ANALISIS DAN VISUALISASI DATA

Review

Rahmatika Pratama Santi, MT

TUGAS BESAR

Bagaimana Progress?

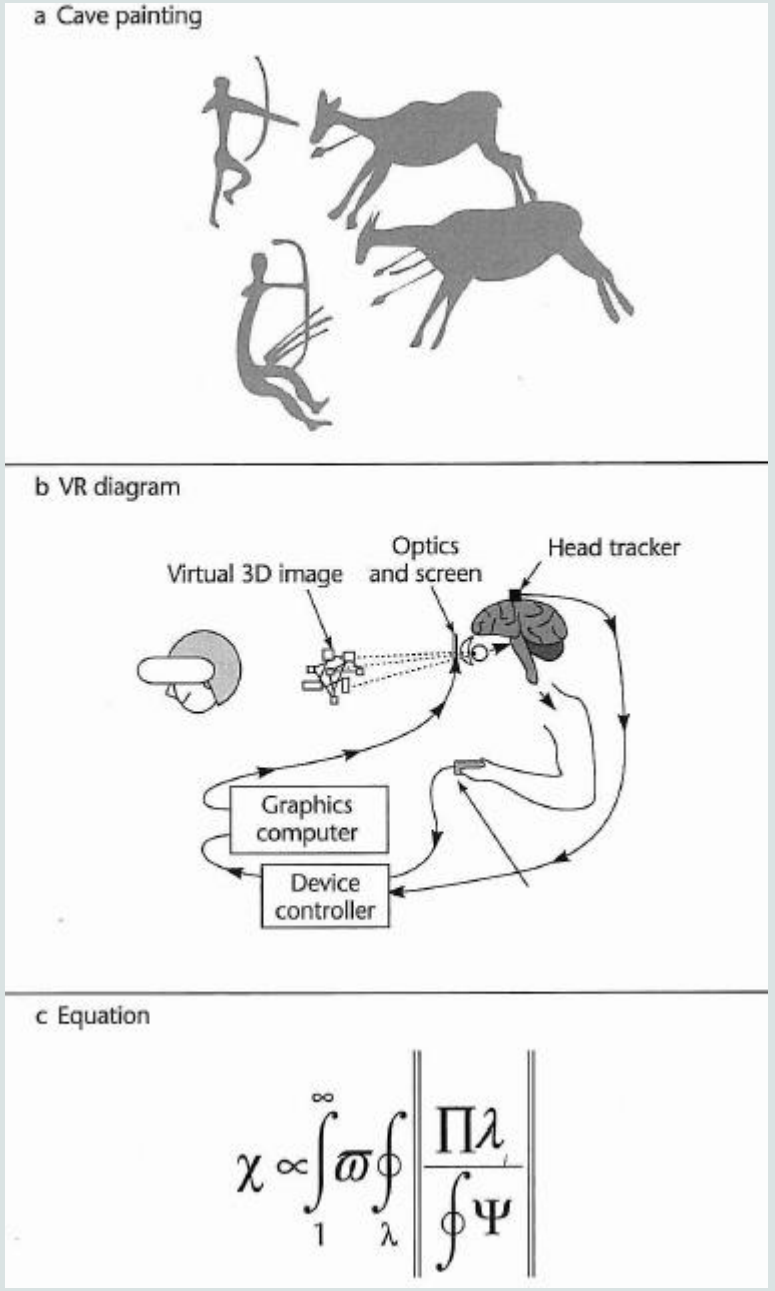
TOPIK

1) Defenisi Konsep Visualisasi Data
2) Proses Visualisasi Data
3) Metode Visualisasi
4) Pembangunan Visualisasi
5) Visual Analytic Techniques
6) Presentasi TB

Data Visualization Concept

Semiotics of Graphics

- Semiotics of Graphics : The study of symbols and how they convey meaning (a.k.a Semiology of Graphics)
  - *Sensory*
  - *Arbitrary*
- Sensory Symbols digunakan untuk merujuk pada simbol dan aspek visualisasi yang memperoleh kekuatan ekspresif mereka dari kemampuan mereka untuk menggunakan kekuatan pemrosesan perseptual otak tanpa belajar.
- Arbitrary Symbols digunakan untuk mendefinisikan aspek representasi yang harus dipelajari karena representasi tidak memiliki persepsi dasar.



