Data Visualization

Visualization Definition

- **Visualization** is the use of computer graphics to create

visual images which aid in the understanding of complex, often massive representations of data.

- **Tables vs graphs**

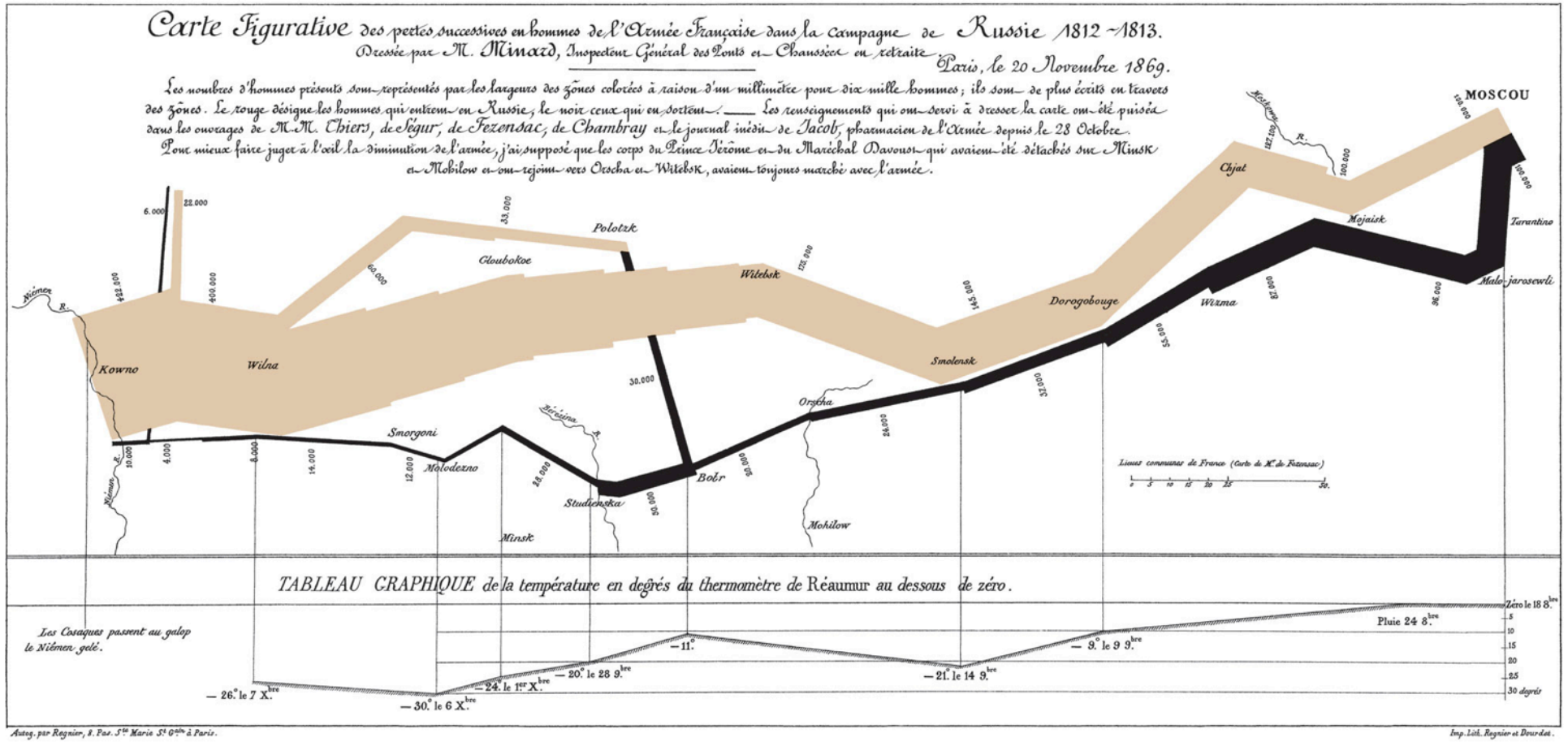
A table

- Need to look up specific values
- Need precise values
- need to precisely compare related values
- have multiple data sets with different units of measure

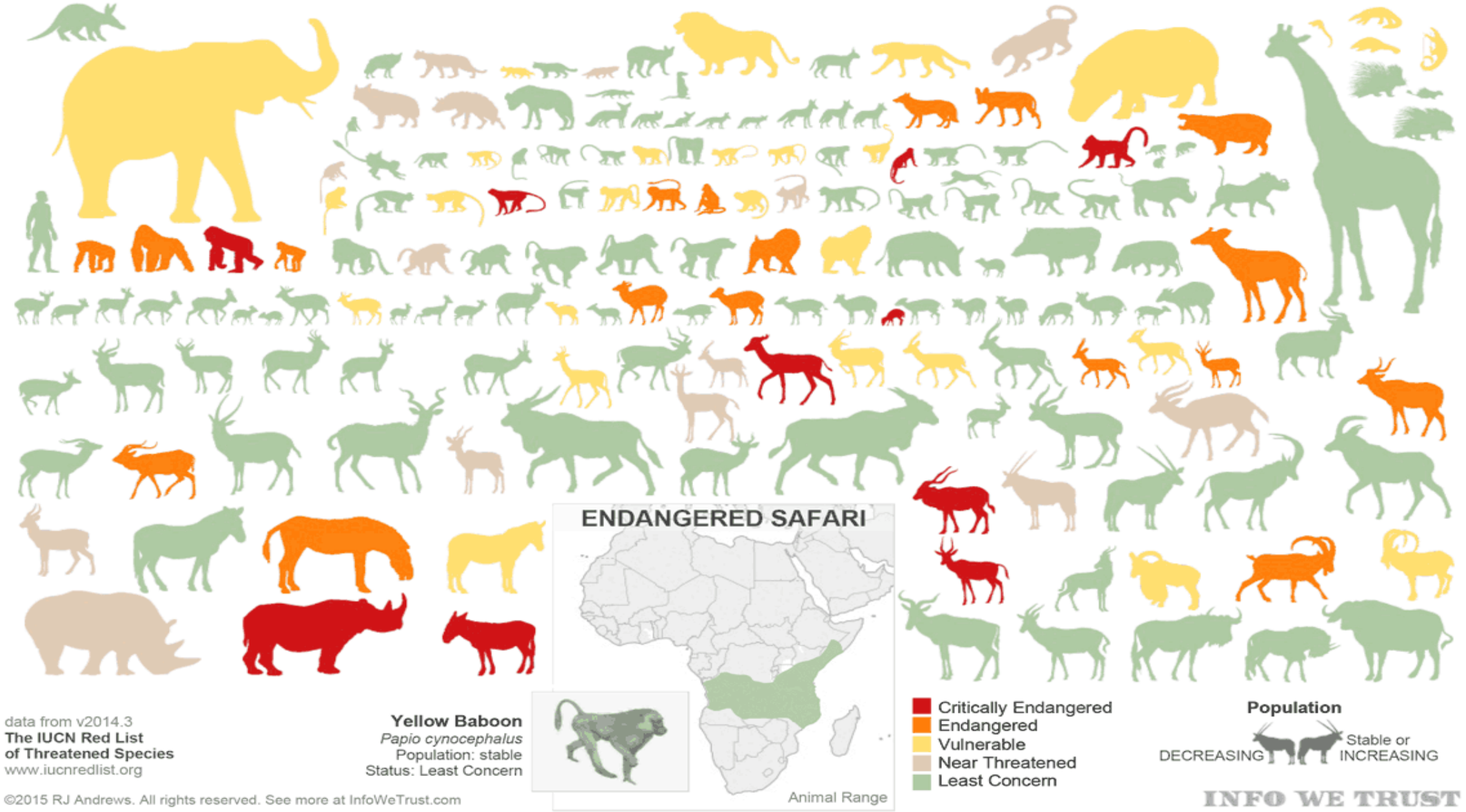
A graph

- The message is contained in the shape of the values
- Want to reveal relationships among multiple values (similarities and differences)
- Show general trends
- Have large data sets

Charles Joseph Minard 1869
Napoleon's March



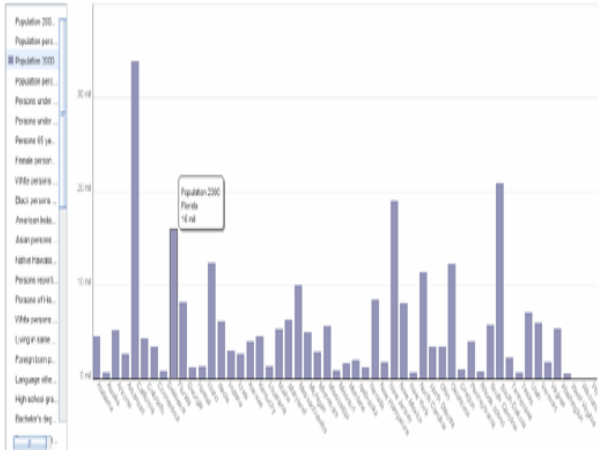
According to Tufte: "It may well be the best statistical graphic ever drawn."
5 variables: Army Size, location, dates, direction, temperature during retreat

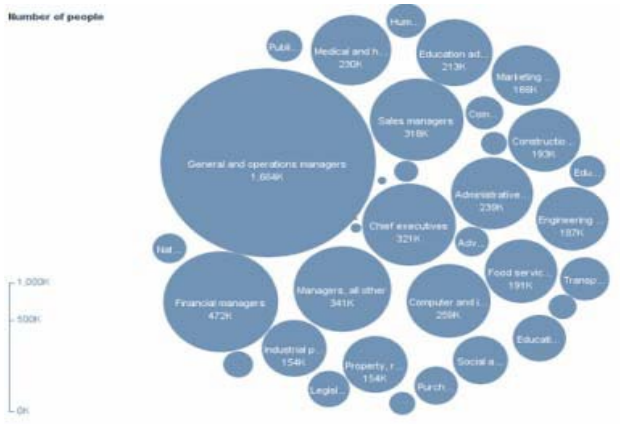


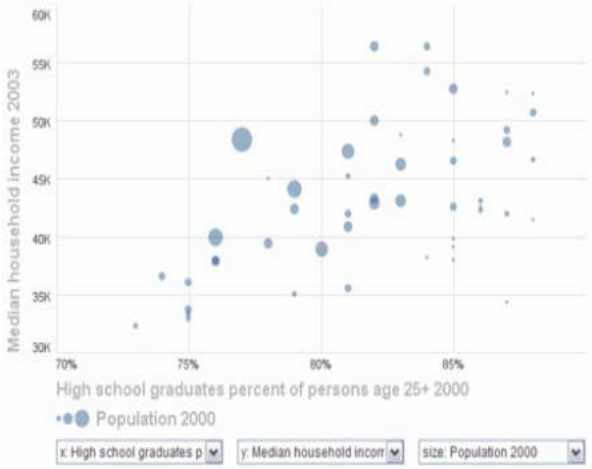
https://public.tableau.com/en-us/gallery/endangered-safari

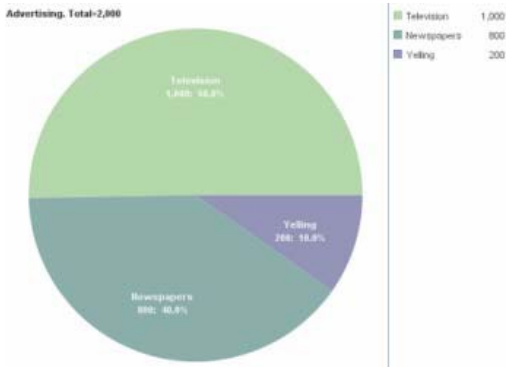
Data Visualization – Common Display Types

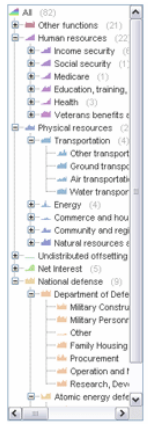
- Common Display Types
  - Bar Charts
  - Line Charts
  - Pie Charts
  - Bubble Charts
  - Stacked Charts
  - Scatterplots

