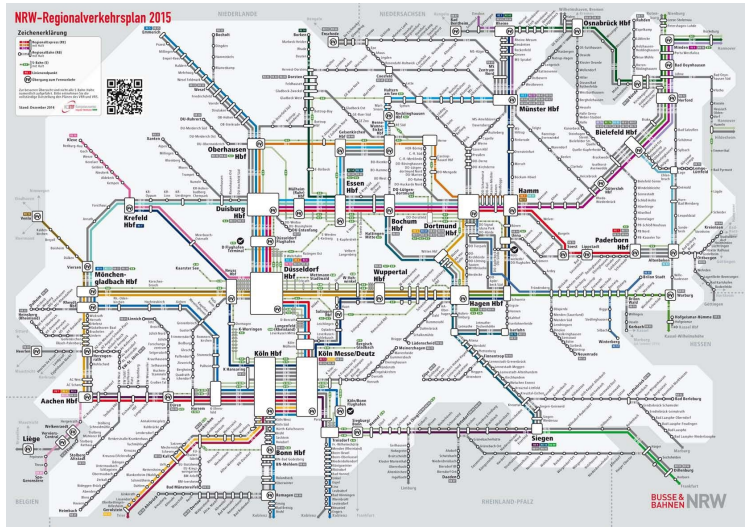
Motivation

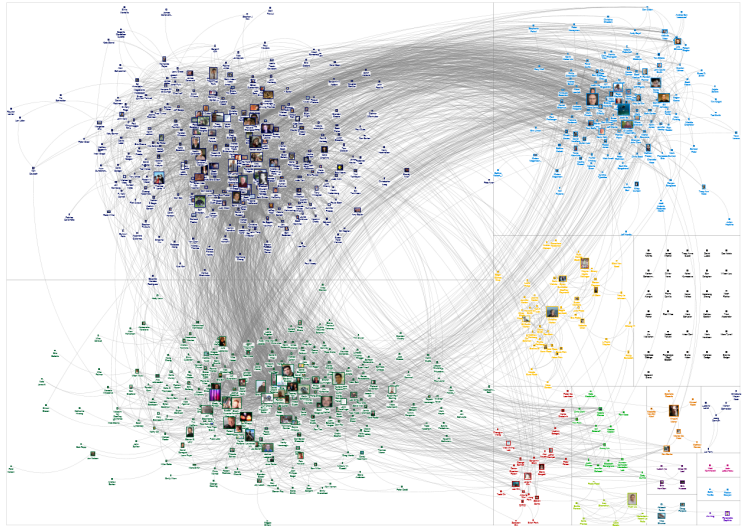
- Minard (1861): Map of Napoleon's campaign in Russia (1812/13)"the best statistical drawing ever made..." (Tufts)army strength
- Casualities
- Troop movements
- Temperature during the retreat conditions



Today visualizations are everywhere

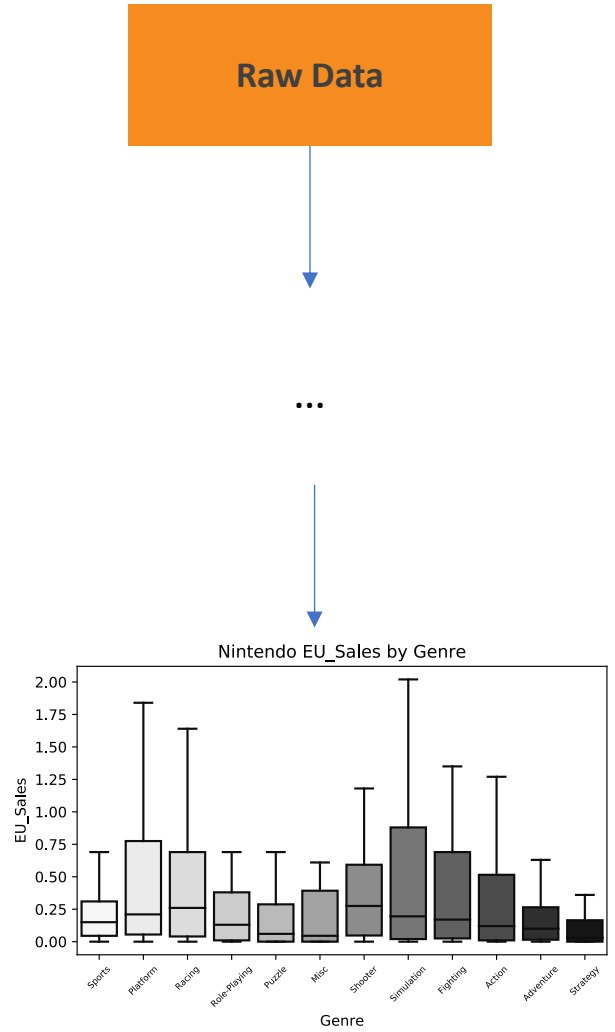
- Traffic
- Stock market
- Company
- Gaming
- …





Data/ Information Visualization

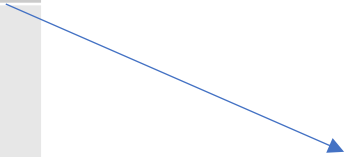
- *The use of computer-supported, interactive, **visual representations** of abstract data to amplify **cognition**. [Card et al 1999]*
- *Is the **communication** of abstract data through the use of interactive **visual interfaces**. [Keim et al 2006]*



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

…



V_Size
1.85
1.55

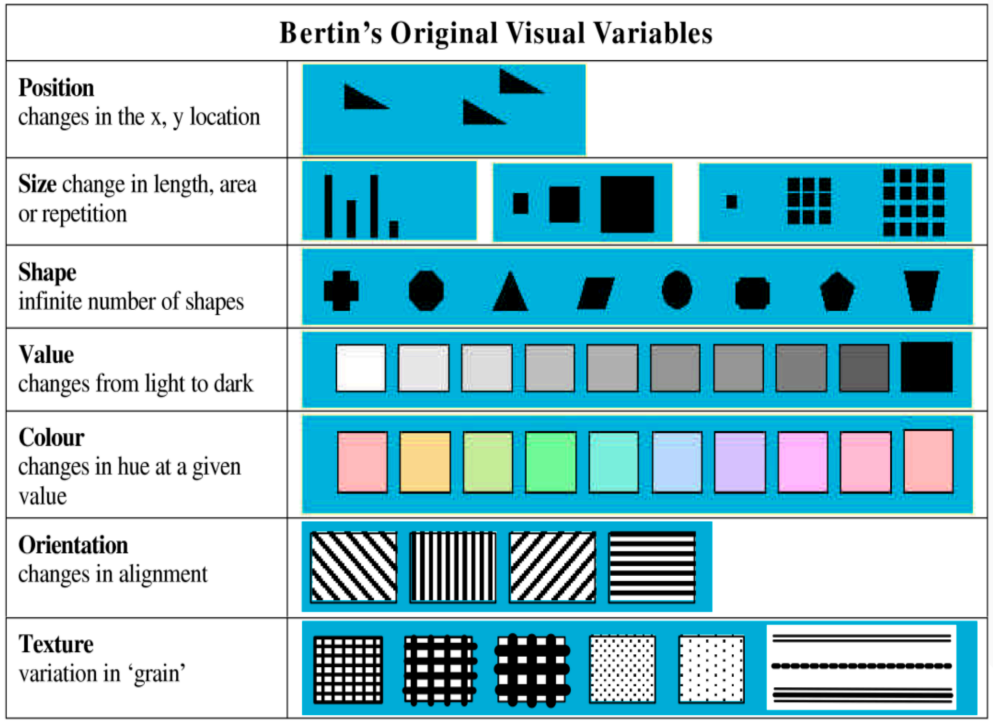
…

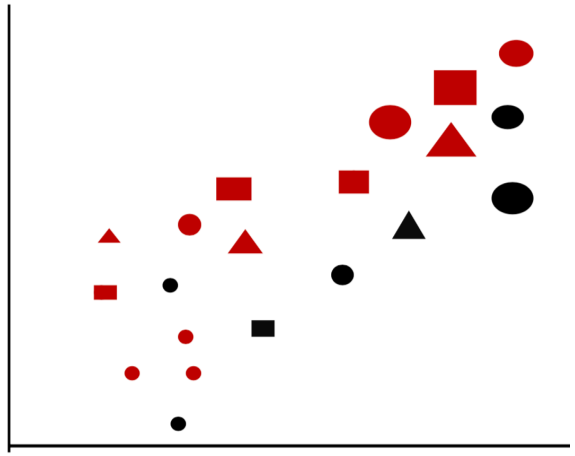
Basic Types of Variables for Visualizations (statistics)

- **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: =, >,<
- **Quantitative 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?

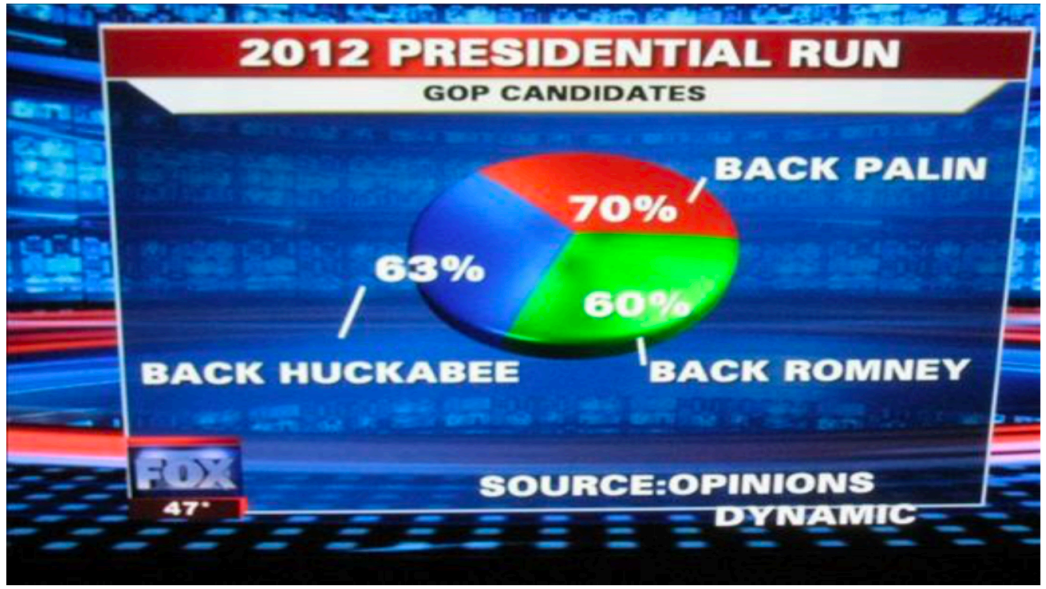
Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas since the dawn of humanity.