Sensory vs Arbritrary Representation

ARBRITARY:

- Hard to learning
- Easy to forget
- Embedded in culture & applications
- Formally powerful
- Capable of rapid change

SENSORY:

- Understanding without training
- Resistance to instructional bias
- Sensory emmediacy
- Cross-cultural validity

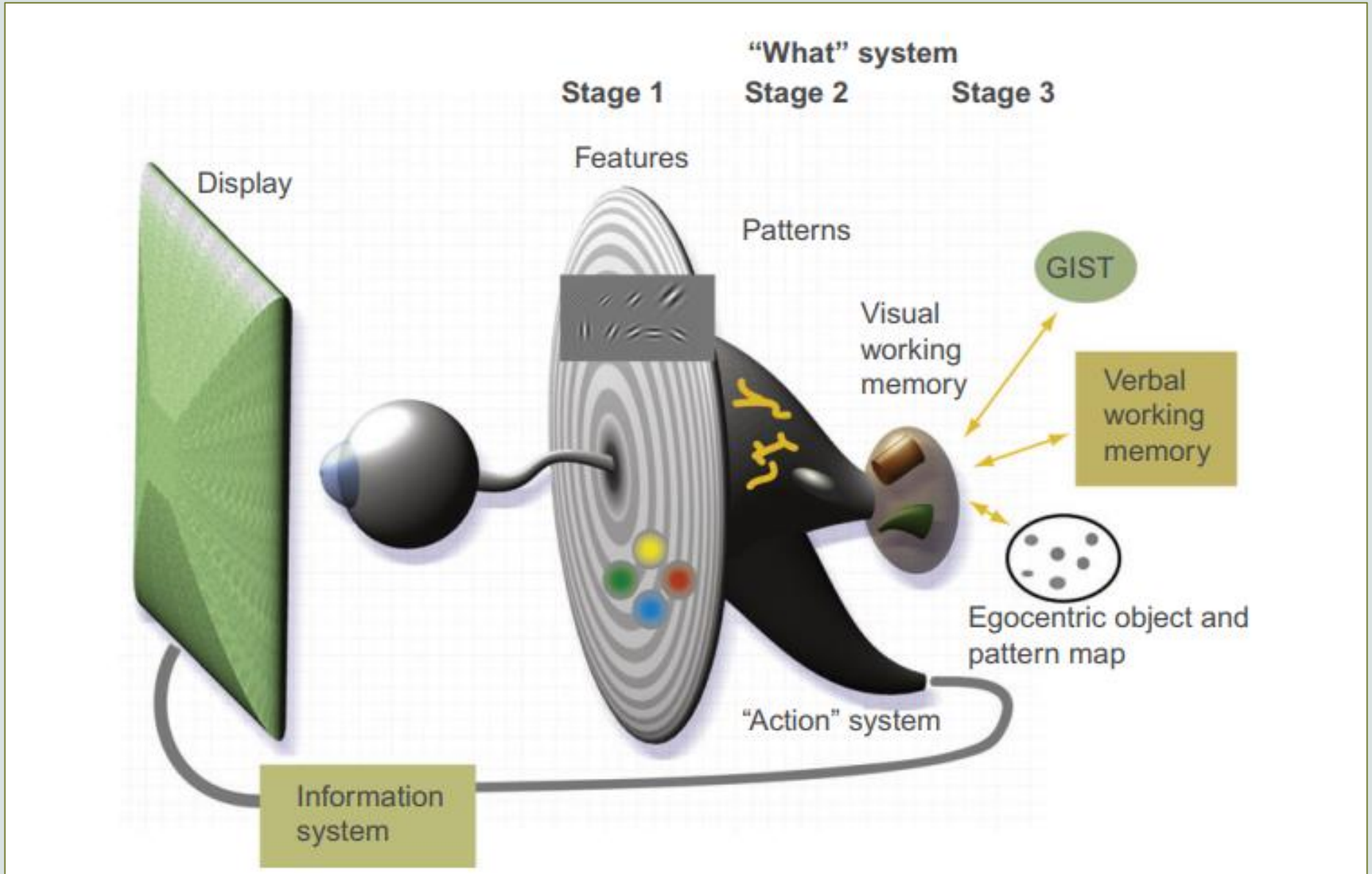
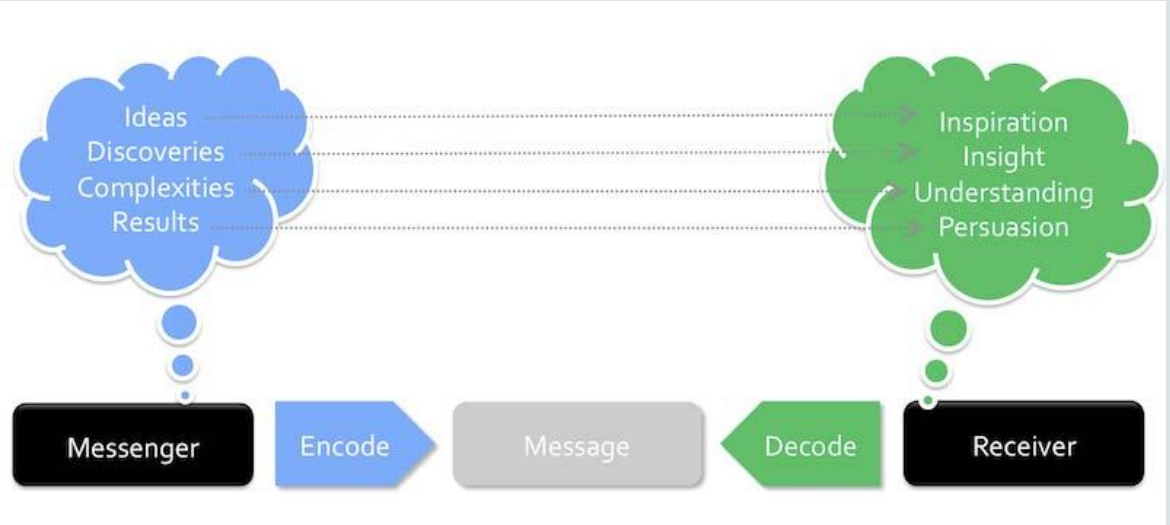