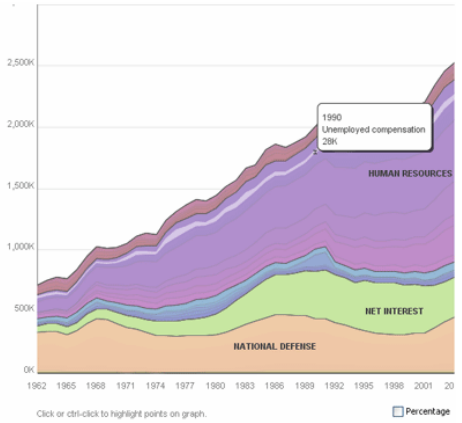


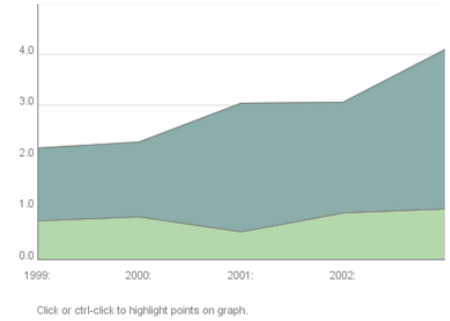






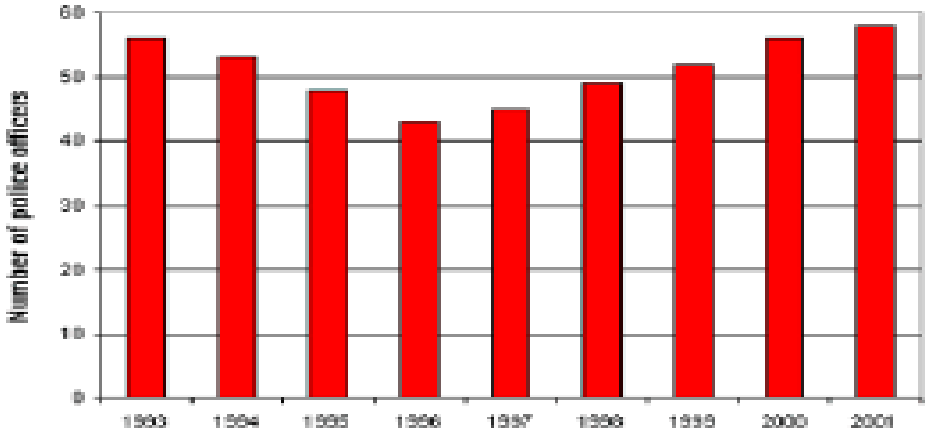




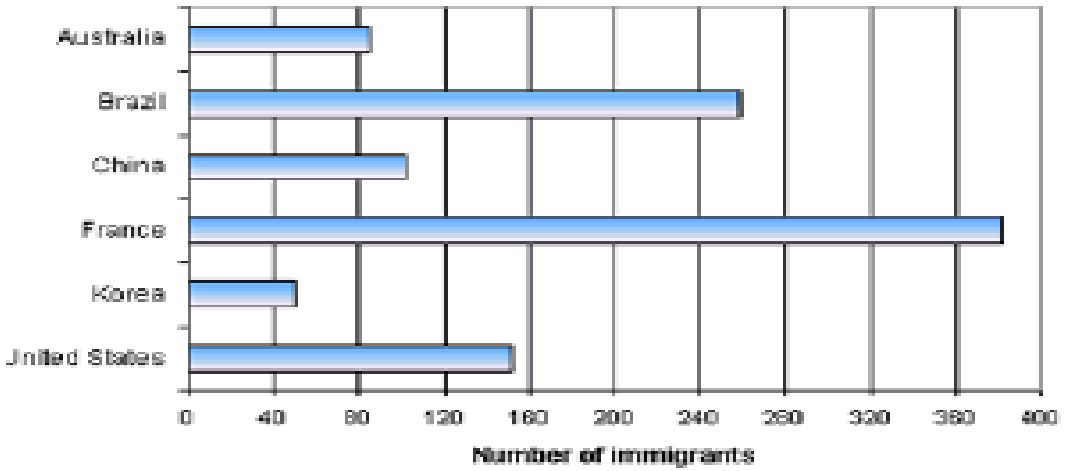


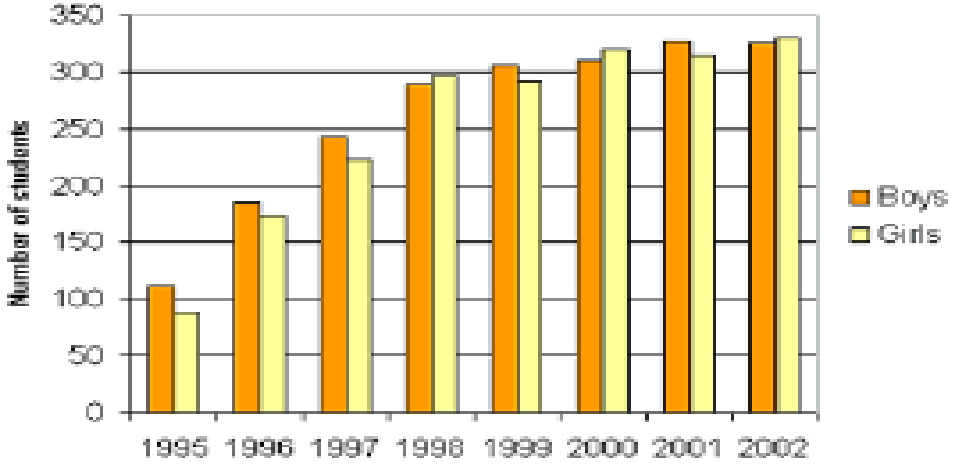
Bar Graph

- Bar graph
  - Presents categorical variables
  - Height of bar indicates value
  - Double bar graph allows comparison
  - Note spacing between bars
  - Can be horizontal (when would you use this?)



Number of police officers

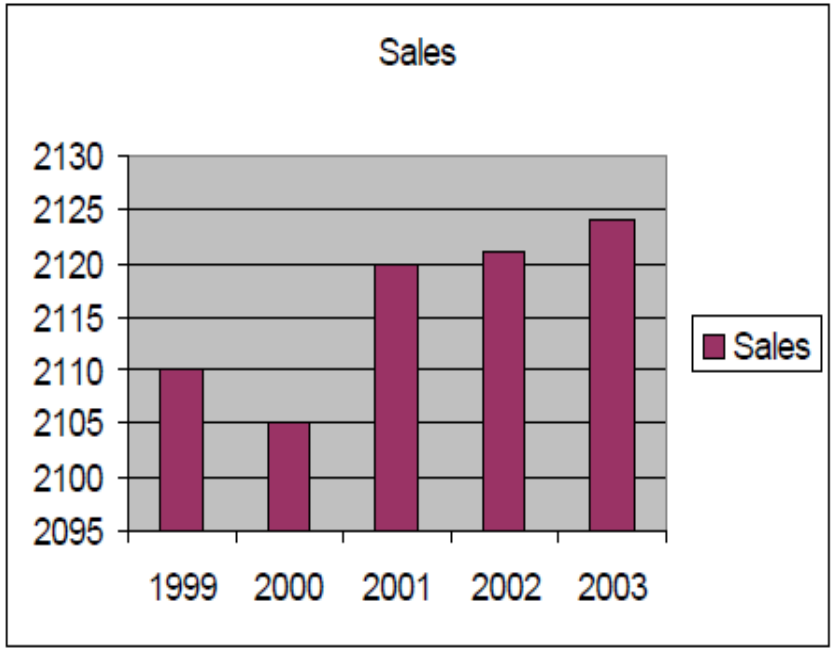




Internet use at a school

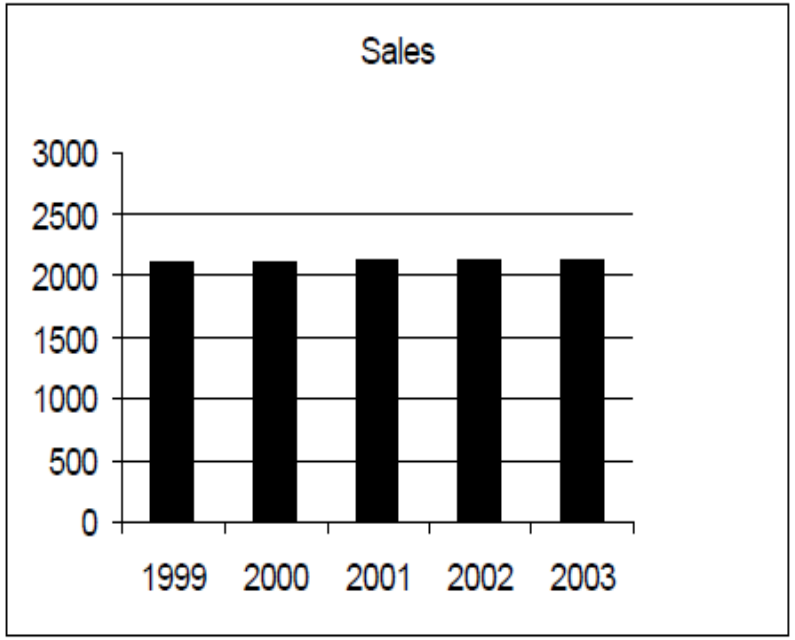
Comparison between good and bad

Year	Sales
1999	2,110
2000	2,105
2001	2,120
2002	2,121
2003	2,124



Y-Axis scale gives **WRONG** impression of big change

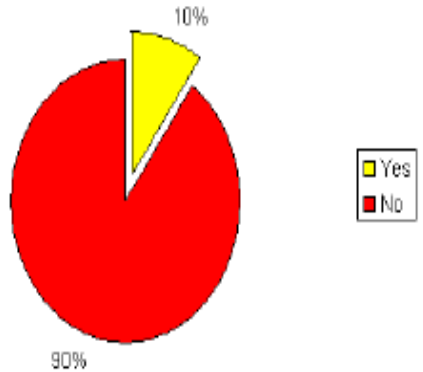
Year	Sales
1999	2,110
2000	2,105
2001	2,120
2002	2,121
2003	2,124



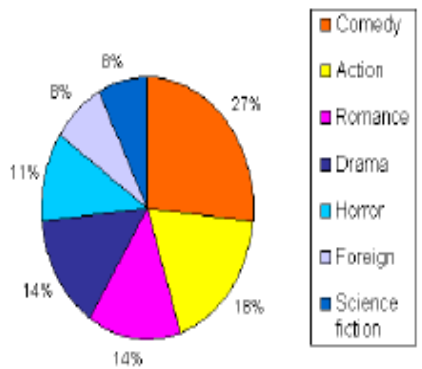
Axis from 0 to 2000 scale gives
correct impression of small change + small formatting tricks

Pie Chart & Scatter Plot

- Pie chart summarises a set of categorical/nominal data
- But use with care…
- … too many segments are harder to compare than in a bar chart



Should we have a long lecture?



Favourite movie genres

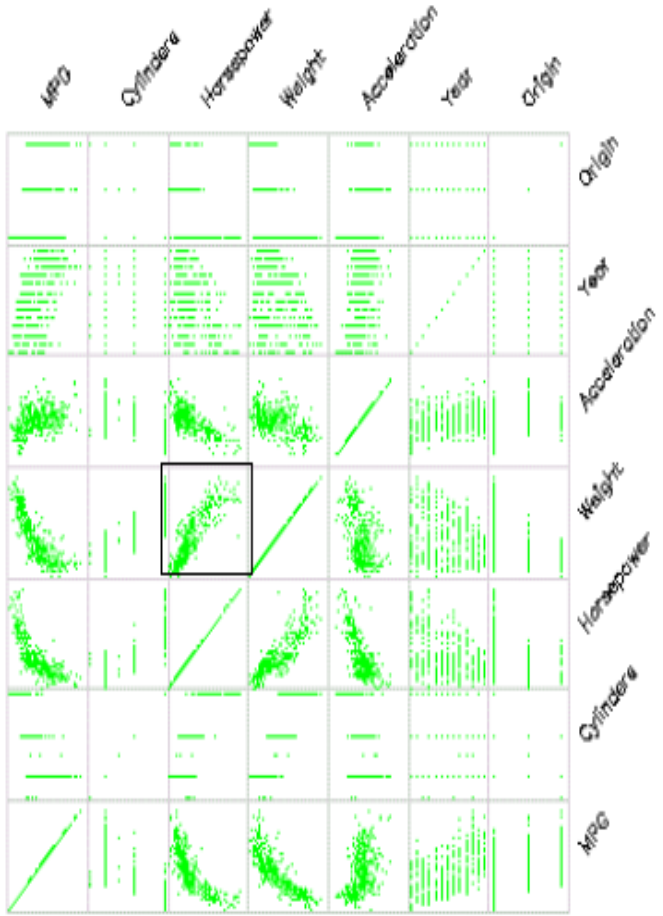
Represent each possible pair of variables in their own 2-D scatterplot (car data)

Q: Useful for what?