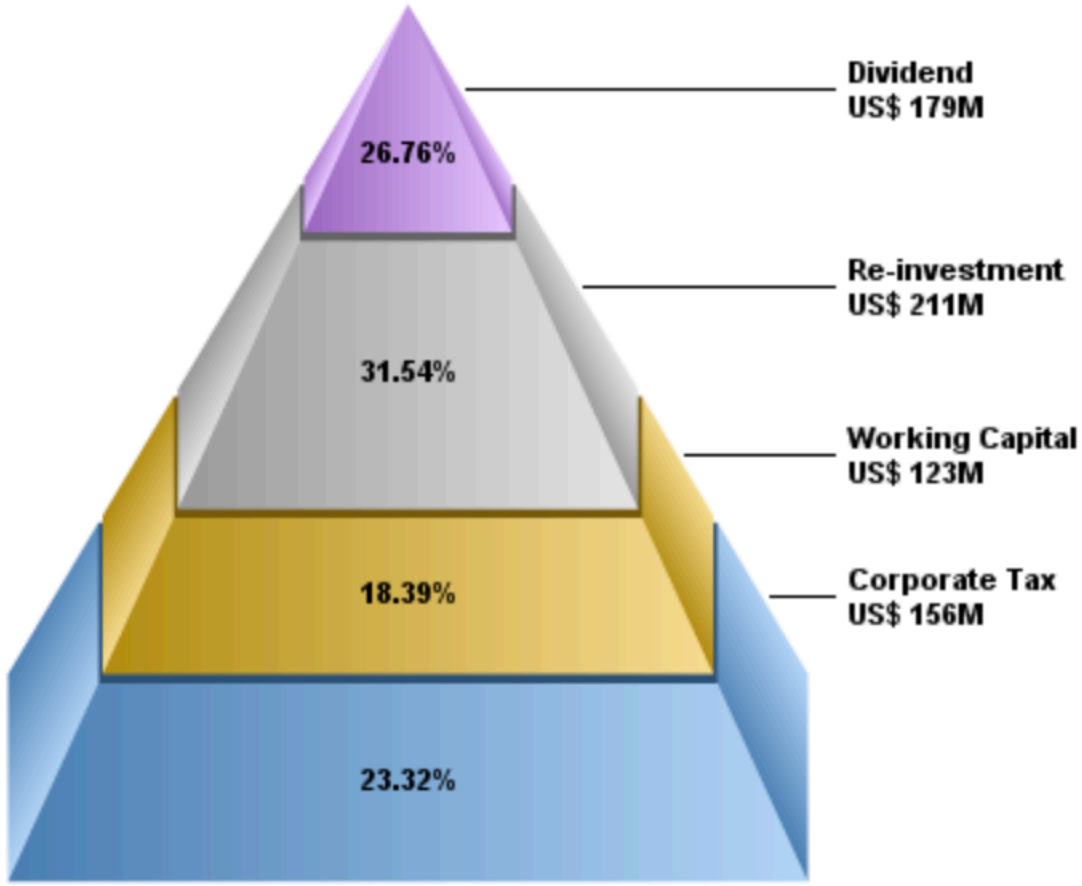


Good Visualizations?

- Let's get some ideas together…



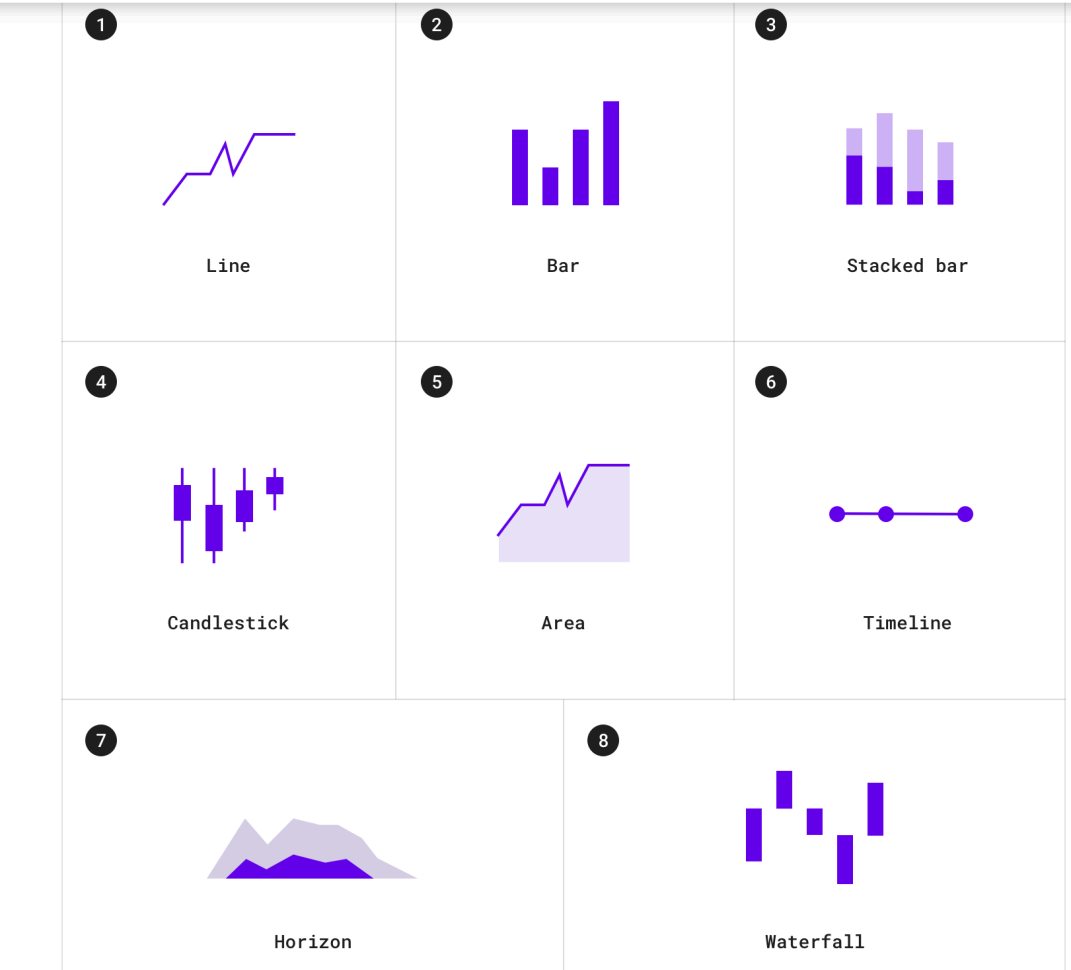
Fox News 2012 Presidential Run Quelle: wonkette.com