Figure 1.11 A three-stage model of visual information processing.

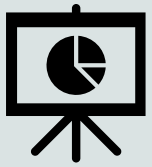
Viz. and Importance



- Menampilkan data dalam bentuk gambar/grafik, video, table, dll untuk memudahkan penyerapan informasi.
- Visualisasi sebagai media
- Messenger : a messenger looking to impart results, analysis, and stories.
  Receiver : reader/user of your visualization
  Message : the communication, like data visualization
- The important point is to ensure that our message is conveyed in the most effective and effcient form

  we need to make sure we design (or "encode") our message in a way that actively exploits how the receiver will most effectively interpret (or "decode") the message through their visual perception capabilities.

Data Visualization Concept



DEFINITION

Representation

Presentation

Visual Perception Abilities

Amplify Cognition

GOALS

Analysis

Communication

GESTALT PRINCIPLES

Proximity

Similarity

Connectedness

Continuity

Symmetry

Closure

Relative Size

Figure and Ground

FUNCTIONS

Explanatory

Exploration

Exhibition

TONE

Pragmatic or Analytic

Emotive of Abstract

1. Definition of Data Visualization

- Data Visualization → *The representation and presentation of data that exploits our visual perception abilities in order to amplify cognition.*

  *The **representation** → the way you decide to depict data through a choice of physical forms.*

  *The **presentation** → overall communicated work, including the choice of colors, annotations, and interactive features.*

  ***Visual perception abilities** → how our eyes and brains process information most effectively.*

  ***Amplify cognition** → how efficiently and effectively to process the information into thoughts, insights, and knowledge.*

- Sistem visualisasi berbasis komputer

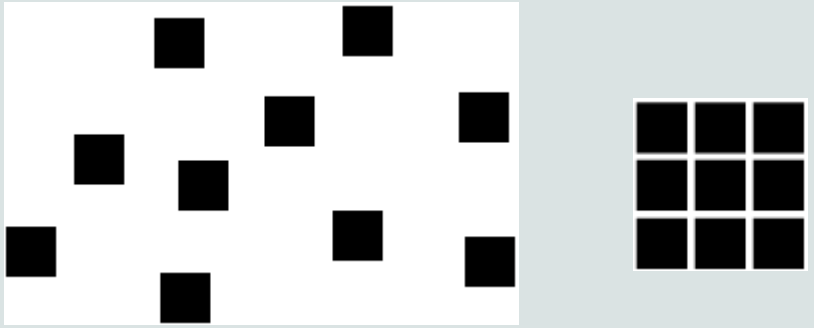
  Representasi visual dari set data yang dirancang untuk membantu orang melaksanakan tugas dengan lebih efektif.

2. Gestalt Principles

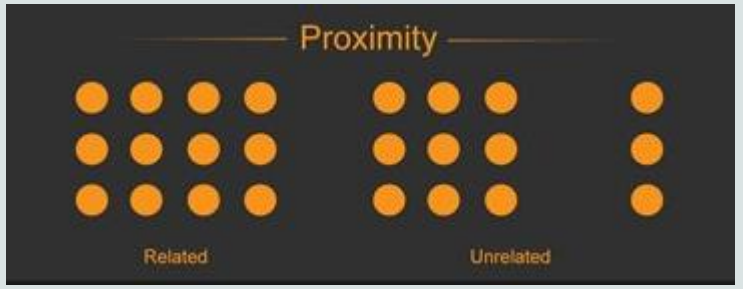
- These theories attempt to describe how people tend to organize visual elements into **groups** or *unified wholes* when certain principles are applied.
- Gestalt Laws: Proximity, similarity, connectedness, continuity, symmetry, closure, relative size, and figure-and-ground

Proximity

- *Proximity* occurs when elements are placed close together. They tend to be perceived as a group.