A: linear correlations (e.g. horsepower & weight)

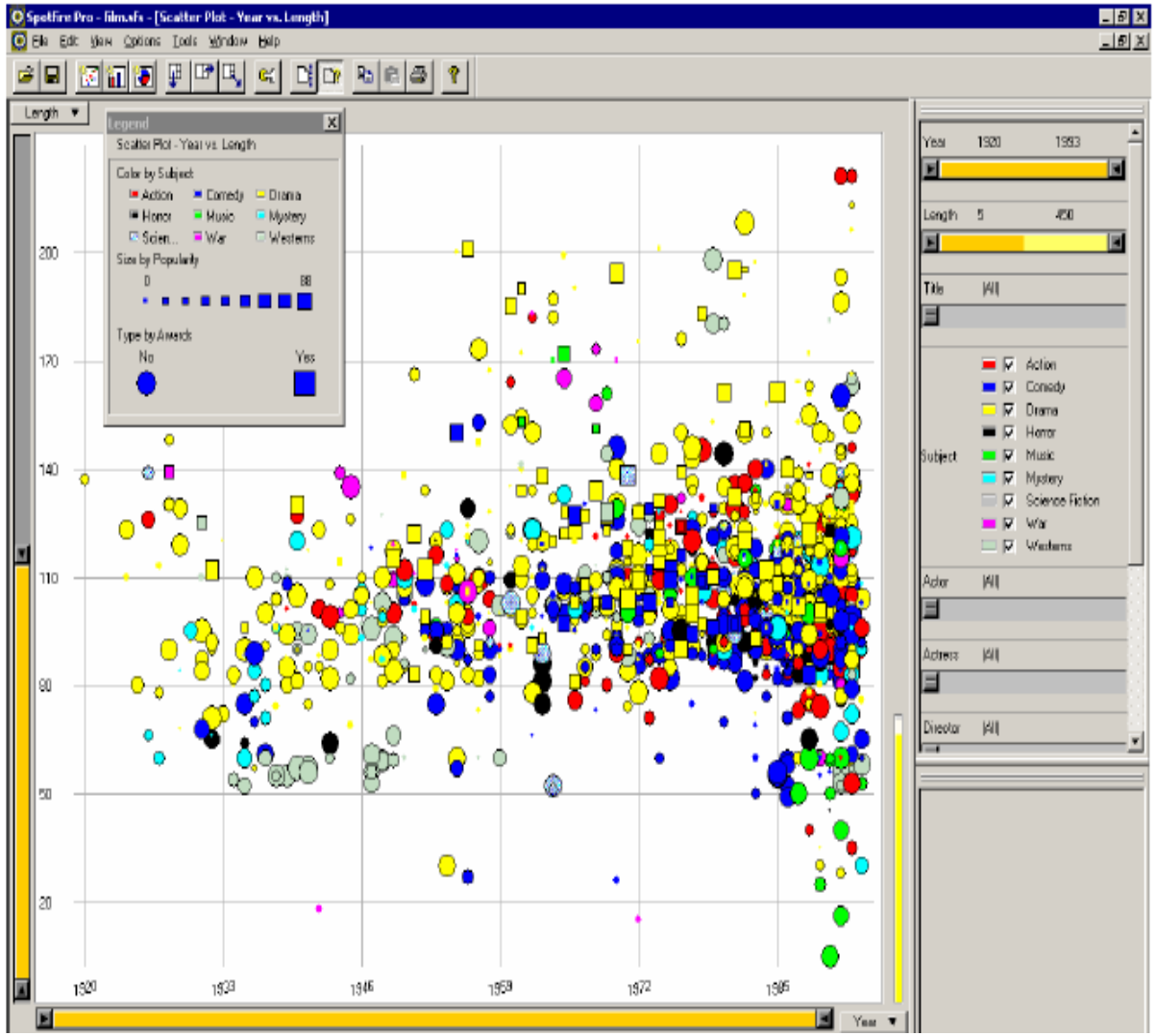
Q: Misses what?

A: multivariate effects



Bubble Graph

Spotfire

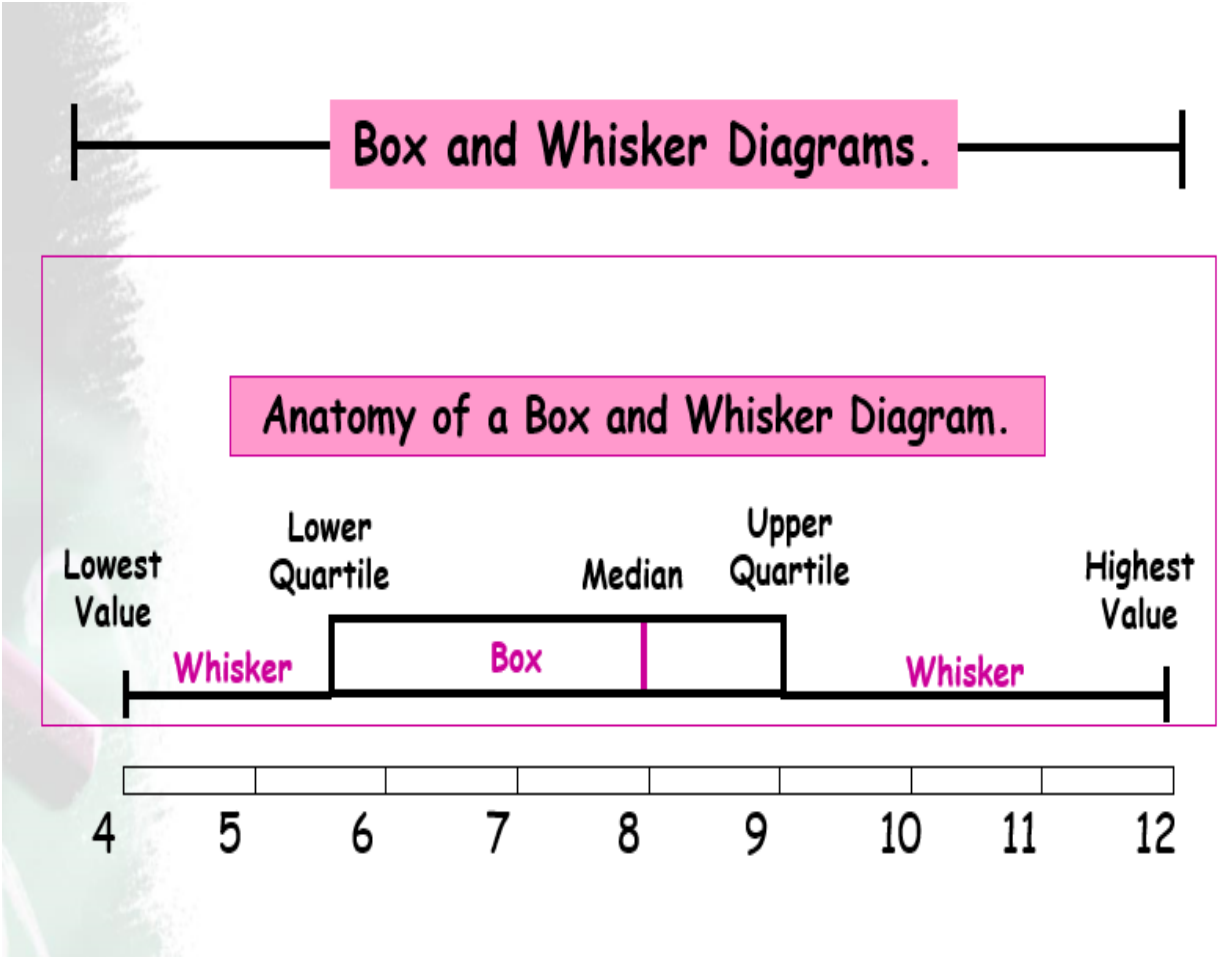


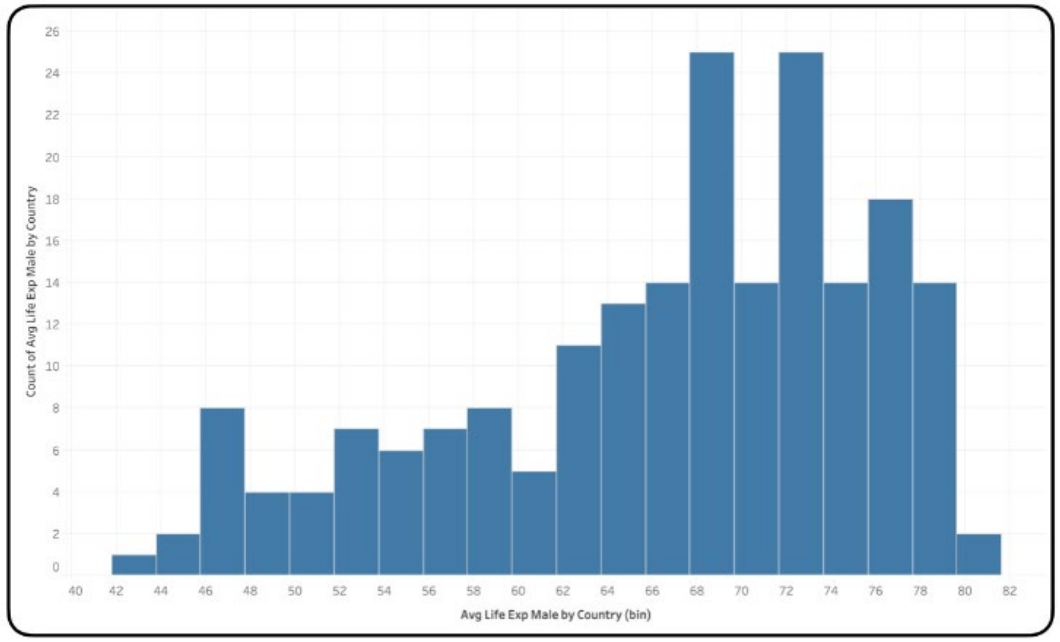
http://www.gapminder.org/

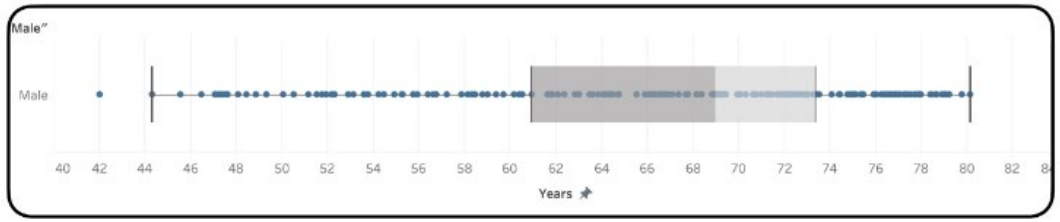
Using Bubble presentations

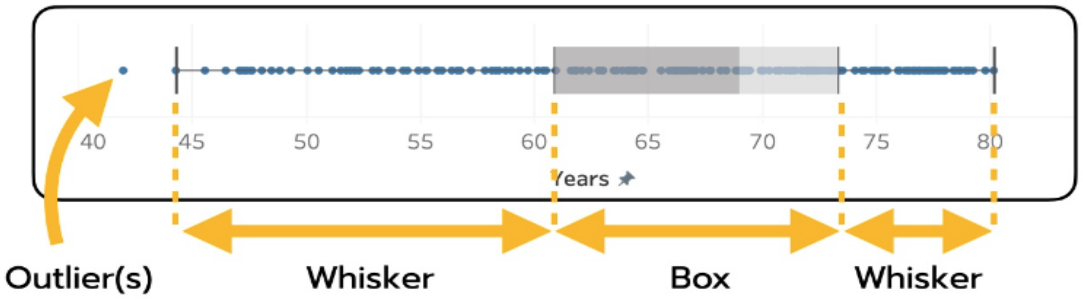
Box Plot

- A box plot summarizes data using the median, upper and lower quartiles, and the extreme (least and greatest) values.
- **Range** or spread of the data and what it means to your graph
- **Quartiles**—compare them. What are they telling you about the data?
- **Median**- this is an important part of the graph, and should be an important part of the interpretation.
- **Percentages** should be used to interpret the data, where relevant.