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

Data Visualization Style Guidelines

Communication > Data visualization > Types

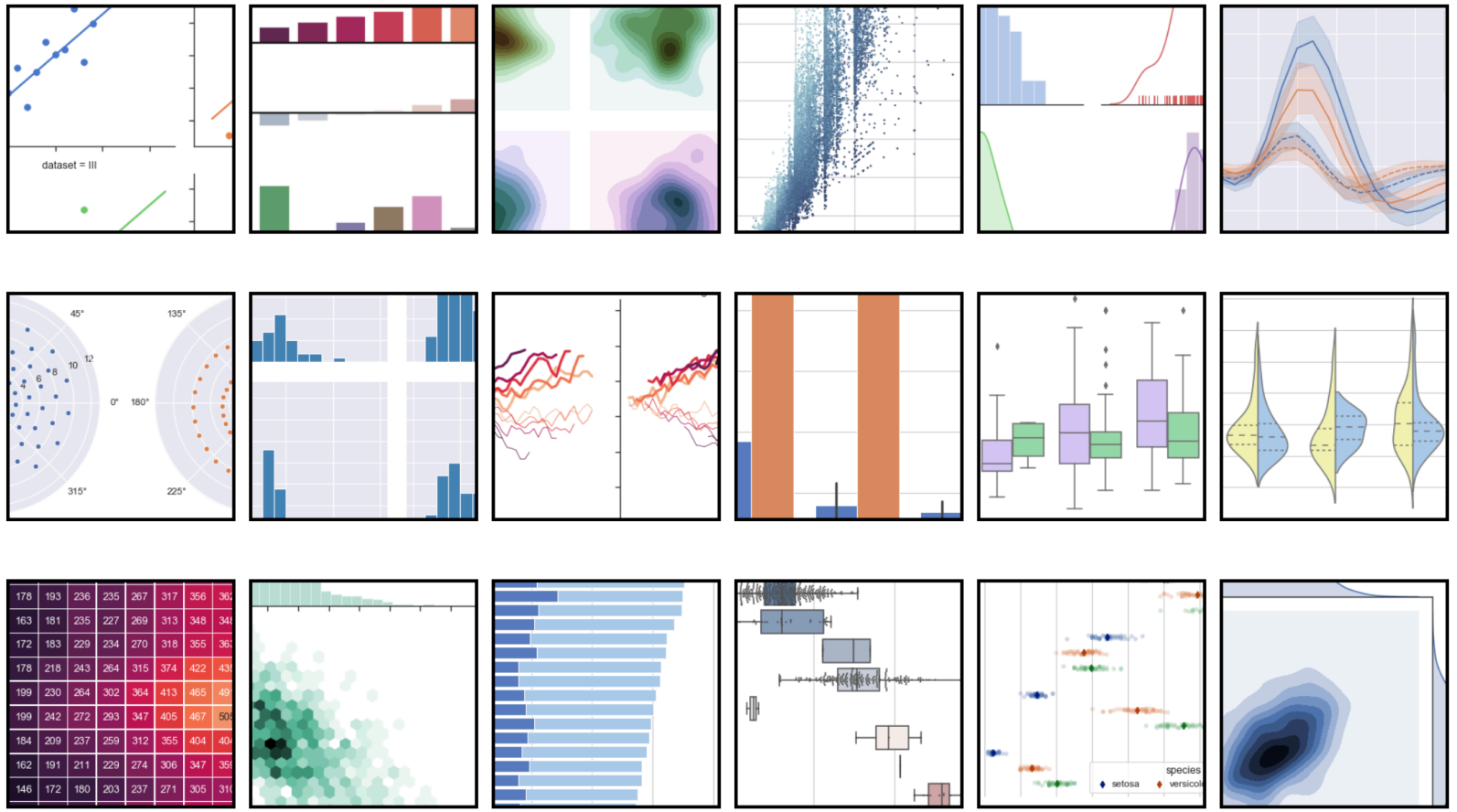


MATERIAL DESIGN

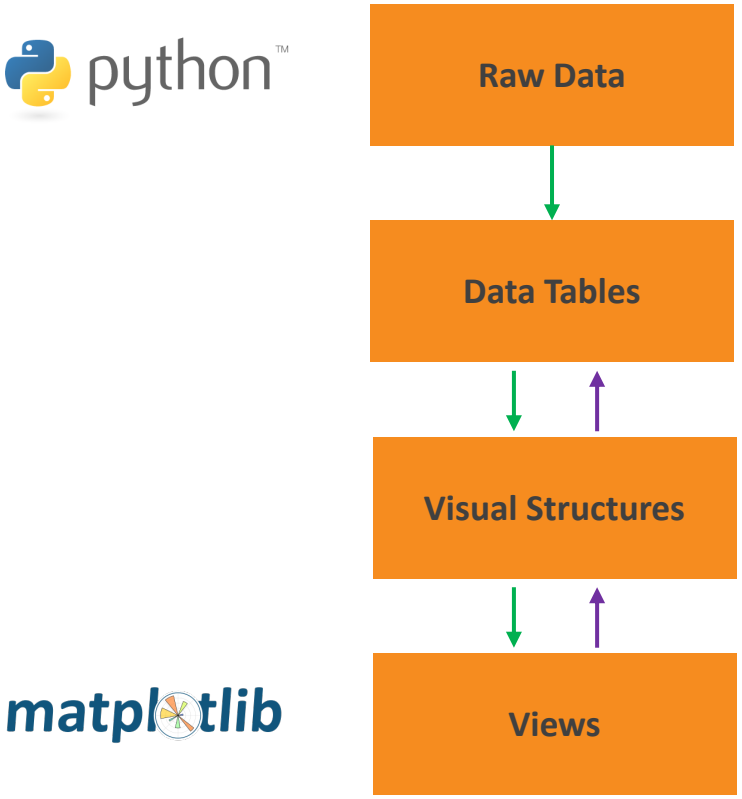
- https://material.io/design/communication/data-visualization.html#

Right Guidelines for good visualizations

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



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**?

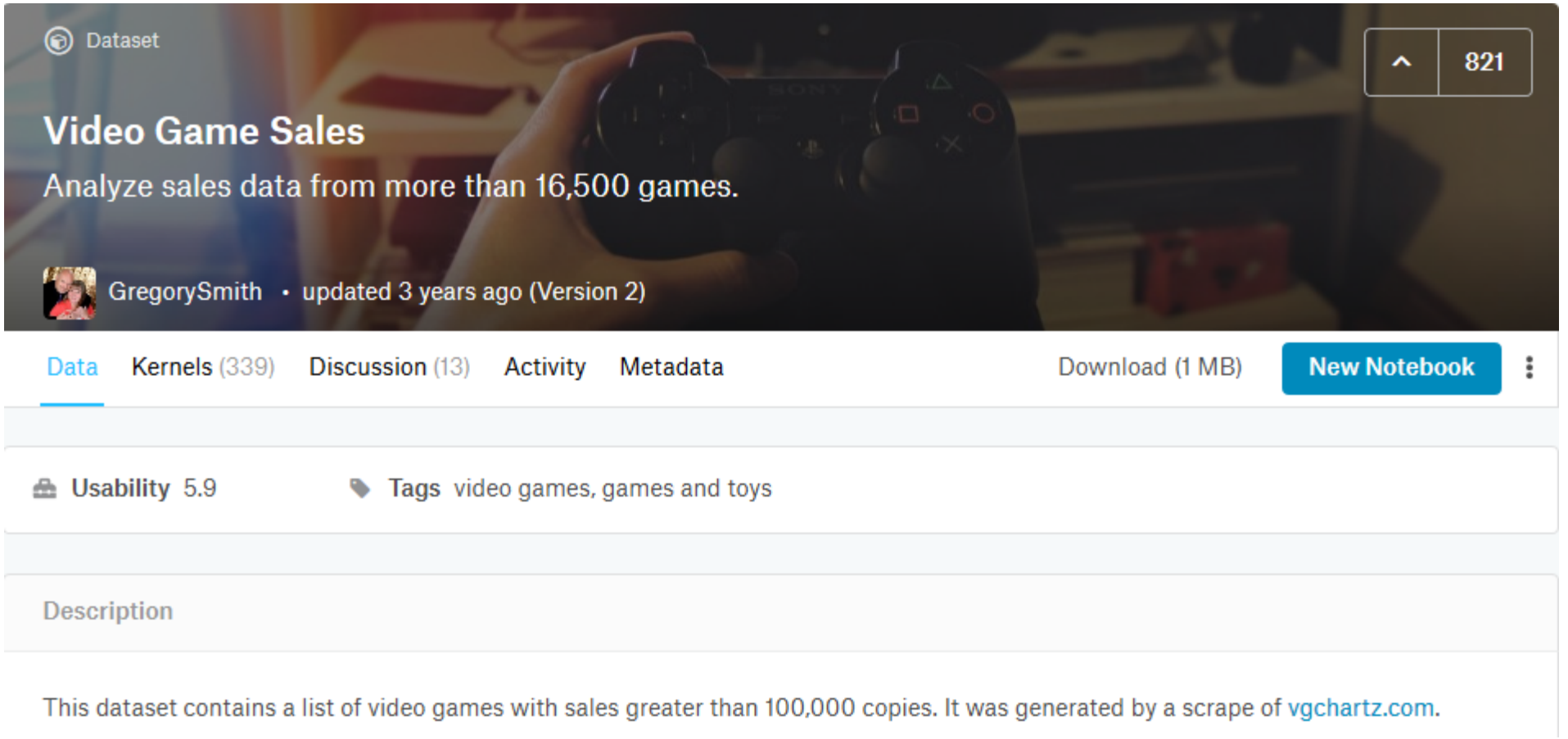
Plot the data.
Are there **anomalies**?
Are there **patterns**?