(a)



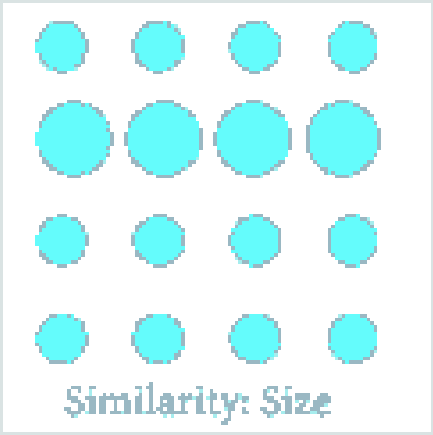
(b)

Similarity

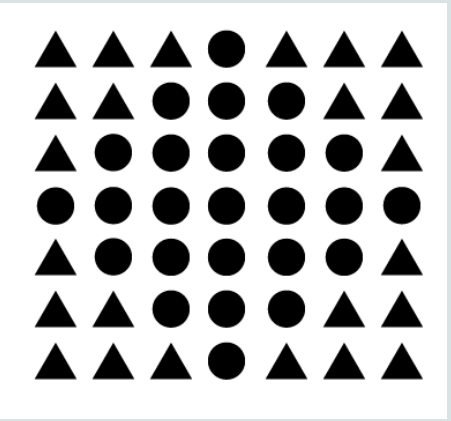
- *Similarity* occurs when **objects look similar** to one another. People often perceive them as a group or pattern.

(a)

(b)



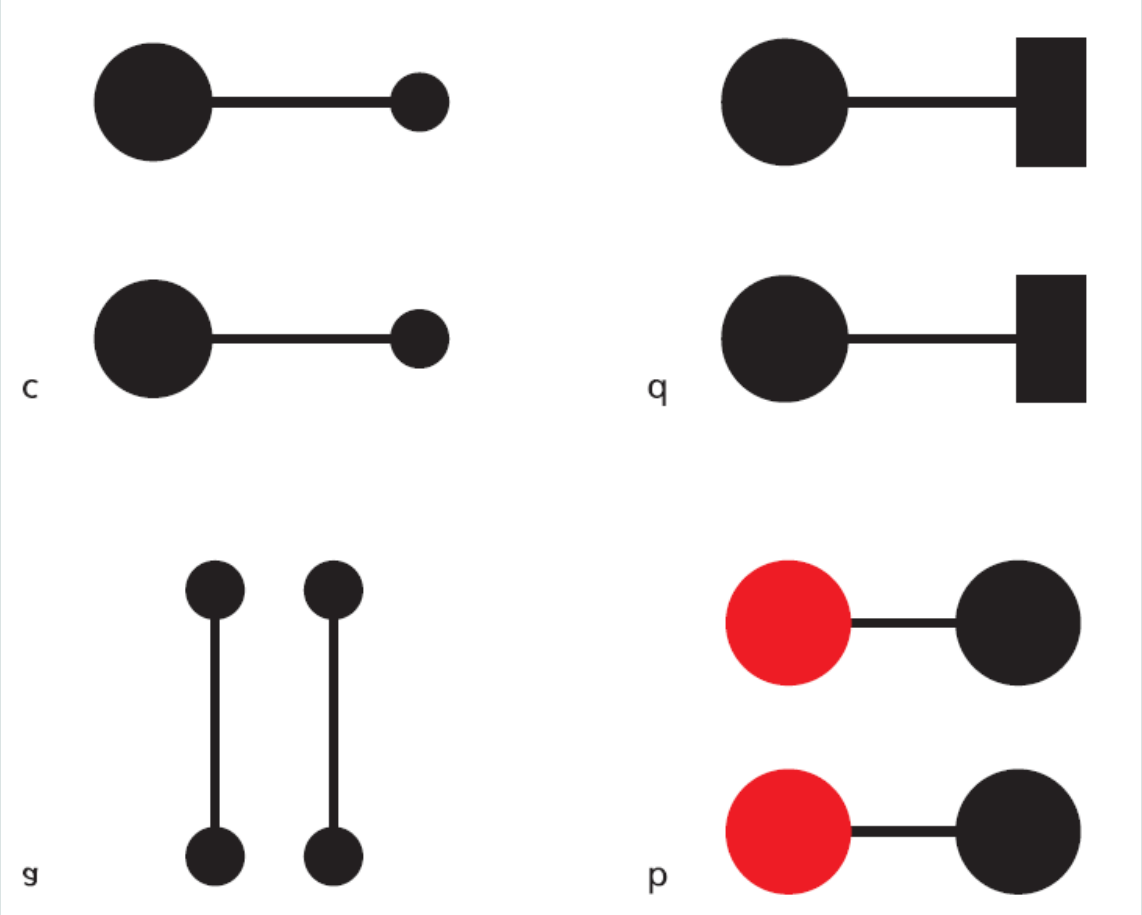
(c)