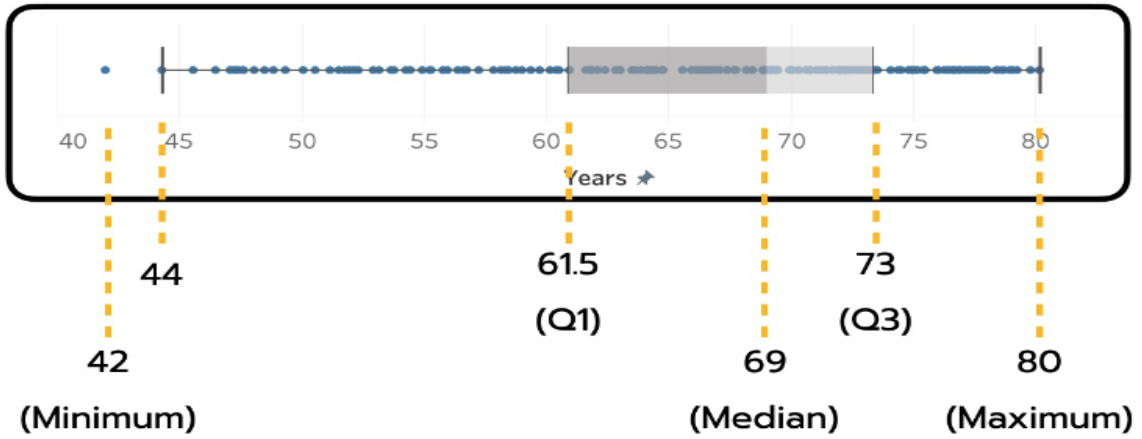








1.ค่าต่ำสุด (Minimum) = 42 years
2.Q1 (The First Quartile) = 61.5 years
3.ค่ามัธยฐาน (Median) = 69 years
4.Q3 (The Third Quartile) = 73 years
5.ค่าสูงสุด (Maximum) = 80 years

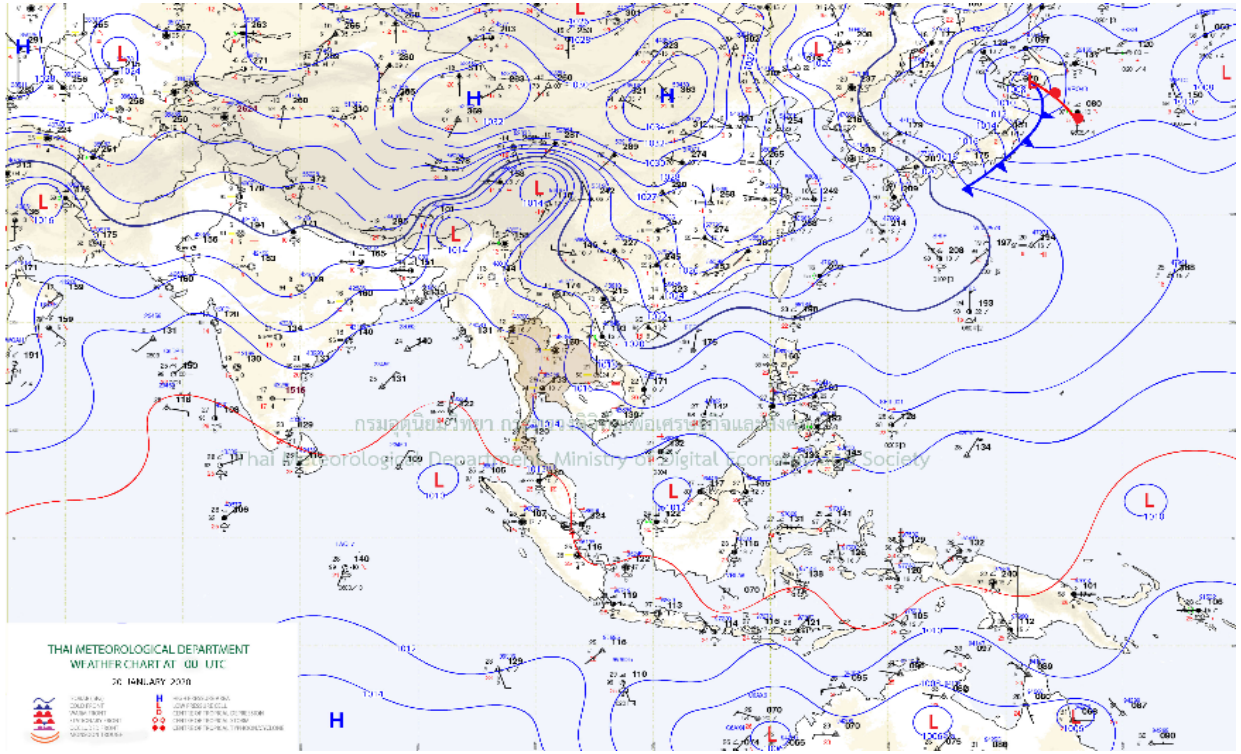


•ประชากรชายทั่วไป มีอายุขัยอยู่ที่ 69 ปี โดยวัดจากค่ามัธยฐานเป็นค่ากลาง สังเกตจากขีดกลางของกล่อง

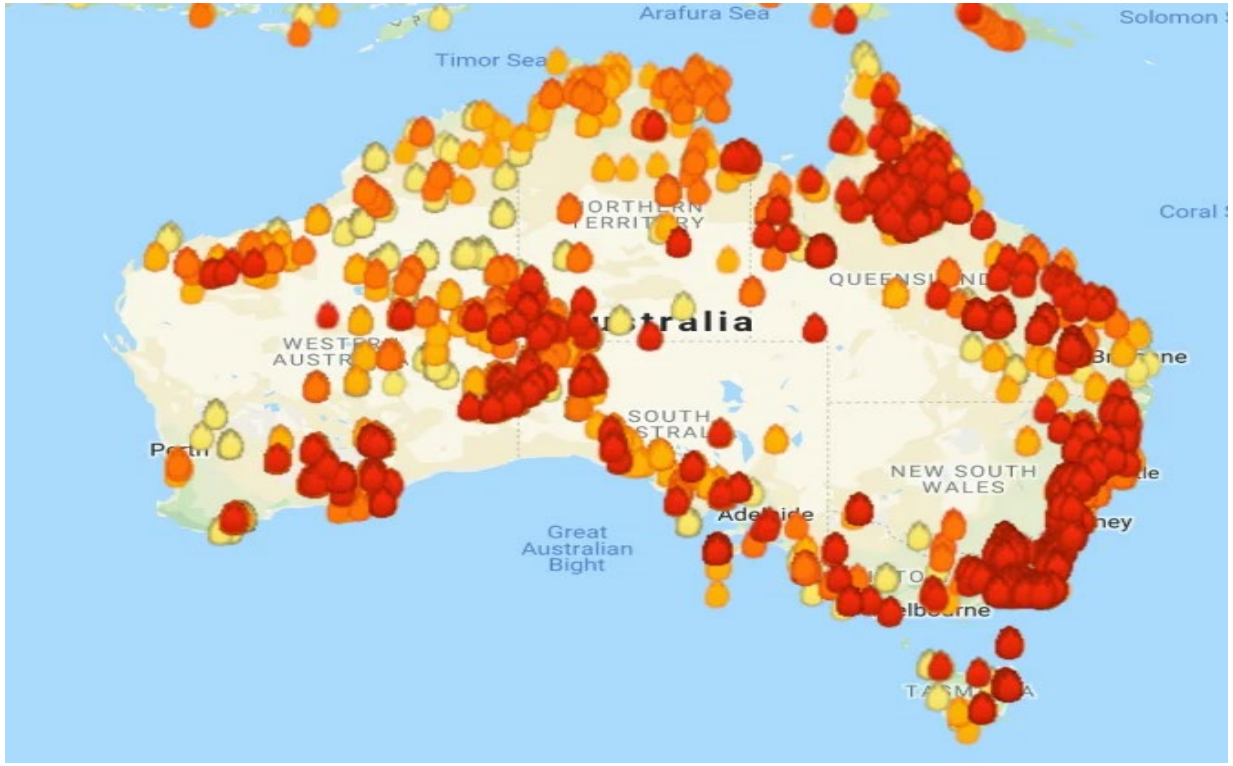
•ประชากรชายกว่า 25% มีอายุขัยไม่เกิน 61.5 ปี และประชากรชายกว่า 75% มีอายุขัยไม่เกิน 73 ปี สังเกตจาก Q1, Q3 ที่บริเวณขอบกล่องทั้งสองด้าน

•ประชากรชายส่วนใหญ่ (อย่างน้อยกึ่งหนึ่ง) มีอายุขัยเฉลี่ย อยู่ในช่วงประมาณ 61.5 – 73 ปี

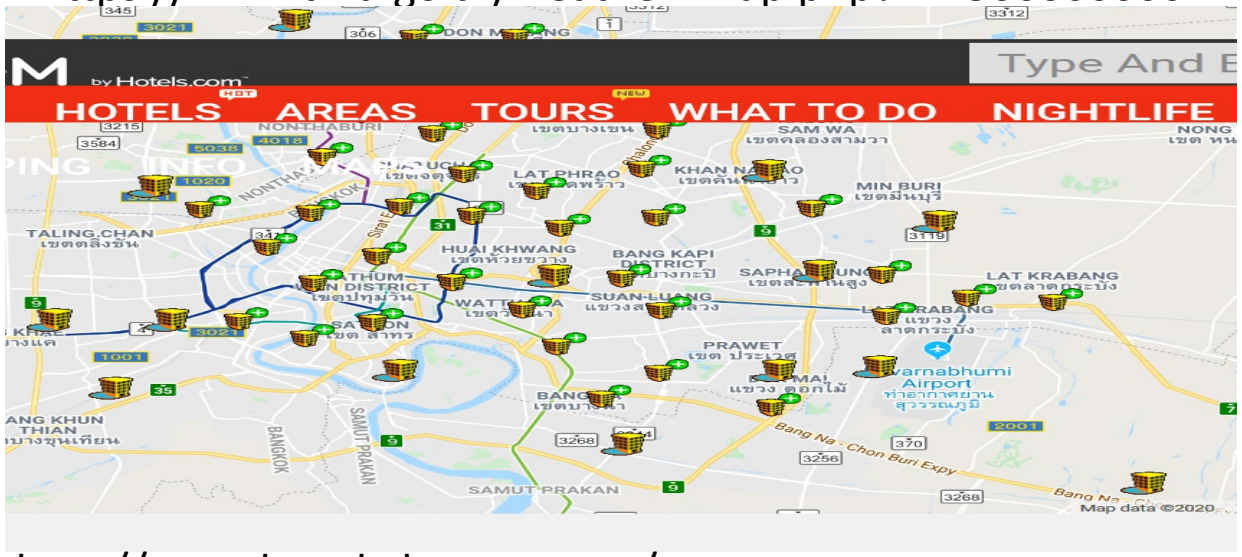
•ข้อมูลนี้ มีการกระจายตัวแบบเบ้ซ้าย (Left Skewed Distribution) เพราะข้อมูลส่วนใหญ่กระจุกตัวอยู่ทางขวามือของช่วงข้อมูลทั้งหมด ทั้งนี้ ถ้าชุดข้อมูลไม่มีการเบ้ ค่ากลางมัธยฐานควรจะอยู่ที่ประมาณ (42 + 80) / 2 = 61 ปี



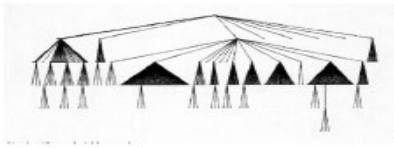
https://www.tmd.go.th/weather_map.php?v=1506060000



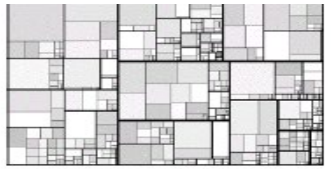
https://www.commondreams.org/news/2019/12/21/everything-burning-australian-inferno-continues-choking-access-cities-across-country



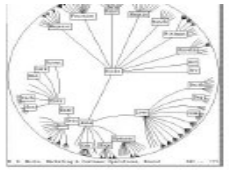
http://www.bangkok-maps.com/



Traditional

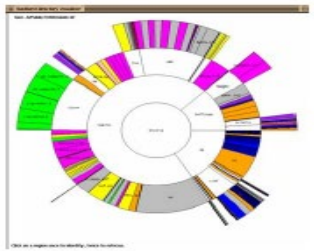


Treemap

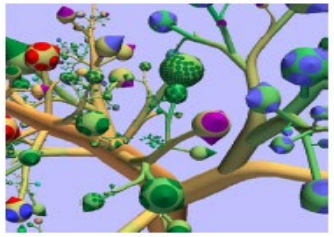


Hyperbolic Tree

ConeTree



SunTree



Botanical

Python Libraries for Data Science

matplotlib:

- python 2D plotting library which produces publication quality figures in a variety of hardcopy formats
- a set of functionalities similar to those of MATLAB
- line plots, scatter plots, barcharts, histograms, pie charts etc.
- relatively low-level; some effort needed to create advanced visualization

Link: https://matplotlib.org/
https://matplotlib.org/gallery.html

Python Libraries for Data Science

Seaborn:

- based on matplotlib
- provides high level interface for drawing attractive statistical graphics
- Similar (in style) to the popular ggplot2 library in R

Link: https://seaborn.pydata.org/

Benefits of Seaborn

- Seaborn offers:
    - Using default themes that are aesthetically pleasing.
    - Setting custom colour palettes.
    - Making attractive statistical plots.
    - Easily and flexibly displaying distributions.
    - Visualising information from matrices and DataFrames.
- The last three points have led to Seaborn becoming the exploratory data analysis tool of choice for many Python users.