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



- 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

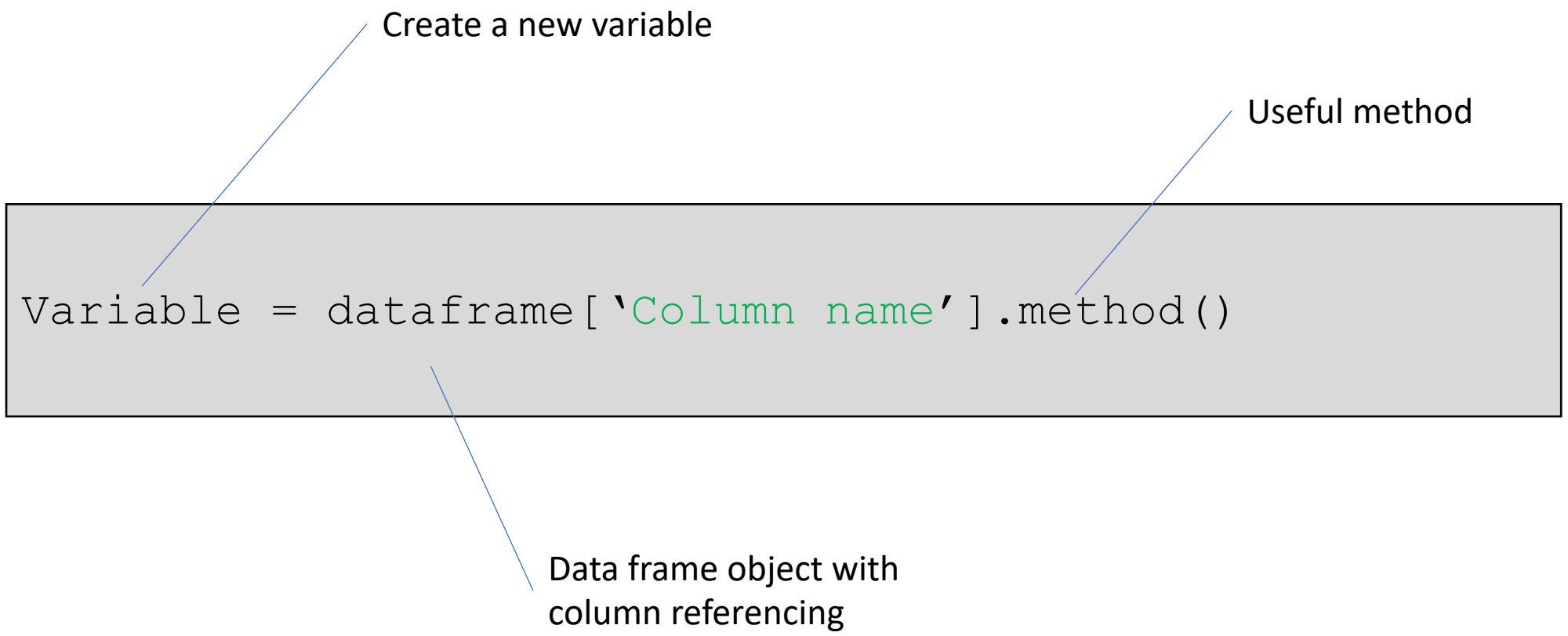
Choose plot

Additional adjustments

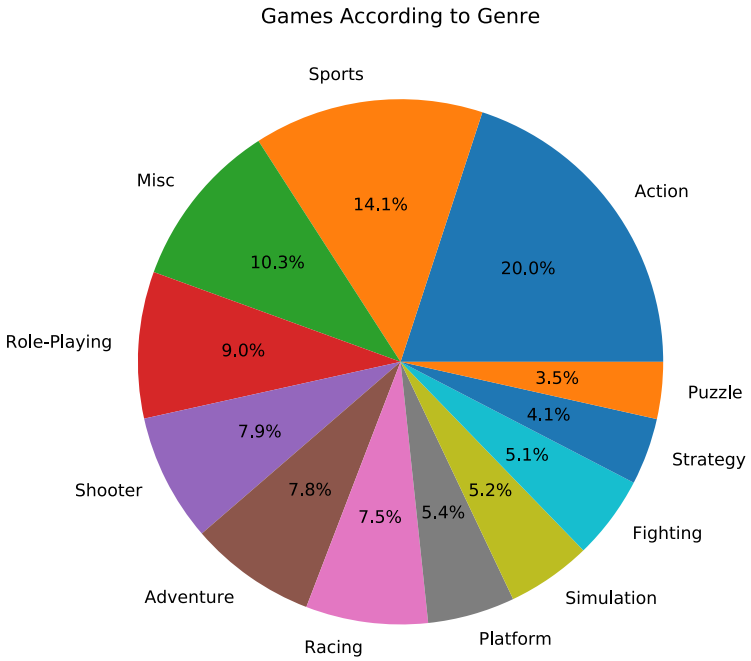
Save plot as .png or .pdf

Close plot

General Syntax for Data Access



Pic Chart

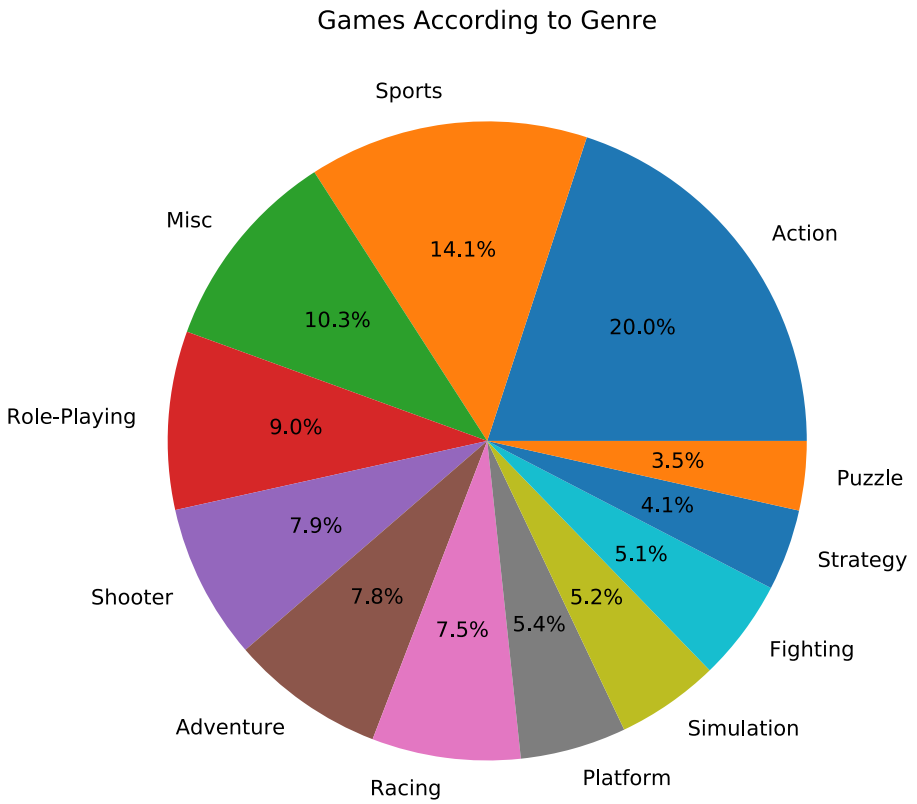

Games According to Genre
Sports
Misc
Action
14.1%
10.3%
20.0%
Role-Playing
9.0%
3.5%
Puzzle
4.1%
7.9%
Strategy
5.1%
Shooter
7.8%
5.2%
7.5%
5.4%
Fighting
Adventure
Simulation
Racing
Platform

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/sum(x). If sum(x) < 1, then the values of *x* give the fractional area directly.



Data Access for Pie Plot

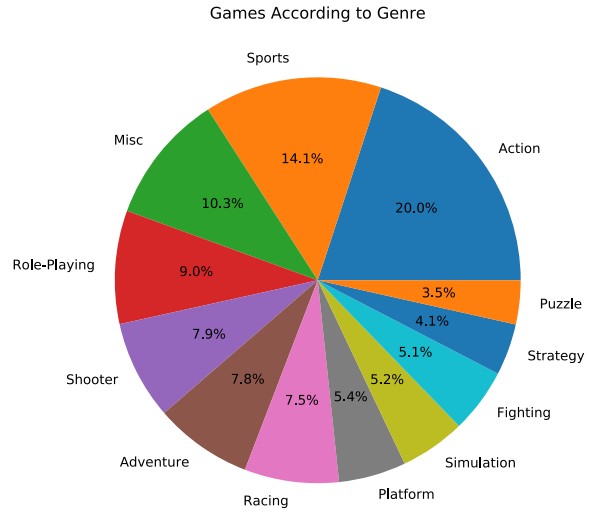
Return a series containing counts of unique values

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

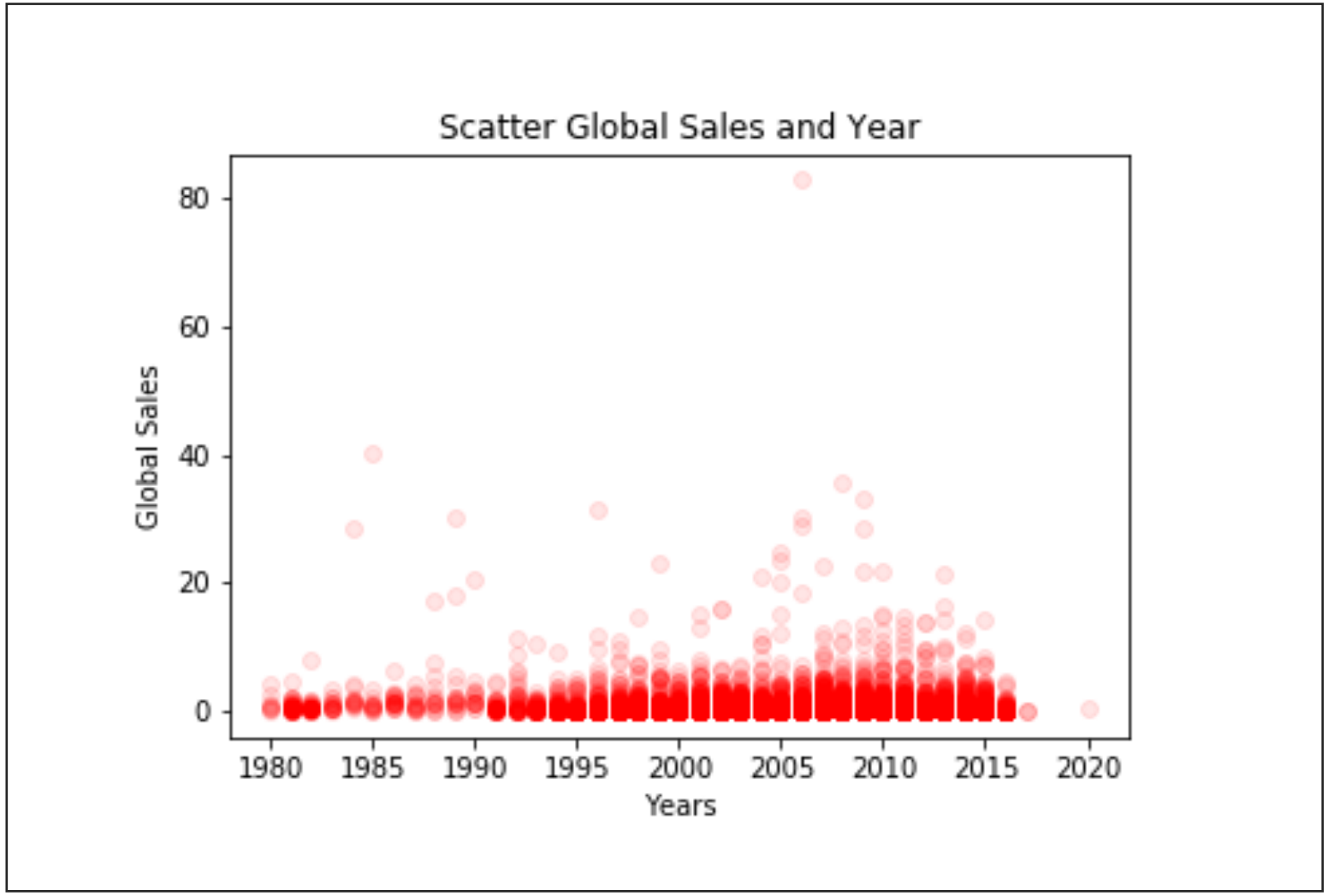
Return unique index like labels

Solution: Scatter Plot

```
plt.figure(figsize = (7,7))
plt.pie(LSizes, labels=LLabels,
autopct='%1.1f%%')
plt.title('Games According to Genre')
plt.savefig('PieGamesGenre.pdf')
```