(d)

Connectedness

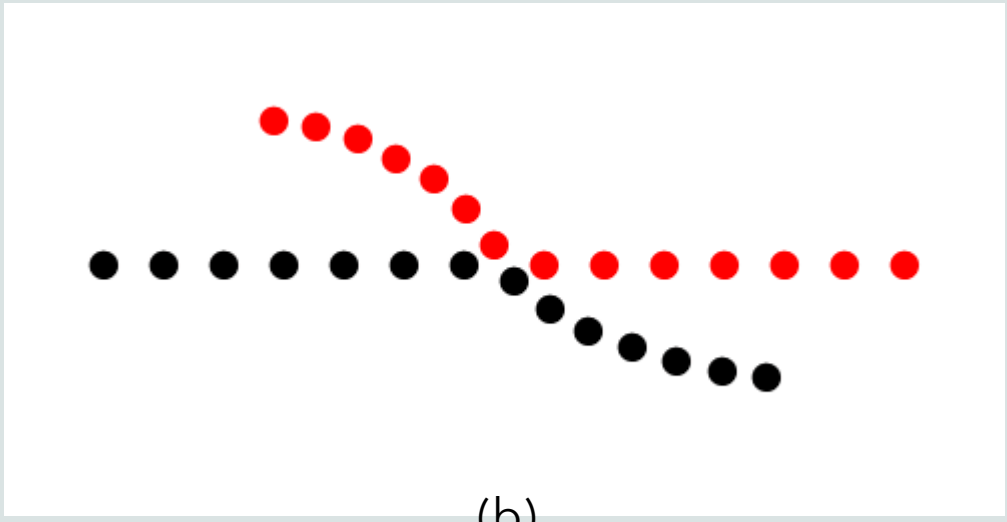
The law of unified connectedness states that elements that are connected to each other using colors, lines, frames, or other shapes are perceived as a single unit when compared with other elements that are not linked in the same manner



Continuity

- Continuation occurs when the eye is compelled to **move through** one object and **continue** to another object.
- The law of continuation asserts that the human eye follows lines, curves, or a sequence of shapes in order to determine a relationship between design elements.

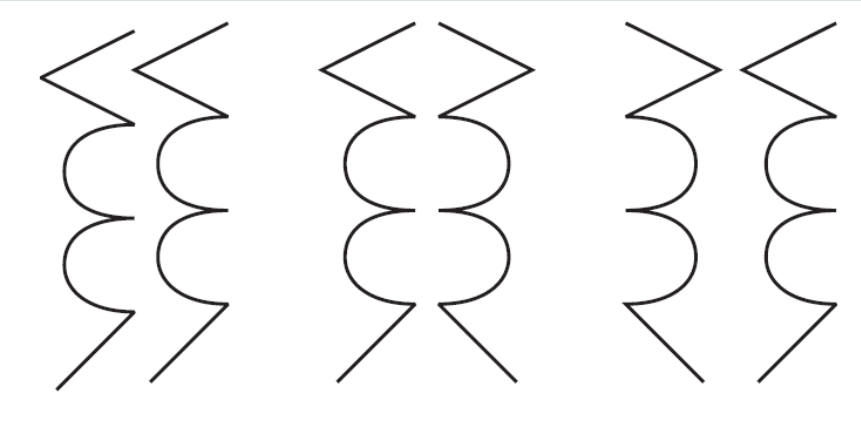
(a)