Graphics to explore the data

Seaborn package is built on matplotlib but provides high level interface for drawing attractive statistical graphics, similar to ggplot2 library in R. It specifically targets statistical data visualization

To show graphs within Python notebook include inline directive:

```
In [ ]: %matplotlib inline
```

```
In [ ]:df.plot()
```

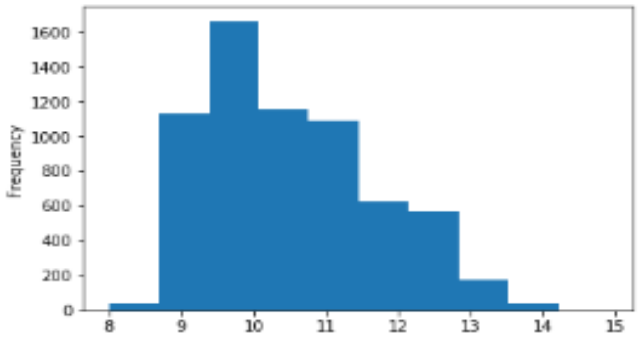
Graphics

	description
distplot	histogram
barplot	estimate of central tendency for a numeric variable
violinplot	similar to boxplot, also shows the probability density of the data
jointplot	Scatterplot
regplot	Regression plot
pairplot	Pairplot
boxplot	boxplot
swarmplot	categorical scatterplot
factorplot	General categorical plot

Histogram vs. Distplot

- Pandas histogram

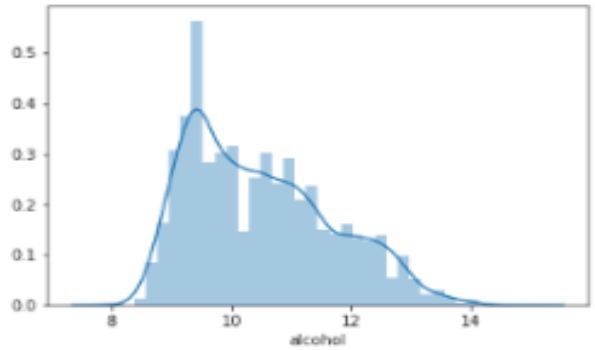
```
df['alcohol'].plot.hist()
```



  - Actual frequency of observations
  - No automatic labels
  - Wide bins

- Seaborn distplot

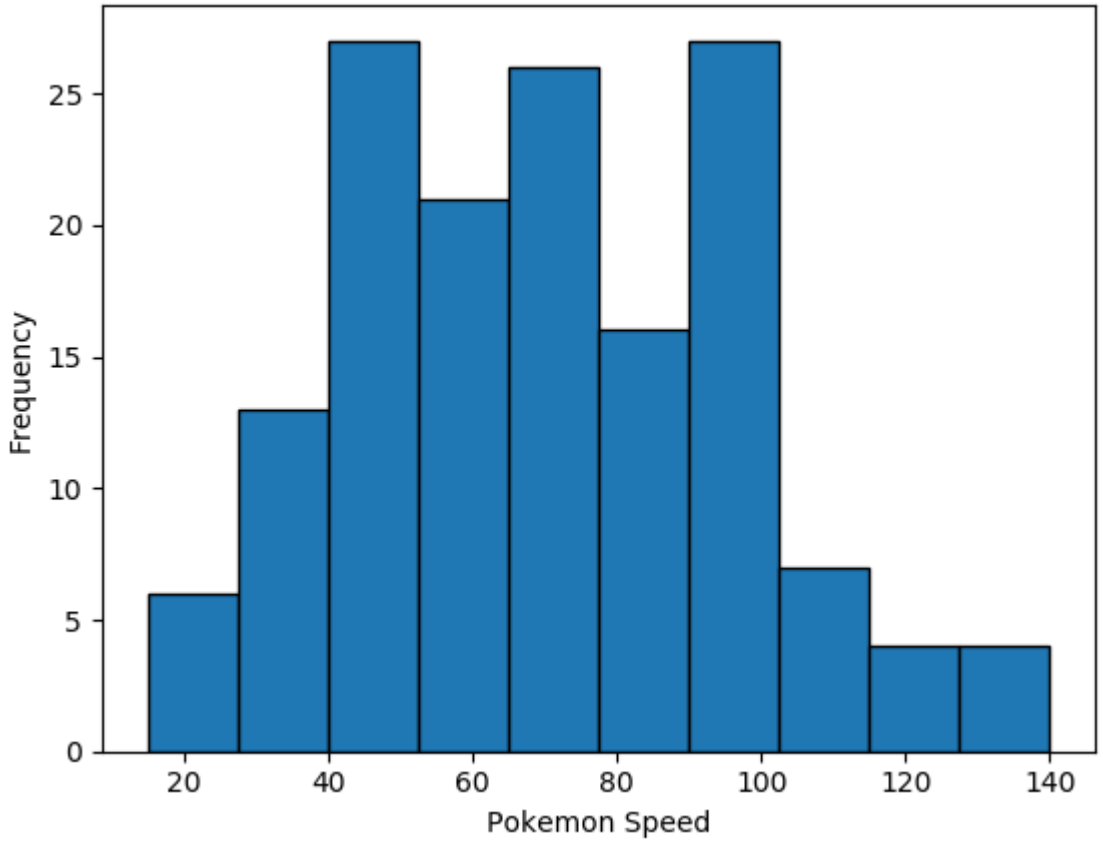
```
sns.distplot(df['alcohol'])
```



  - Automatic label on x axis
  - Muted color palette
  - KDE plot
  - Narrow bins

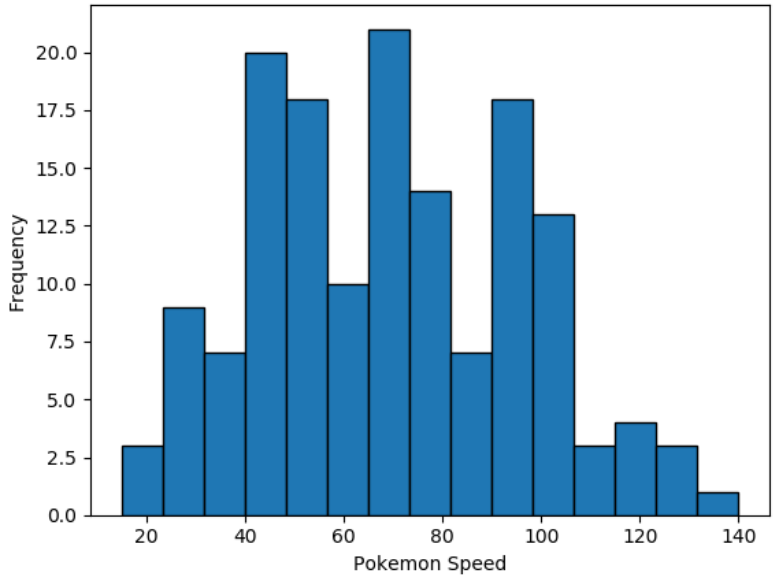
Plotting a histogram in Python

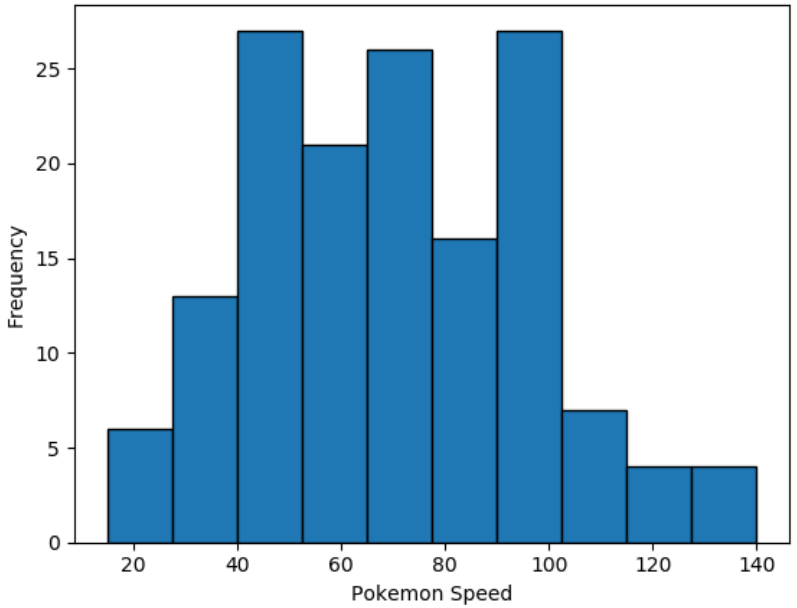
```
g = plt.hist(df1['Speed'], histtype='bar', ec='black',)
g = plt.xlabel('Pokemon Speed')
g = plt.ylabel('Frequency')
plt.show()
```



Bins

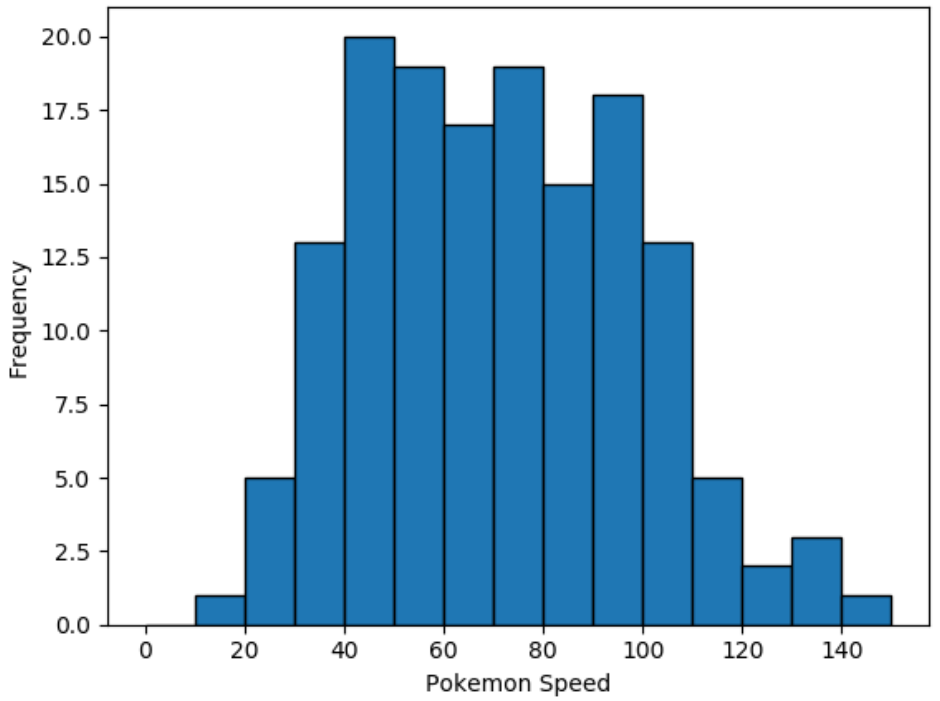
- two histograms are different, despite using the **exact** same data.
- They have different bin values.
- The left graph used the default bins generated by plt.hist(), while the one on the right used bins that was specified.