Scatter Plot

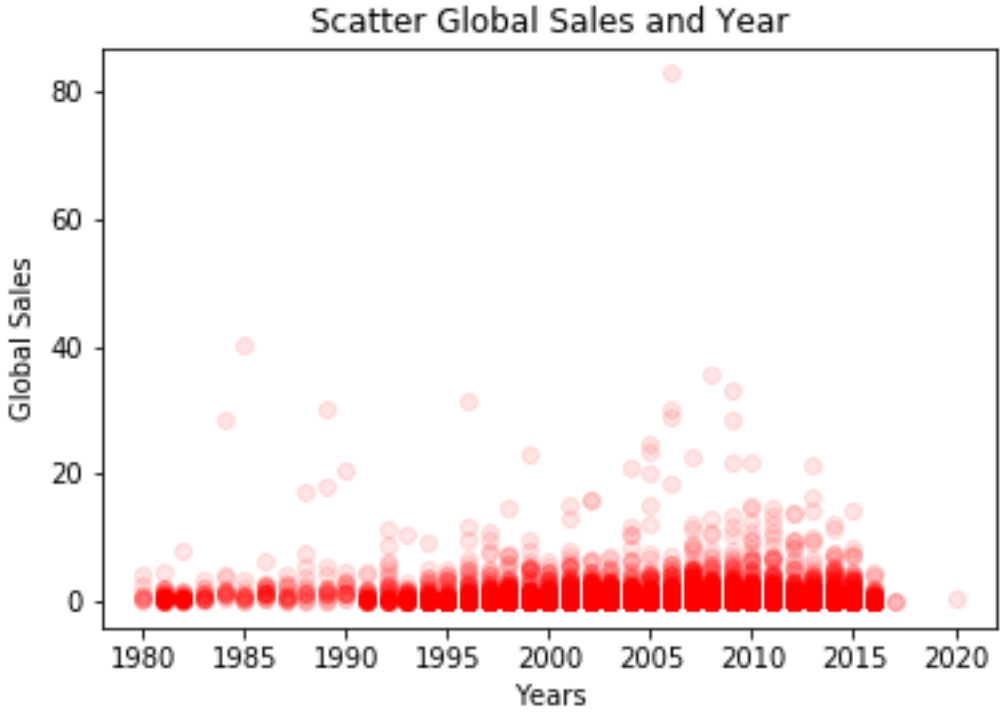


Video Games Global Sales by Years

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

Scatter plot:

- Scatter plot is a type of plot or mathematica 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’].get_values()
LYears = df['Year’].get_values()
```

Matplotlib Options

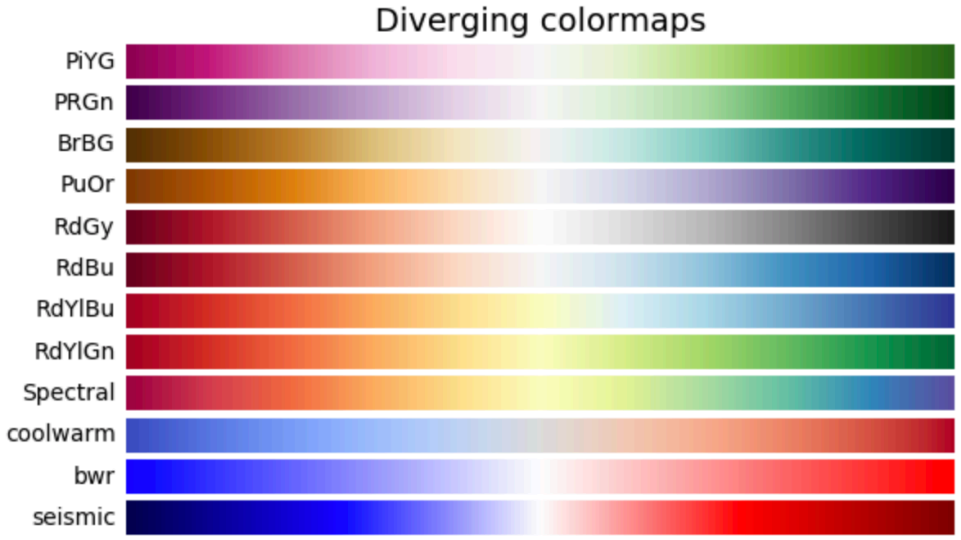
- Matplotlib plots has numerous options.
- See documentation is a good starting point
- https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.scatter.html
- Plots are functions with function parameters

```
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

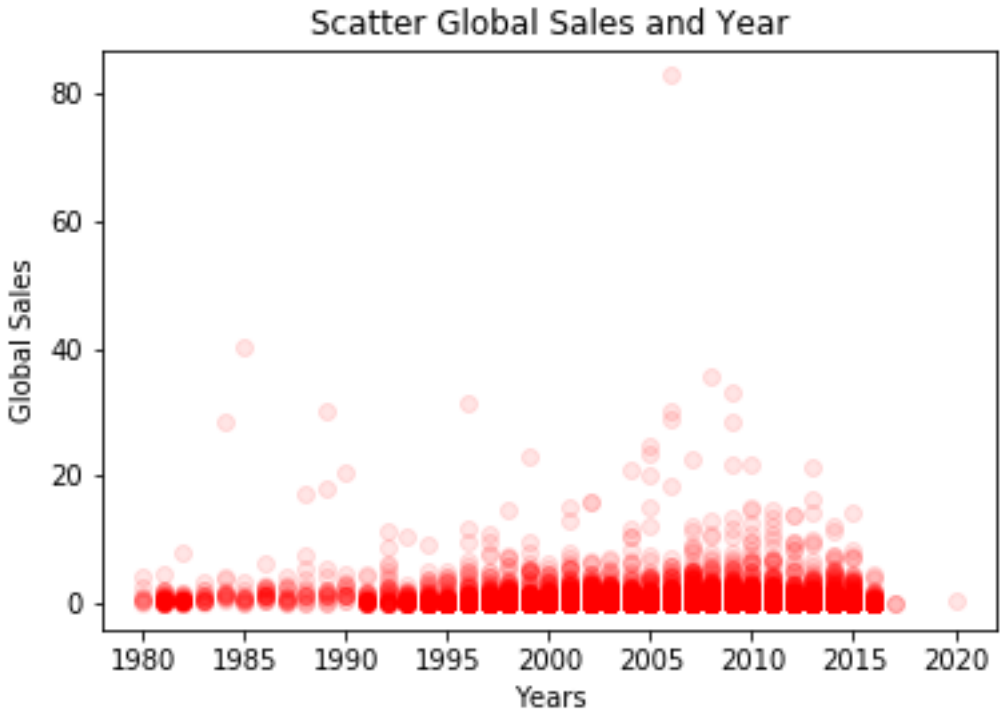
```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None,
 cmap=None, norm=None, vmin=None, vmax=None, alpha=None, lin
ewidths=None, verts=None, edgecolors=None, *, plotnonfinite=
False, data=None, **kwargs)[source]
```

There are hundreds of colours options:
https://matplotlib.org/3.1.0/tutorials/colors/colormaps.html



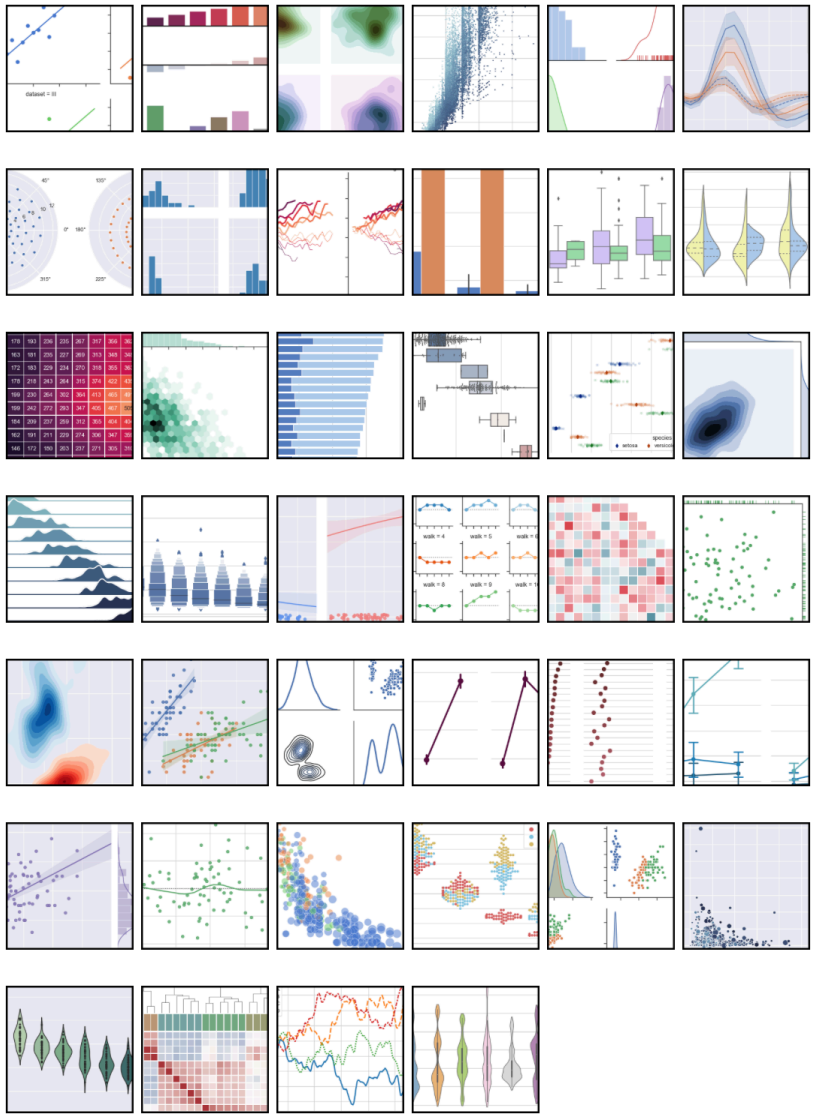
Solution: Scatter Plot

```
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()
```



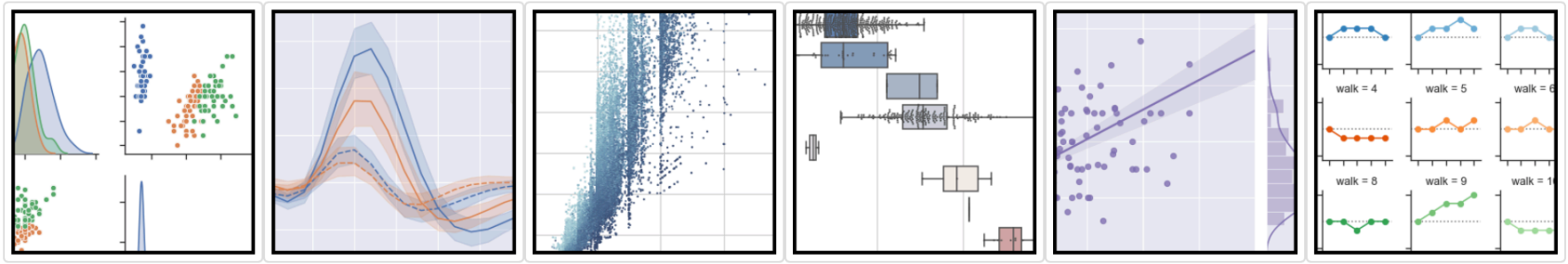
Seaborn

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.