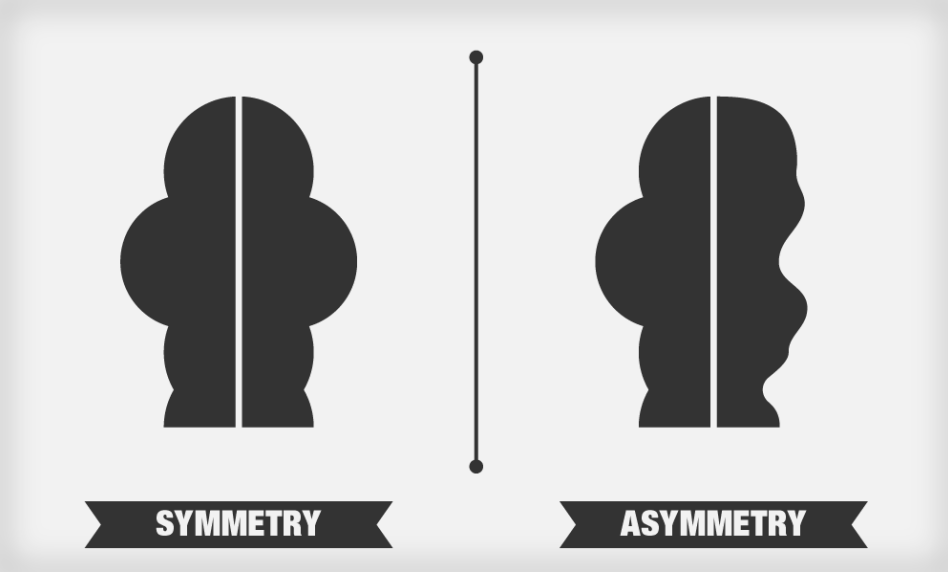


(b)

Symmetry

- Symmetrical elements are perceived as part of the same group.
- Have you ever looked at figures that look like mirror reflections of each other? This relationship helps us perceive these elements as a single figure.

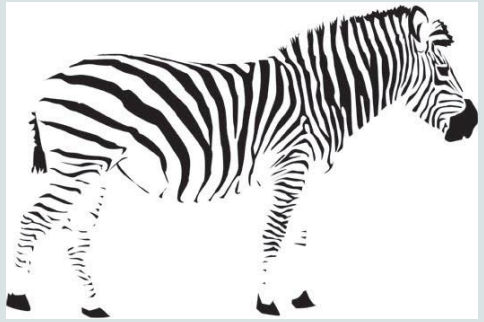




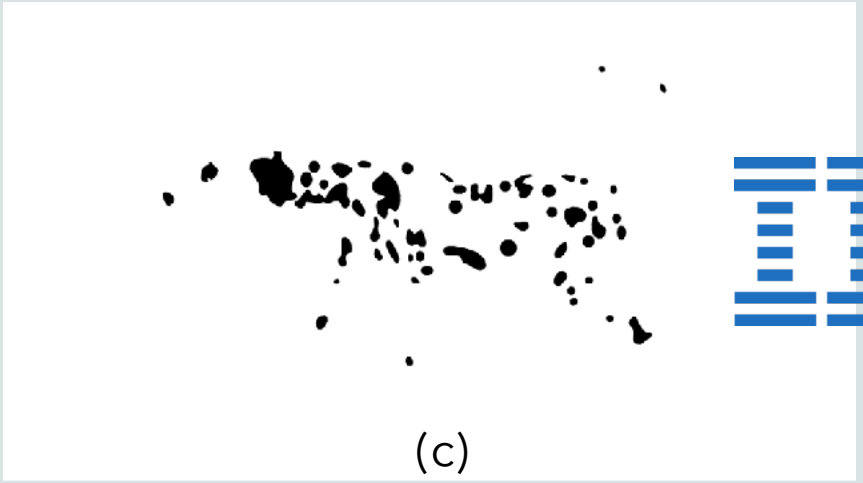
Closure

- *Closure* occurs when an object is *incomplete* or a space is not *completely enclosed*. If enough of the shape is indicated, people percieve the whole by filling in the missing infomation

(a)



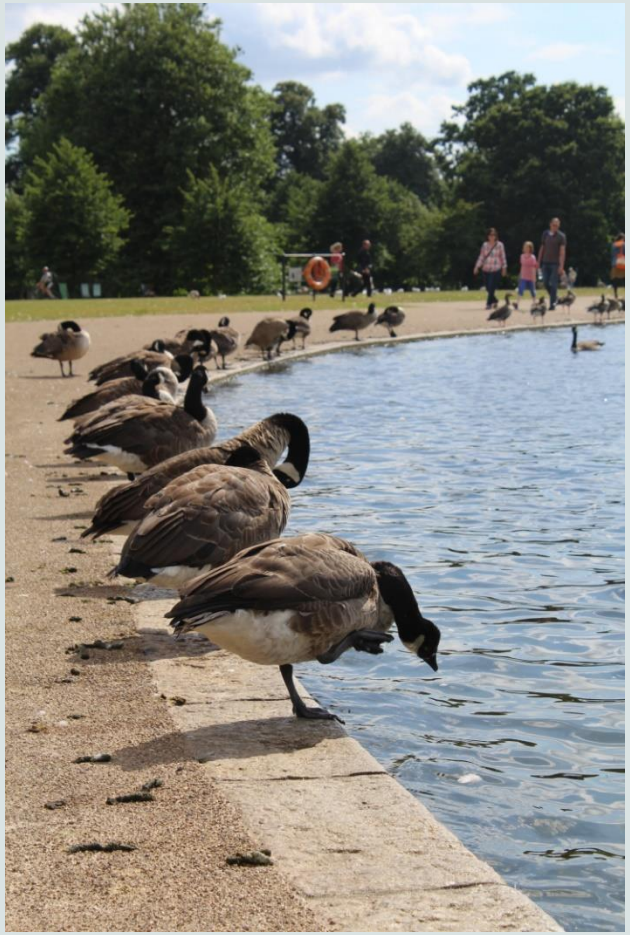
(b)