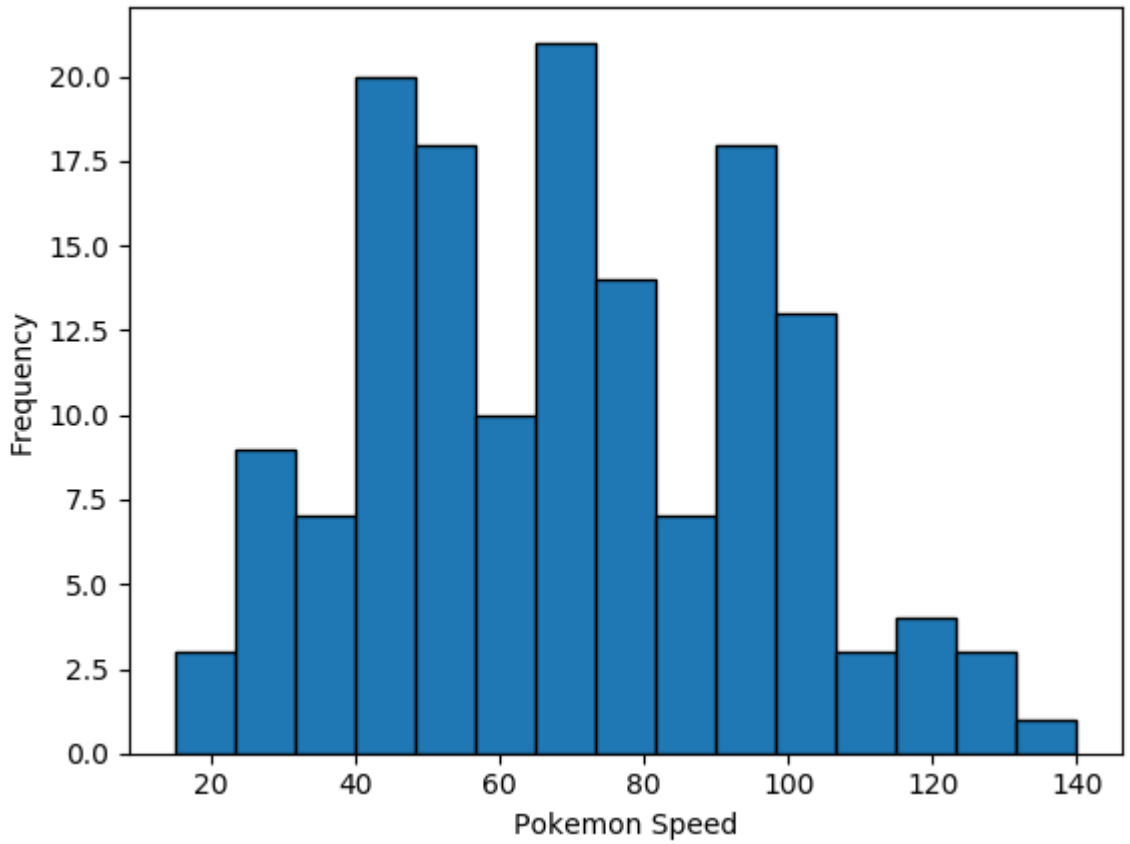
- There are a couple of ways to manipulate bins in matplotlib.
- Here, I specified where the edges of the bars of the histogram are; the bin edges.

```
bin_edges = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150]
g = plt.hist(df1['Speed'], histtype='bar', ec='black', bins=bin_edges)
g = plt.xlabel('Pokemon Speed')
g = plt.ylabel('Frequency')
plt.show()
```

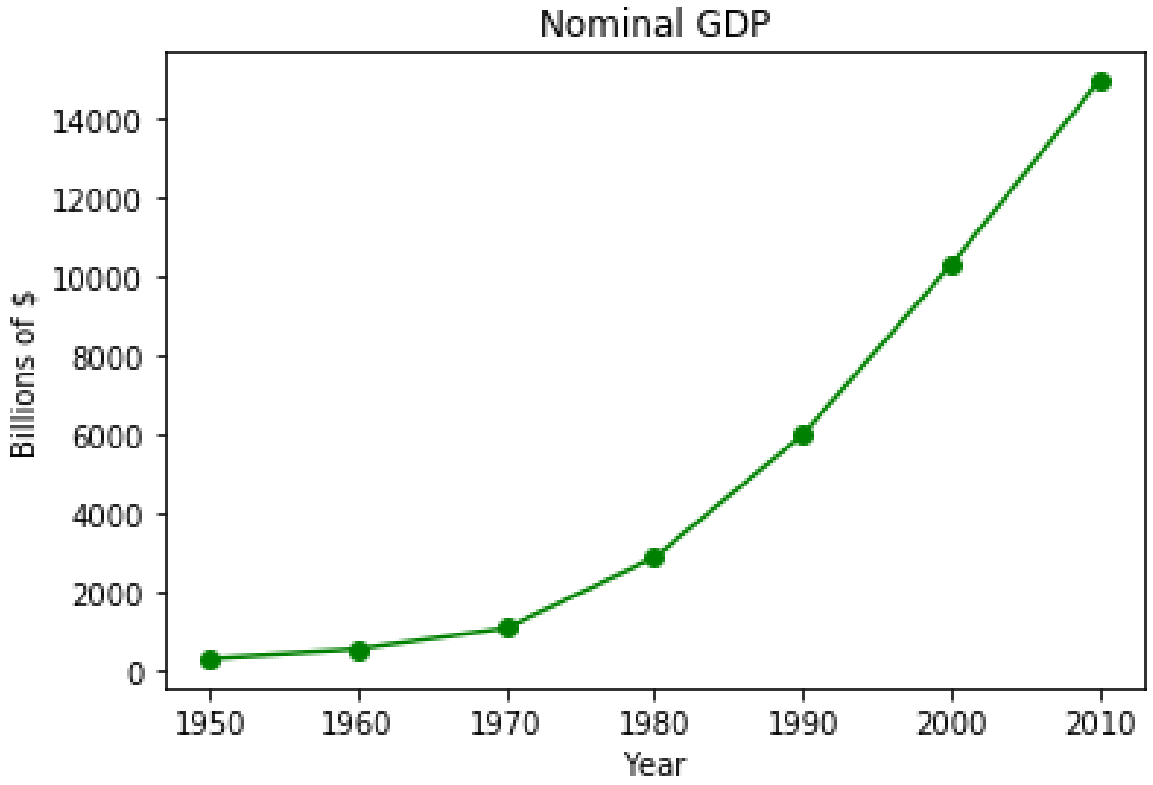


- It can be specified the number of bins, and Matplotlib will automatically generate a number of evenly spaced bins.

```
g = plt.hist(df1['Speed'], histtype='bar', ec='black', bins=15)
g = plt.xlabel('Pokemon Speed')
g = plt.ylabel('Frequency')
plt.show()
```



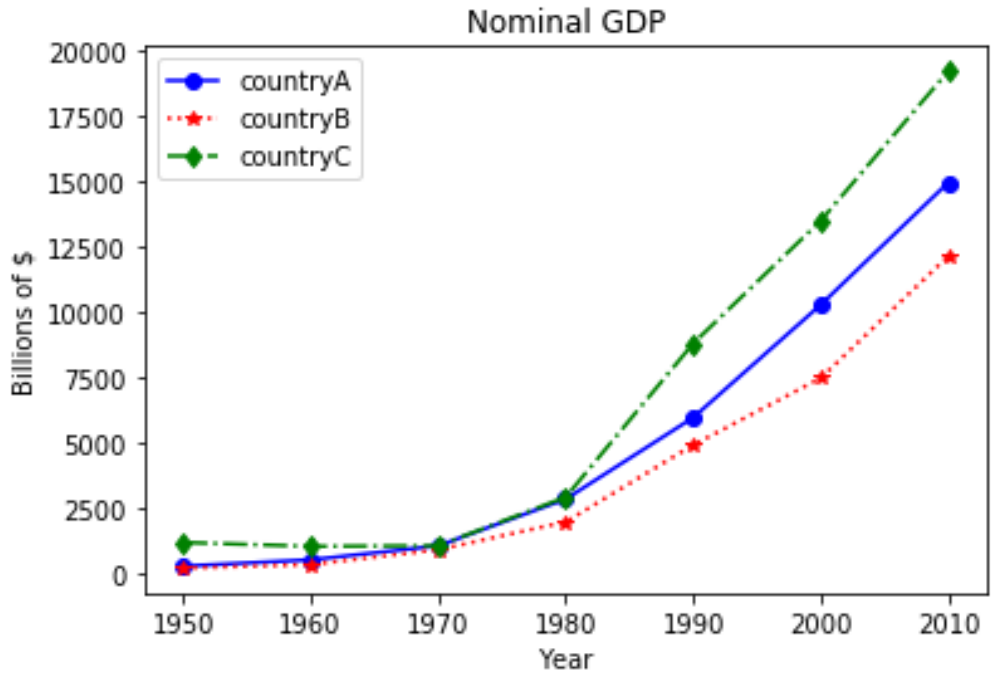
```
import matplotlib.pyplot as plt

years = list(range(1950, 2011, 10))
gdp = [300.2, 543.3, 1075.9, 2862.5, 5979.6, 10289.7, 14958.3]

# create a line chart, years on x-axis, gdp on y-axis
plt.plot(years, gdp, color='green', marker='o', linestyle='solid')

# add a title
plt.title("Nominal GDP")

# add a label to the y-axis
plt.ylabel("Billions of $")

# add a label to the x-axis
plt.xlabel("Year")
plt.show()
```



Line graph.

- Good for showing trend.
- Type plt.plot? to see more options, such as different marker and line styles, colors, etc.

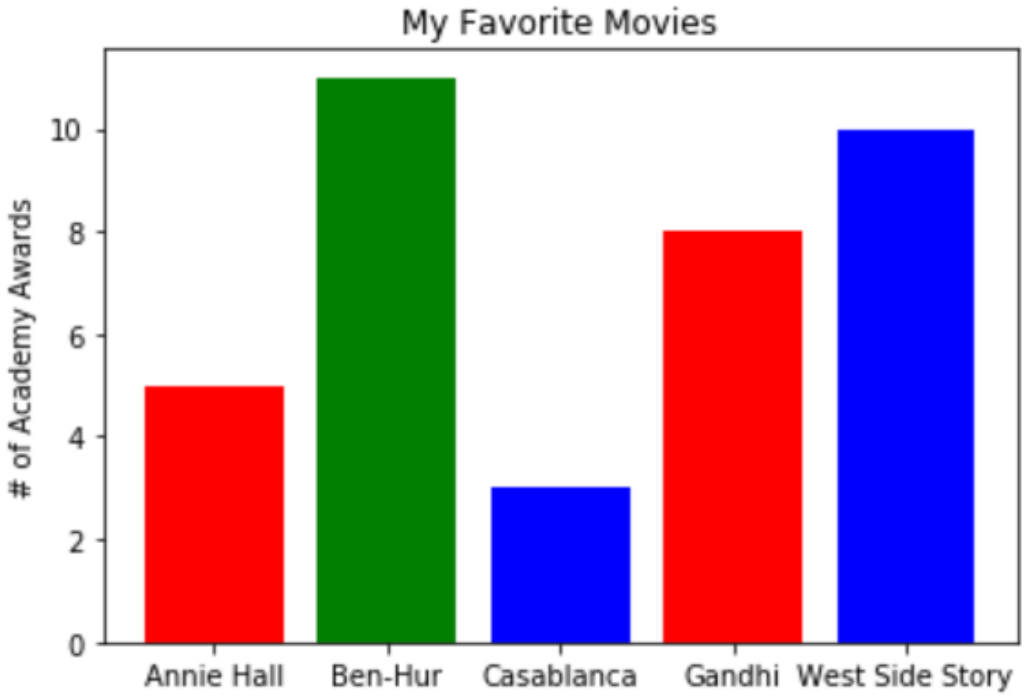
```
import matplotlib.pyplot as plt

years = list(range(1950, 2011, 10))
gdp1 = [300.2, 543.3, 1075.9, 2862.5, 5979.6, 10289.7, 14958.3]
gdp2 = [226.0, 362.0, 928.0, 1992.0, 4931.0, 7488.0, 12147.0]
gdp3 = [1206.0, 1057.0, 1081.0, 2940.0, 8813.0, 13502.0, 19218.0]

# create a line chart, years on x-axis, gdp on y-axis
# use format string to specify color, marker, and line style
# e.g. 'bo-': color='blue', marker='o', linestyle='solid'
plt.plot(years, gdp1, 'bo-',
         years, gdp2, 'r*:',
         years, gdp3, 'gd-.')
# add a title
plt.title("Nominal GDP")
# add a label to the y-axis
plt.ylabel("Billions of $")
# add a label to the x-axis
plt.xlabel("Year")
# add legend
plt.legend(['countryA', 'countryB', 'countryC'])
plt.show()
```



Bar charts (matplotlib)