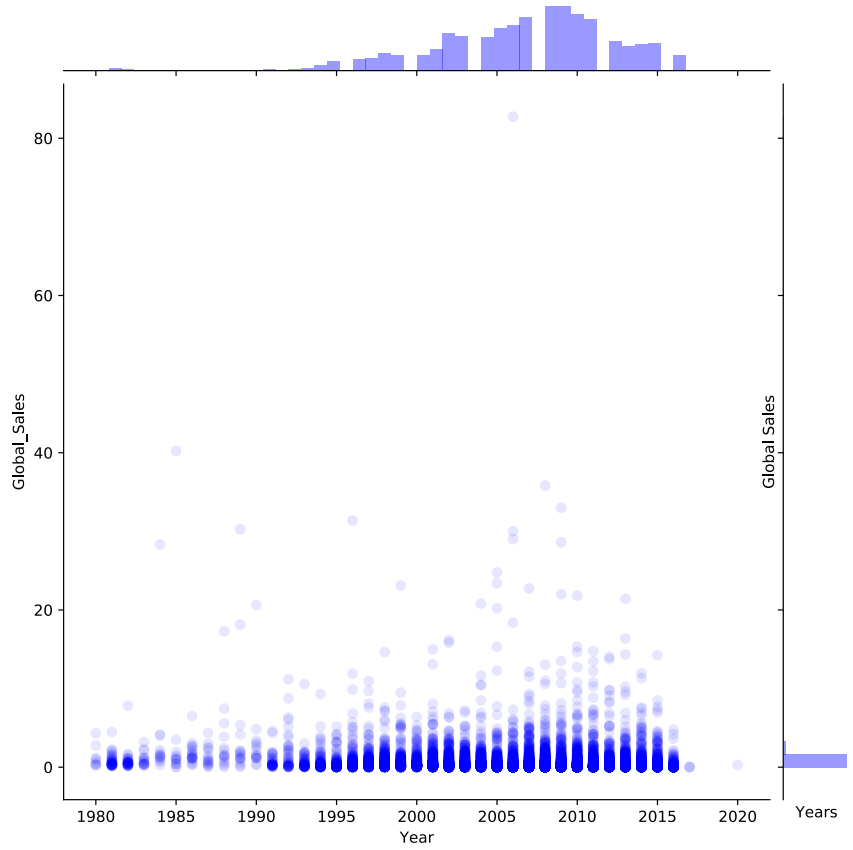


Joint Plot

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

Scatter plot:

- Draw a scatterplot with marginal histograms



Access Data for Joint Plot

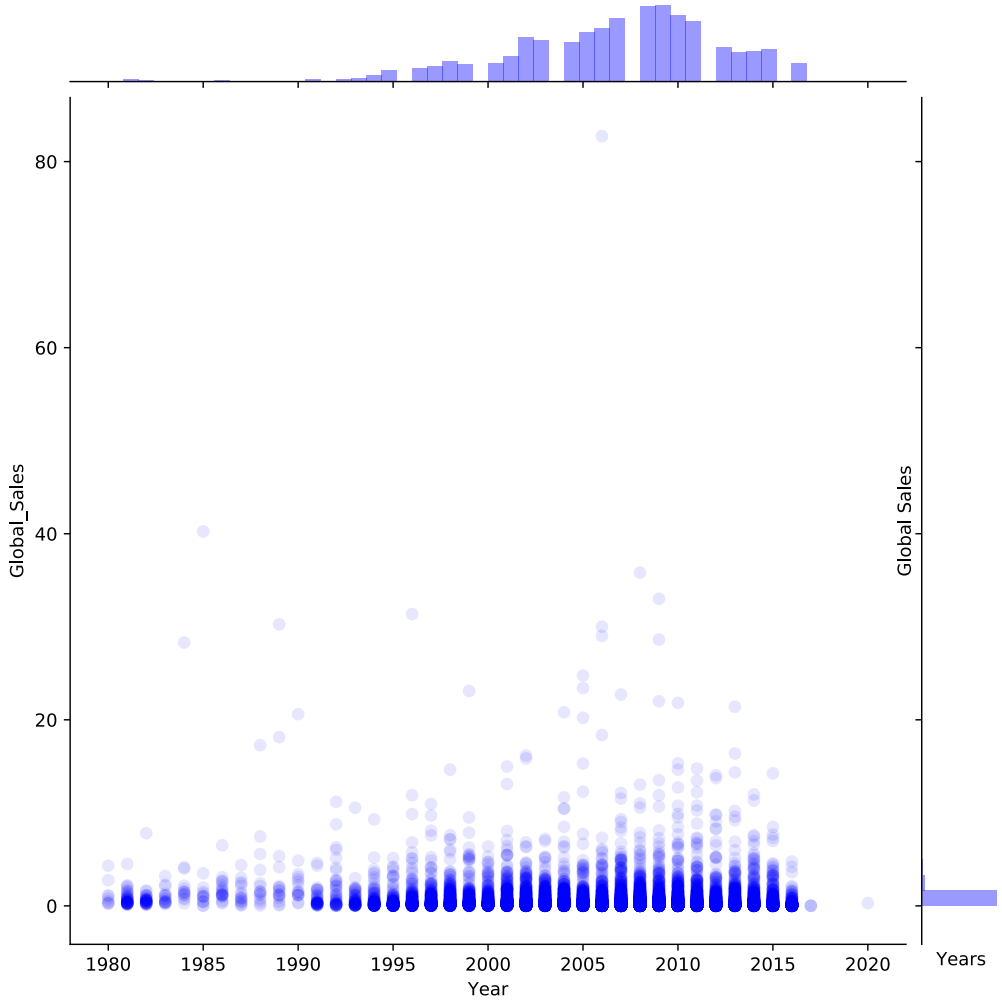
- The data frame object has some powerful operations

```
# Import the matploblib liberary.
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

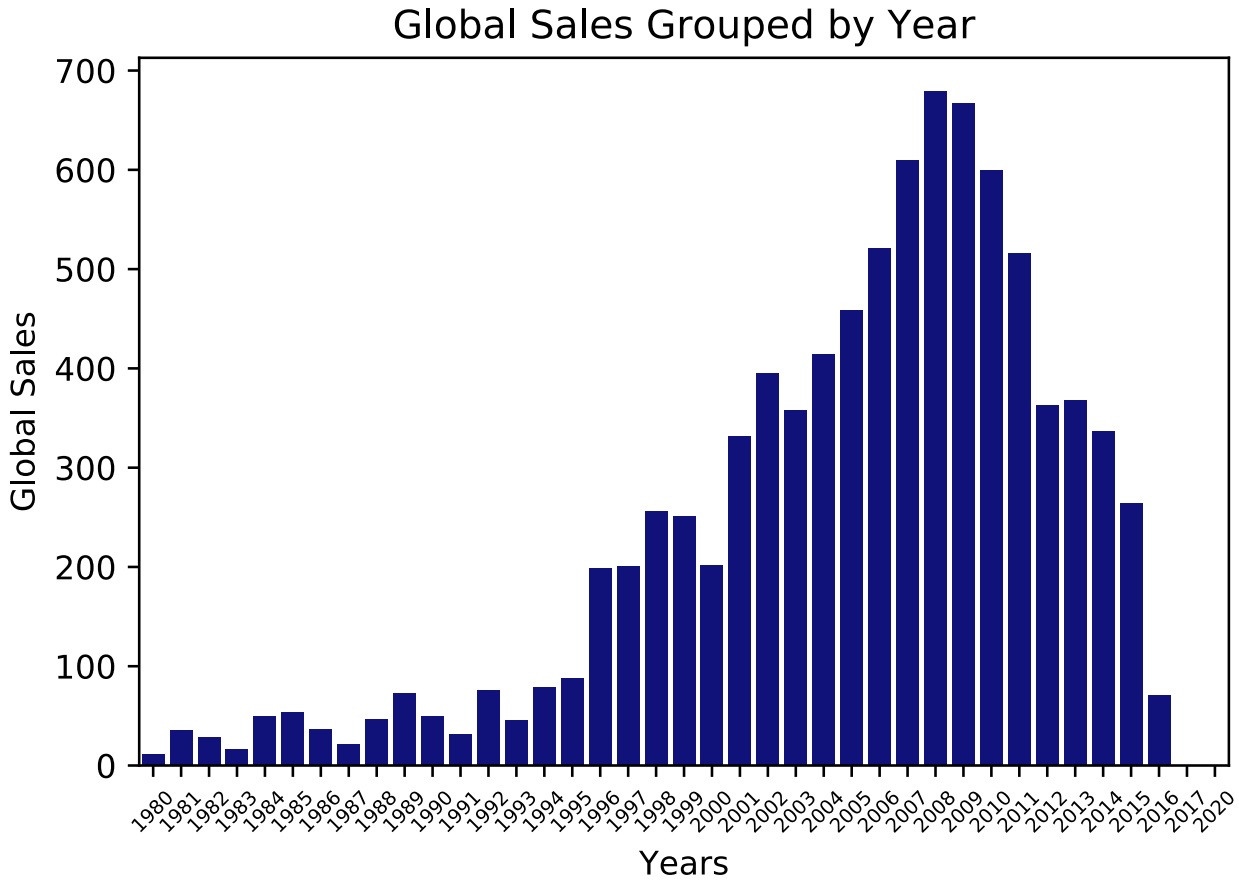
```
·# For using seaborn we need to use the bib.
import seaborn as sns