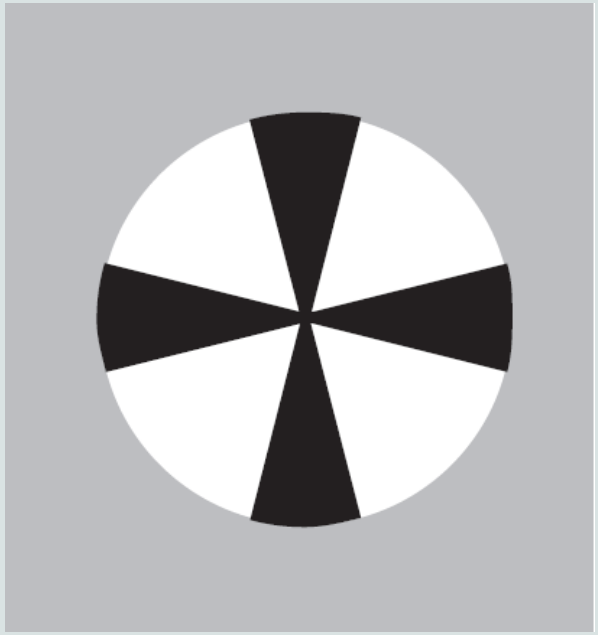


(c)

(d)

Relative Size





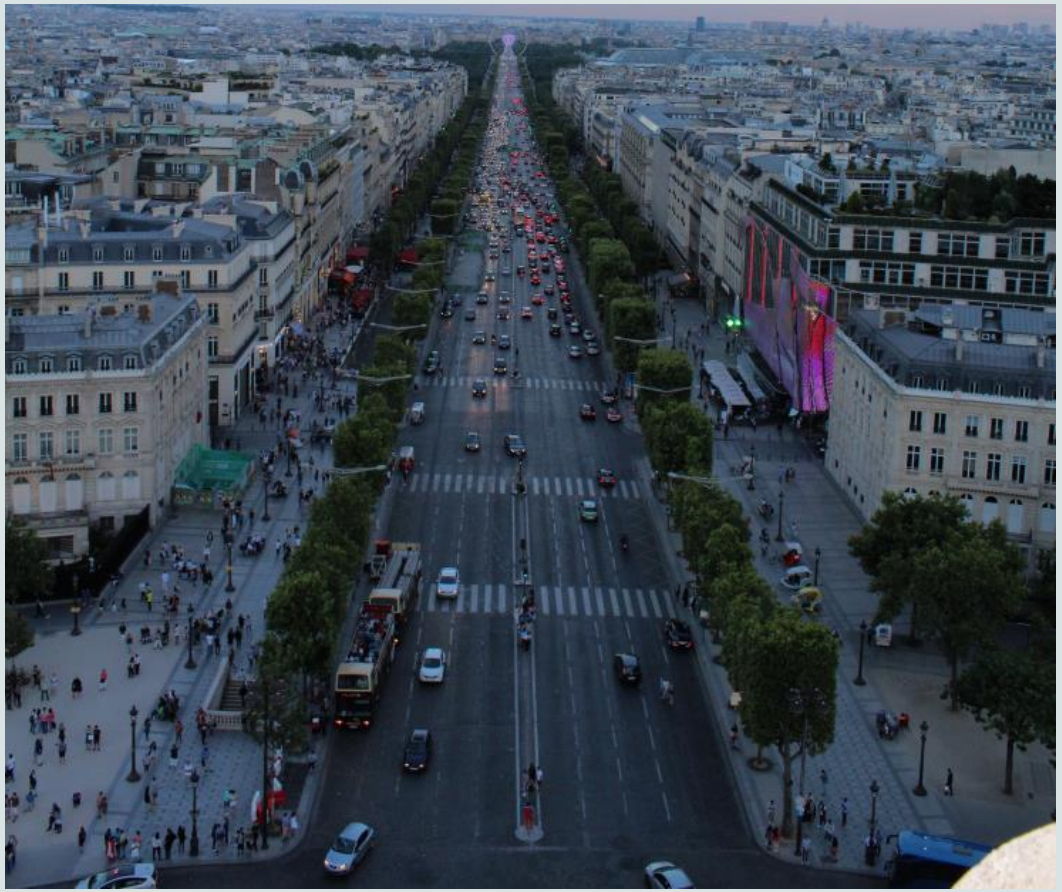
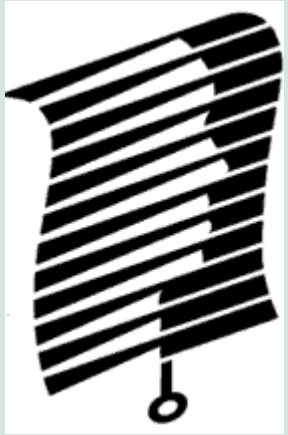