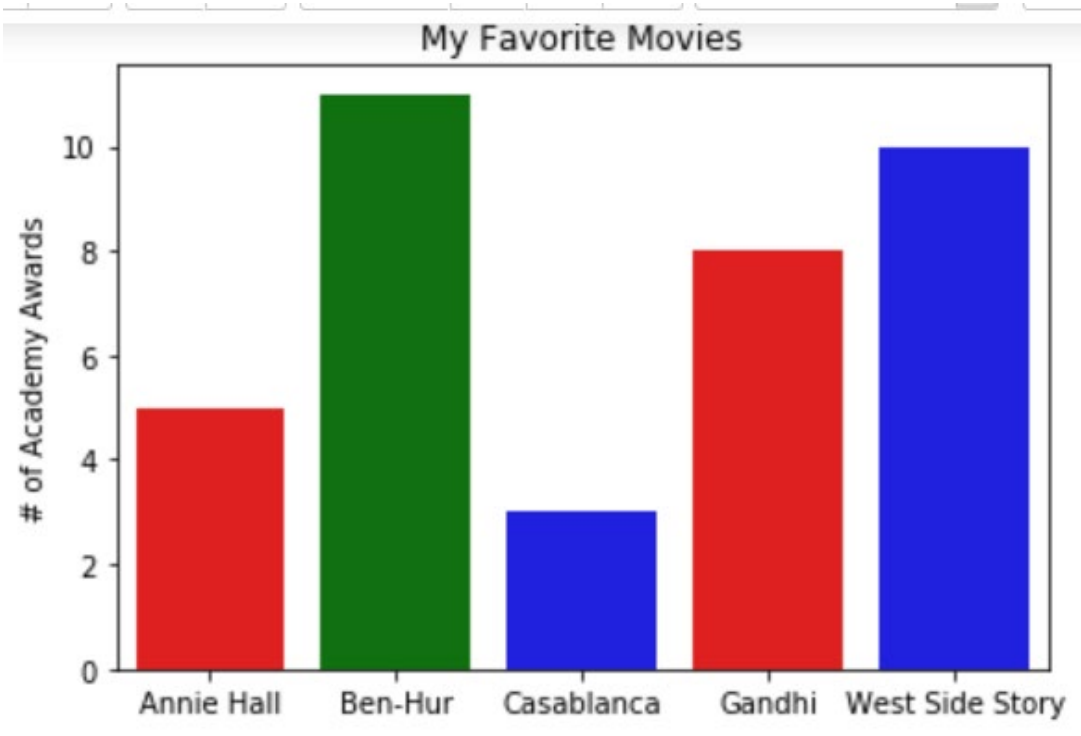
- Good for presenting/comparing numbers in discrete set of items

```
import matplotlib.pyplot as plt

movies = ["Annie Hall", "Ben-Hur", "Casablanca", "Gandhi", "West Side Story"]
num_oscars = [5, 11, 3, 8, 10]

xs = range(len(movies)) # xs is range(5)

# plot bars with left x-coordinates [xs],
# heights [num_oscars]
plt.bar(xs, num_oscars, color=('r','g','b','r','b')

# label x-axis with movie names at bar centers
plt.xticks(xs, movies)

# alternatively, use the following to replace
# the two lines above
#plt.bar(xs, num_oscars, tick_label=movies)

plt.ylabel("# of Academy Awards")
plt.title("My Favorite Movies")
plt.show()
```



Bar charts (seaborn)

```
import seaborn as sb
movies = ["Annie Hall", "Ben-Hur", "Casablanca", "Gandhi", "West Side Story"]
num_oscars = [5, 11, 3, 8, 10]

sb.barplot(movies, num_oscars,palette=('r','g','b','r','b')).set(ylabel = "# of
Academy Awards", title ='My Favorite Movies')

plt.show()
```

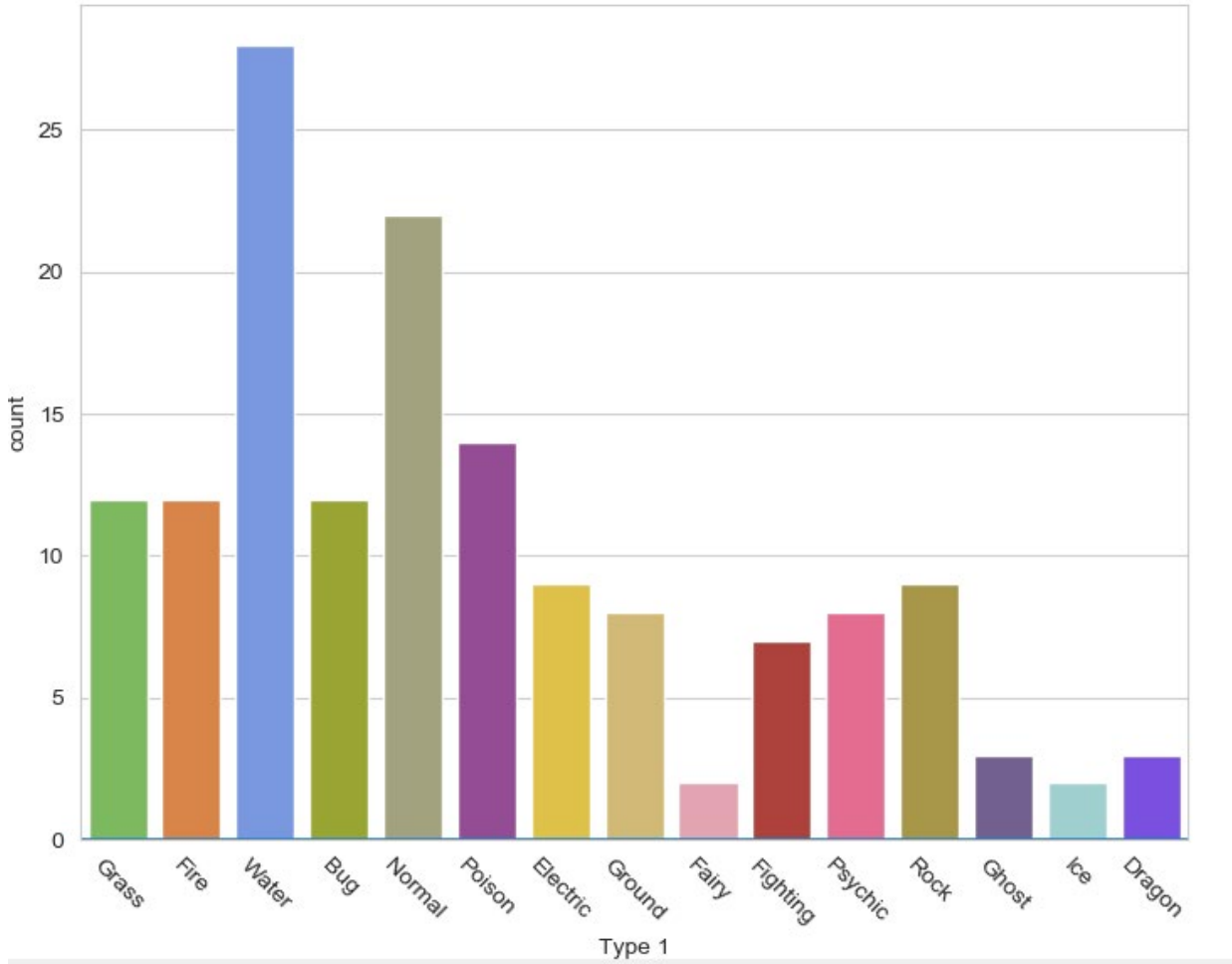


Bar plot

- Visualises the distributions of categorical variables.

```
sns.countplot(x='Type 1', data=df1,
              palette=type_colors)
plt.xticks(rotation=-45)
```

Rotates the x-ticks 45 degrees



Other types of graphs: Creating a scatter plot

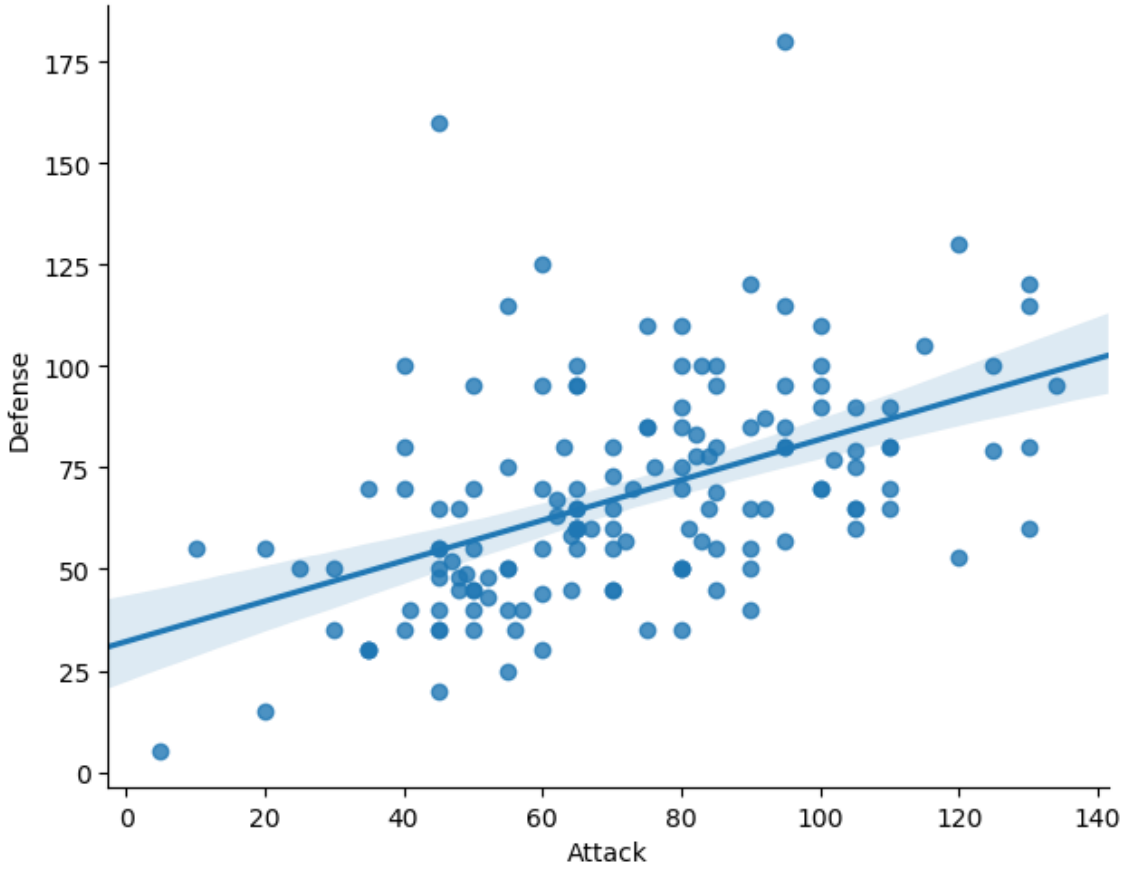
Name of variable we want on the x-axis

Name of our dataframe fed to the "data=" command

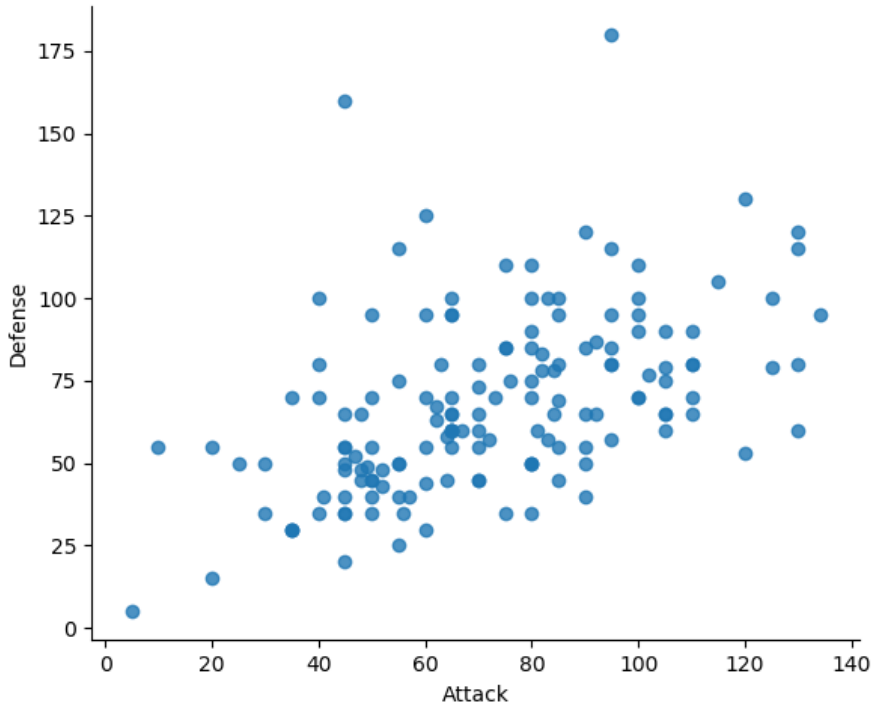
```
sns.lmplot(x='Attack', y='Defense', data=df1)
```