# 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 the One Plot



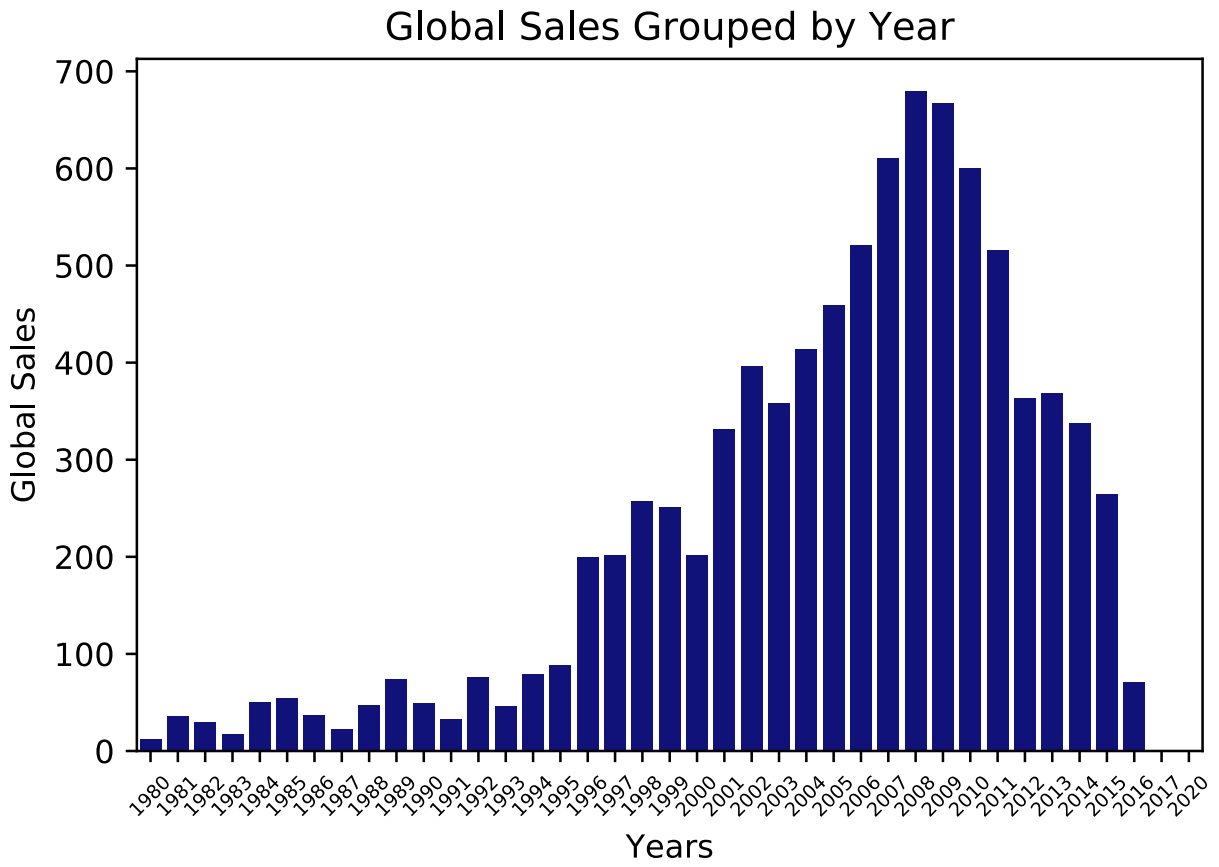
http://seaborn.pydata.org/generated/seaborn.jointplot.html?highlight=join%20plot

Bar Chart


Global Sales Grouped by Year
Global Sales
Years
0
100
200
300
400
500
600
700
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2020