Seaborn "linear model plot" function for creating a scatter graph

Name of variable we want on the y-axis

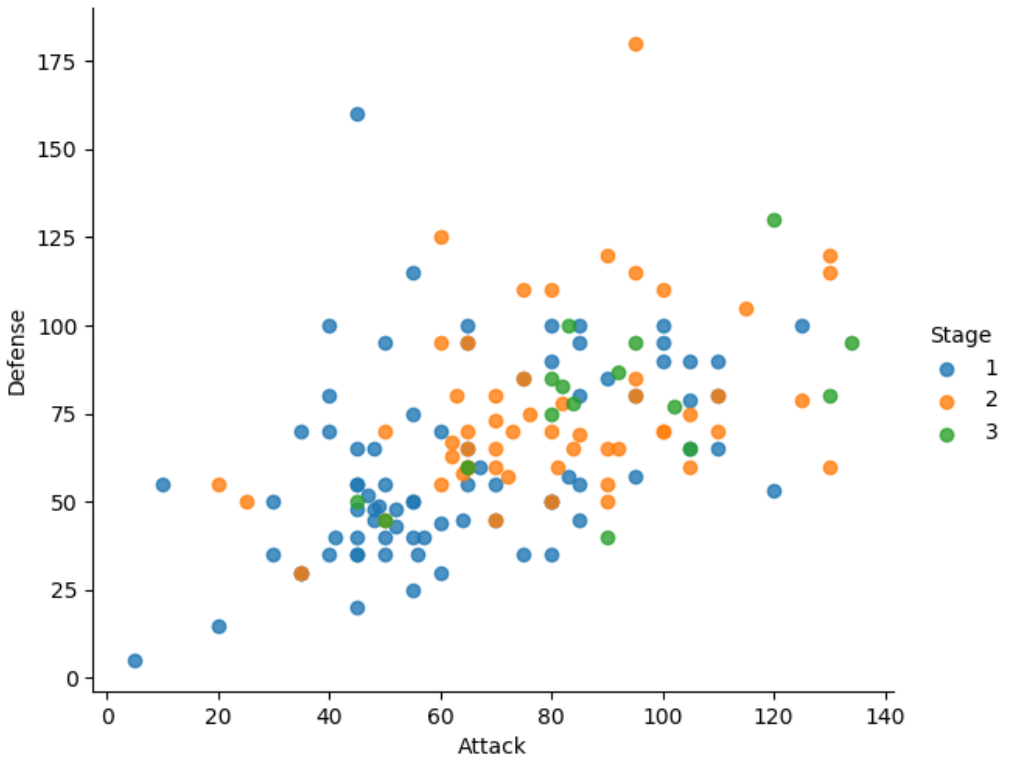


```
sns.lmplot(x='Attack', y='Defense', data=df1, fit_reg=False)
```



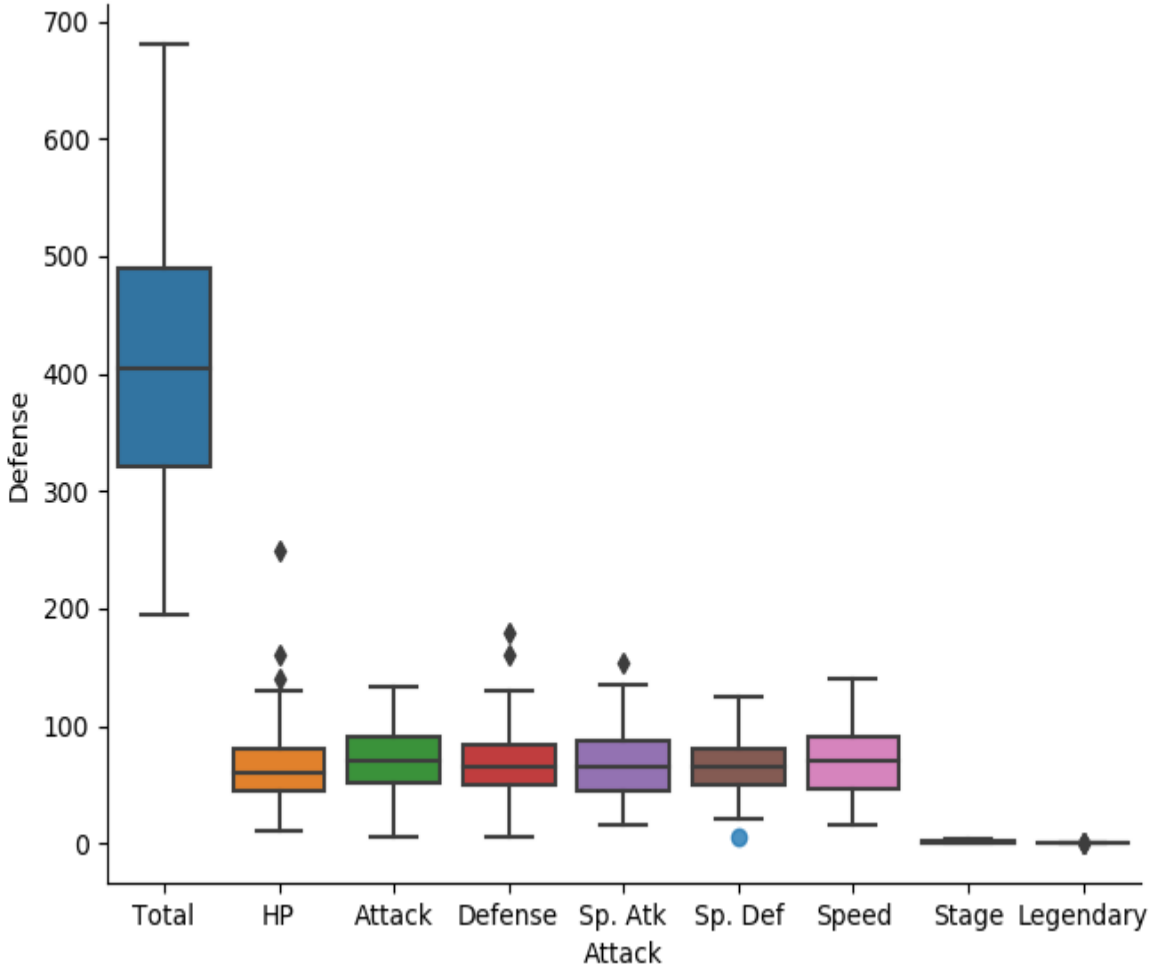
The hue function

```
sns.lmplot(x='Attack', y='Defense', data=df1,
           fit_reg=False,
           hue='Stage')
```

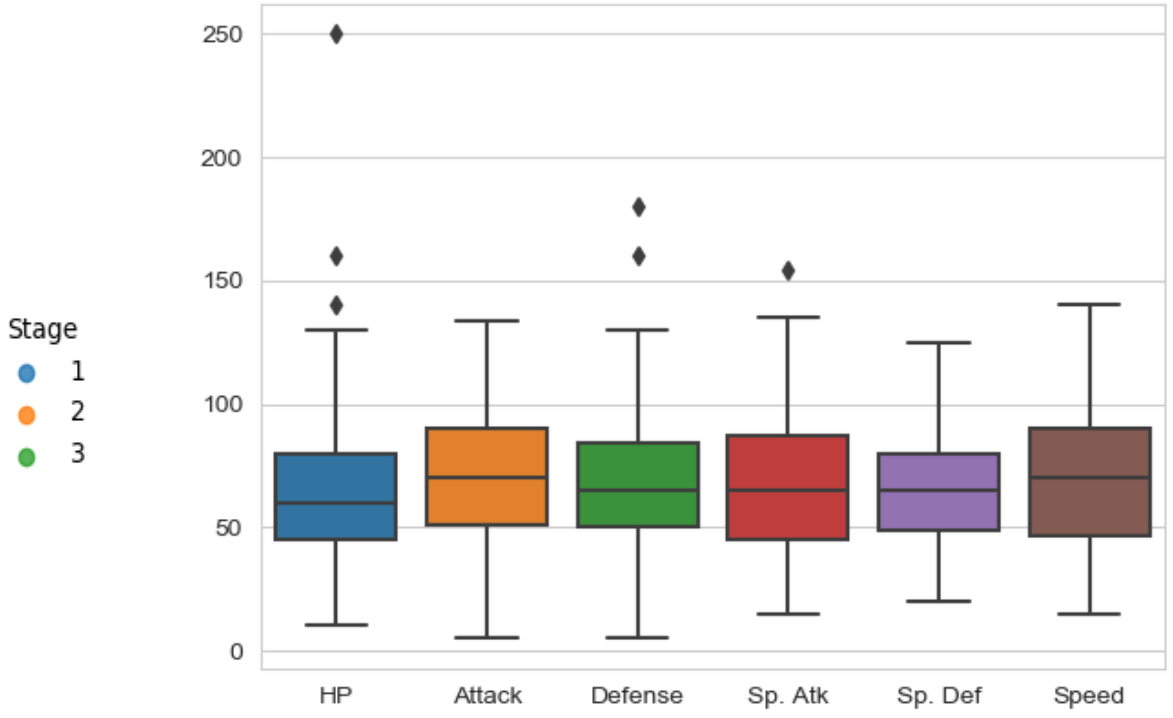


A box plot

```
sns.boxplot(data=df1)
```




```
stats_df = df1.drop(['Total', 'Stage', 'Legendary'], axis=1)
sns.set_style('whitegrid')
sns.boxplot(data=stats_df)
```

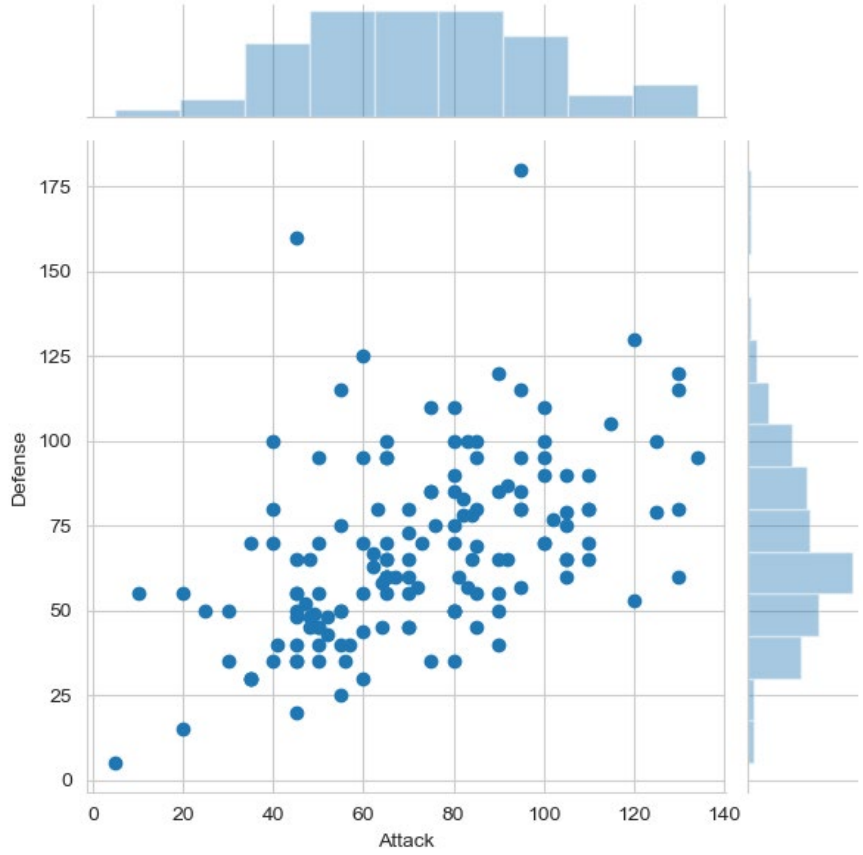


Stage
1
2
3

Joint Distribution Plot

- Joint distribution plots combine information from scatter plots and histograms to give you detailed information for bi-variate distributions.

```
sns.jointplot(x='Attack',
              y='Defense',
              data=df1)
```



Heatmaps

- Useful for visualising matrix-like data.
- Here, we'll plot the correlation of the stats_df variables

```
corr = stats_df.corr()
sns.heatmap(corr)
```