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



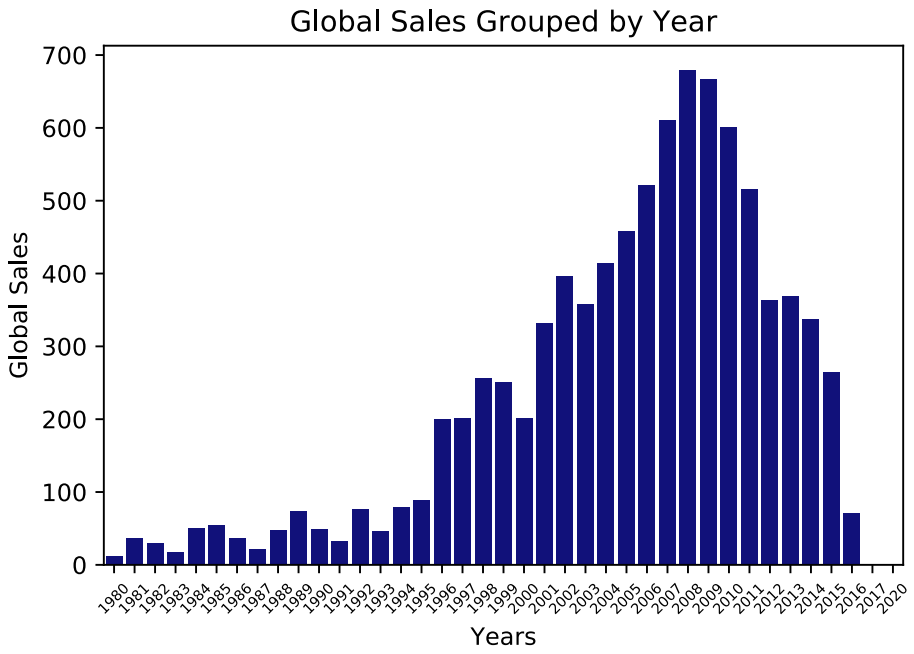
Access Data for Bar Char

- The data frame object has some powerful operations

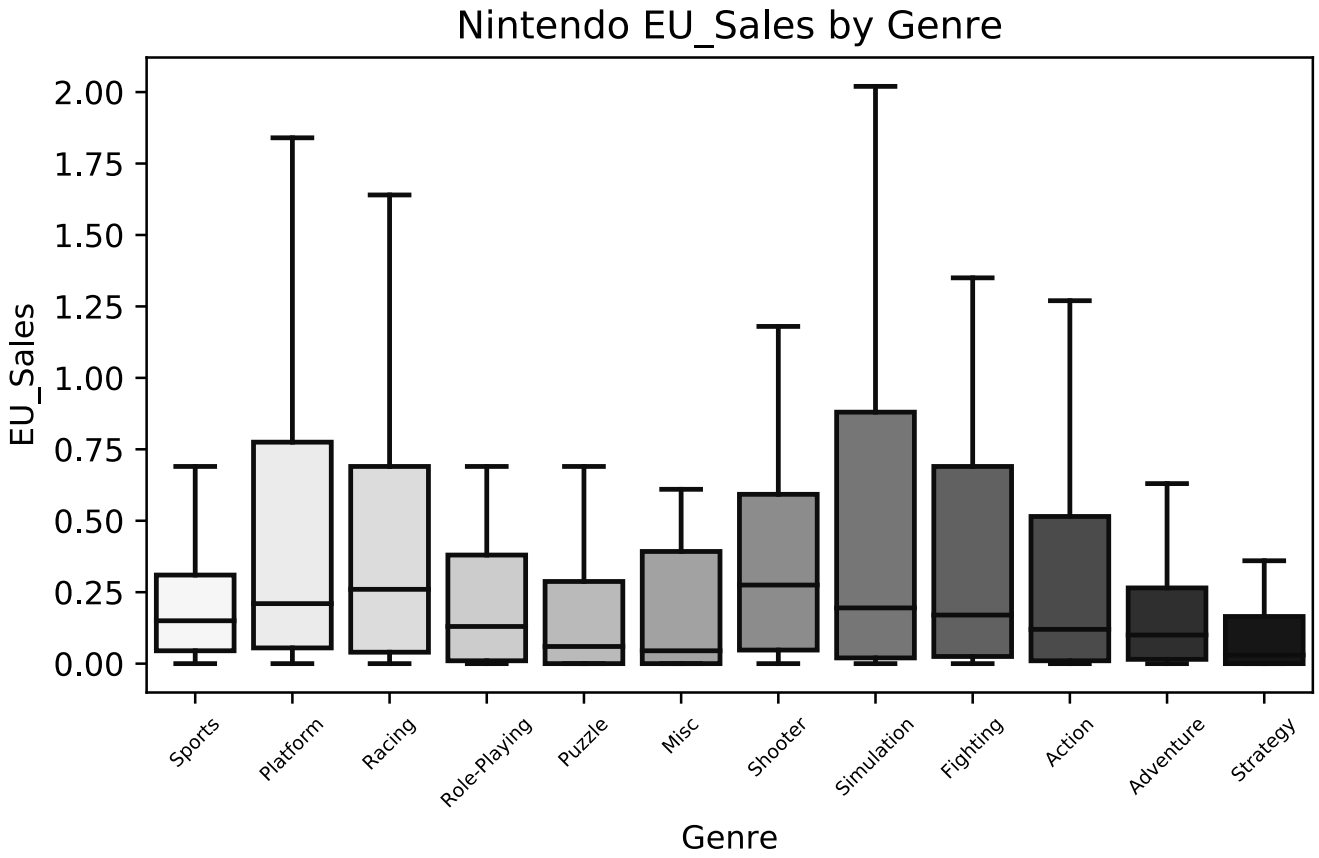
```
# Access data
# In our case sum data for each year.
# @code: df_groupData Grouped data by years and summed up.
# @code: LIndexesOfGroupData years converted as int!
df_groupData = df.groupby(['Year']).sum()
LdfGroupSales = df_groupData['Global_Sales']
LIndexesOfGroupData = df_groupData.index.astype(int)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

Solution: Bar Plot

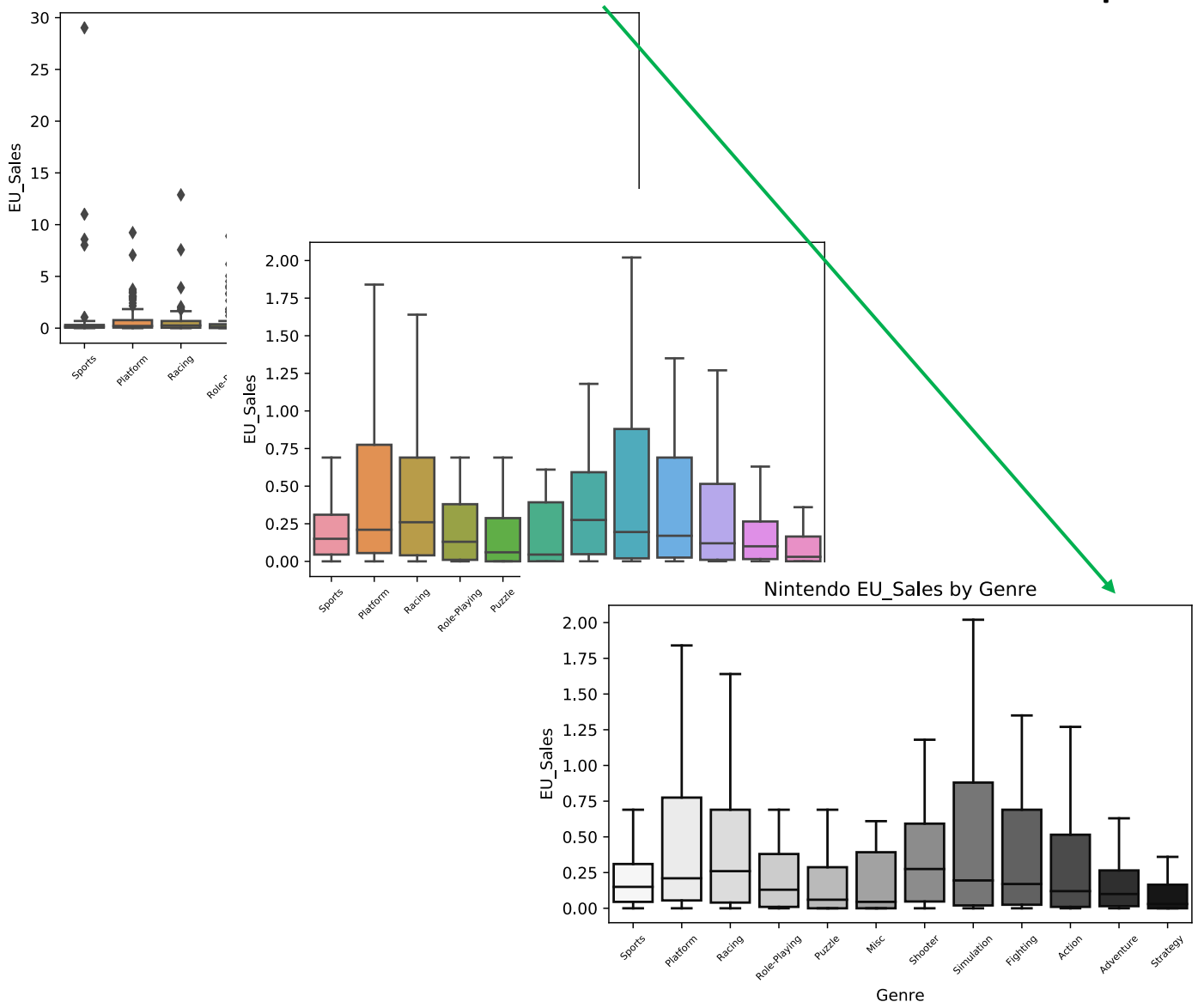
```
plt.figure()
sns.barplot(y=LdfGroupSales,
x=LIndexesOfGroupData, color='darkblue')
plt.title('Global Sales Grouped by Year')
plt.xticks(rotation=45, fontsize=6)
plt.ylabel('Global Sales')
plt.xlabel('Years')
plt.savefig('BarChart.pdf')
plt.show()
```

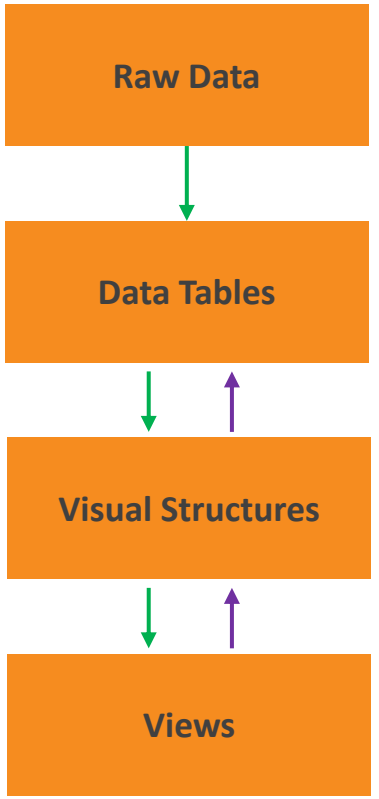


Box Plot


Nintendo EU_Sales by Genre
EU_Sales
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Sports
Platform
Racing
Role-Playing
Puzzle
Misc
Shooter
Simulation
Fighting
Action
Adventure
Strategy
Genre

Visualization Process - A deeper Look





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

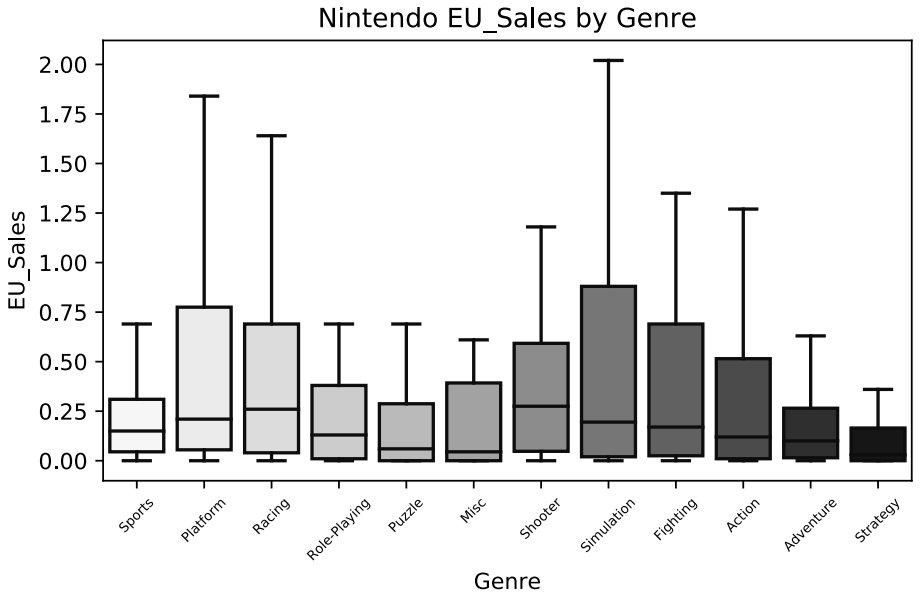
How is the data **sampled**?
Which data is **relevant**?

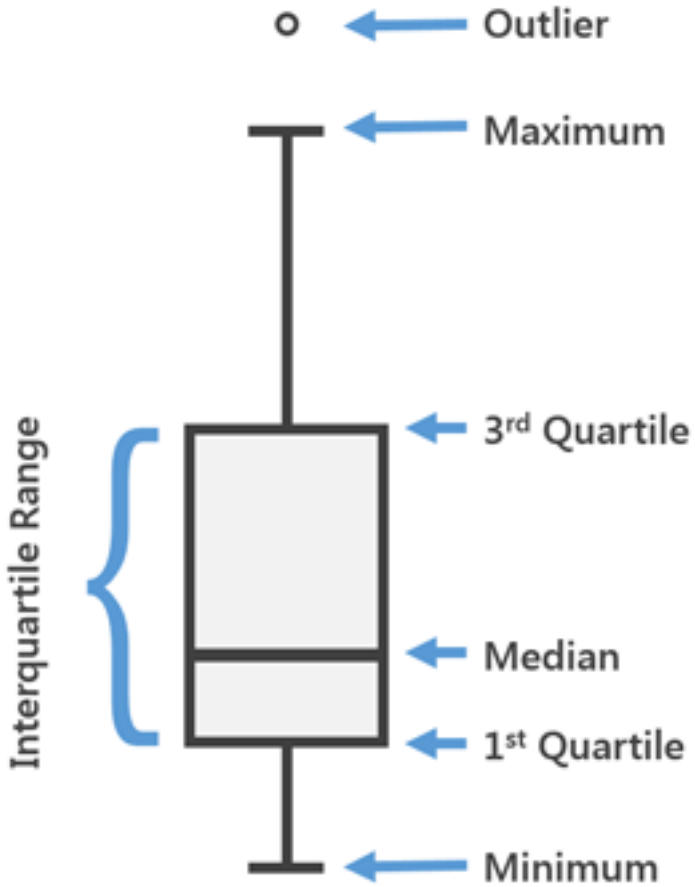
Plot the data.
Are there **anomalies**?
Are there **patterns**?

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_y0vPkAhVQYVAKHS5WAOQQjRx6BAgBEAQ&url=https%3A%2F%2Fpro.arcgis.com%2Fde%2Fpro-app%2Fhelp%2Fanalysis%2Fgeoprocessing%2Fcharts%2Fbox-plot.htm&psig=AOvVaw3Q-zP6jy0RYzK5LHM60kVF&ust=1569764533344246

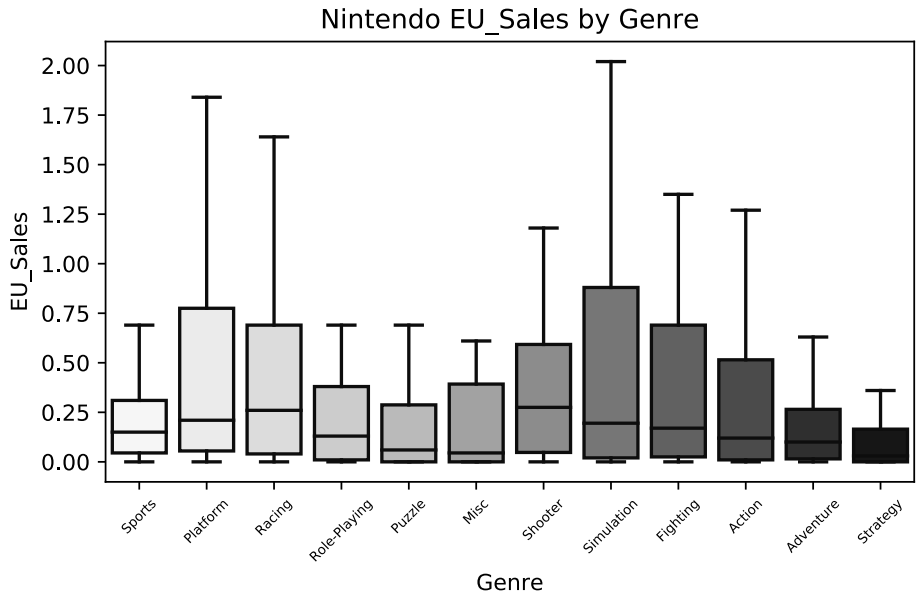
Access Data for Bar Char

- 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.

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://drive.google.com/drive/folders/0ByIrJAE4KMTtaGhRcXkxNHhmY2M
- https://python-graph-gallery.com/cheat-sheets/