Figure and Ground

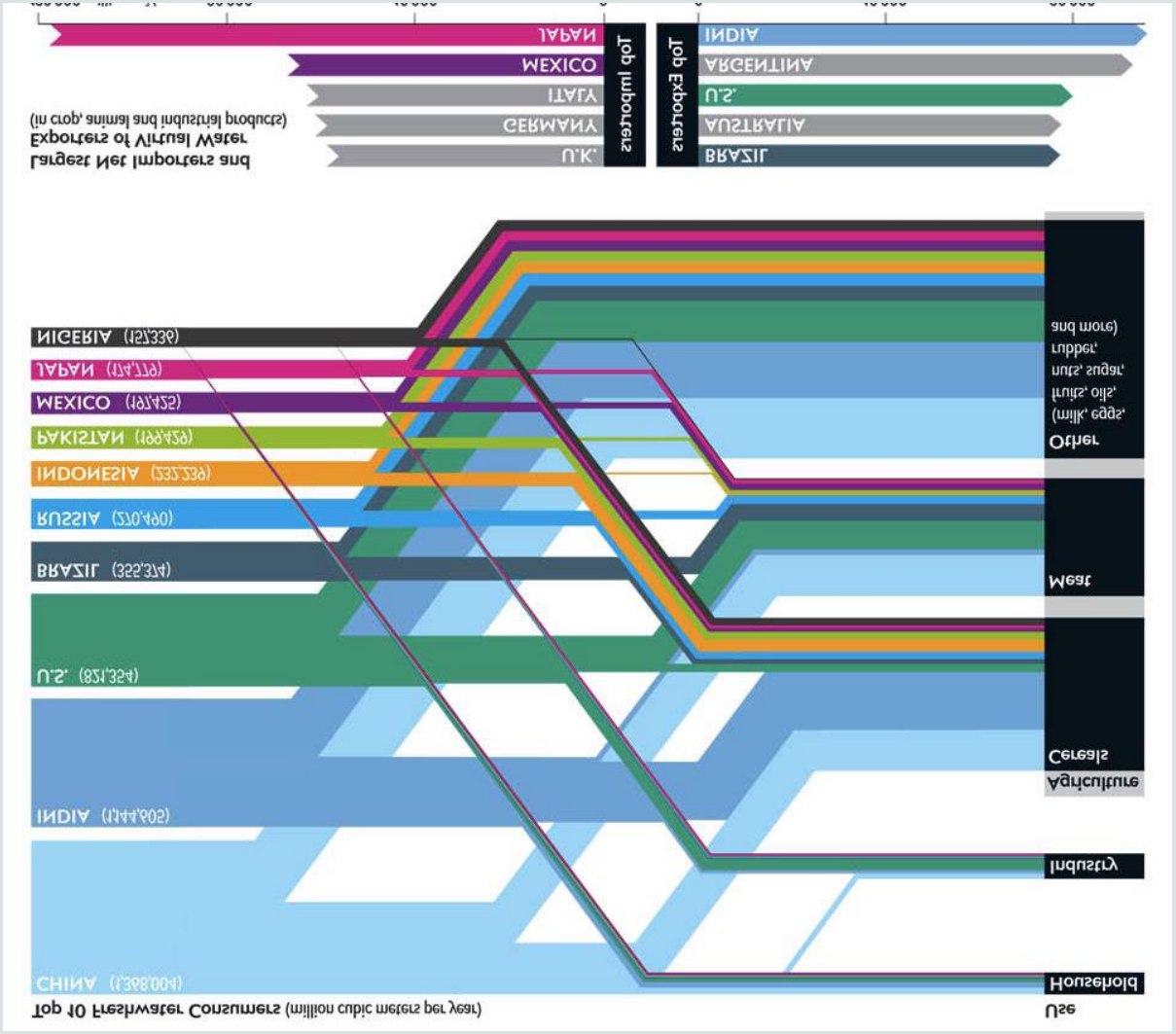
- The eye differentiates an object from its surrounding area. a form, silhouette, or shape is naturally perceived as **figure** (object), while the surrounding area is perceived as **ground** (background).
- Balancing figure and ground can make the perceived image more clear. Using unusual figure/ground relationships can add interest and subtlety to an image.



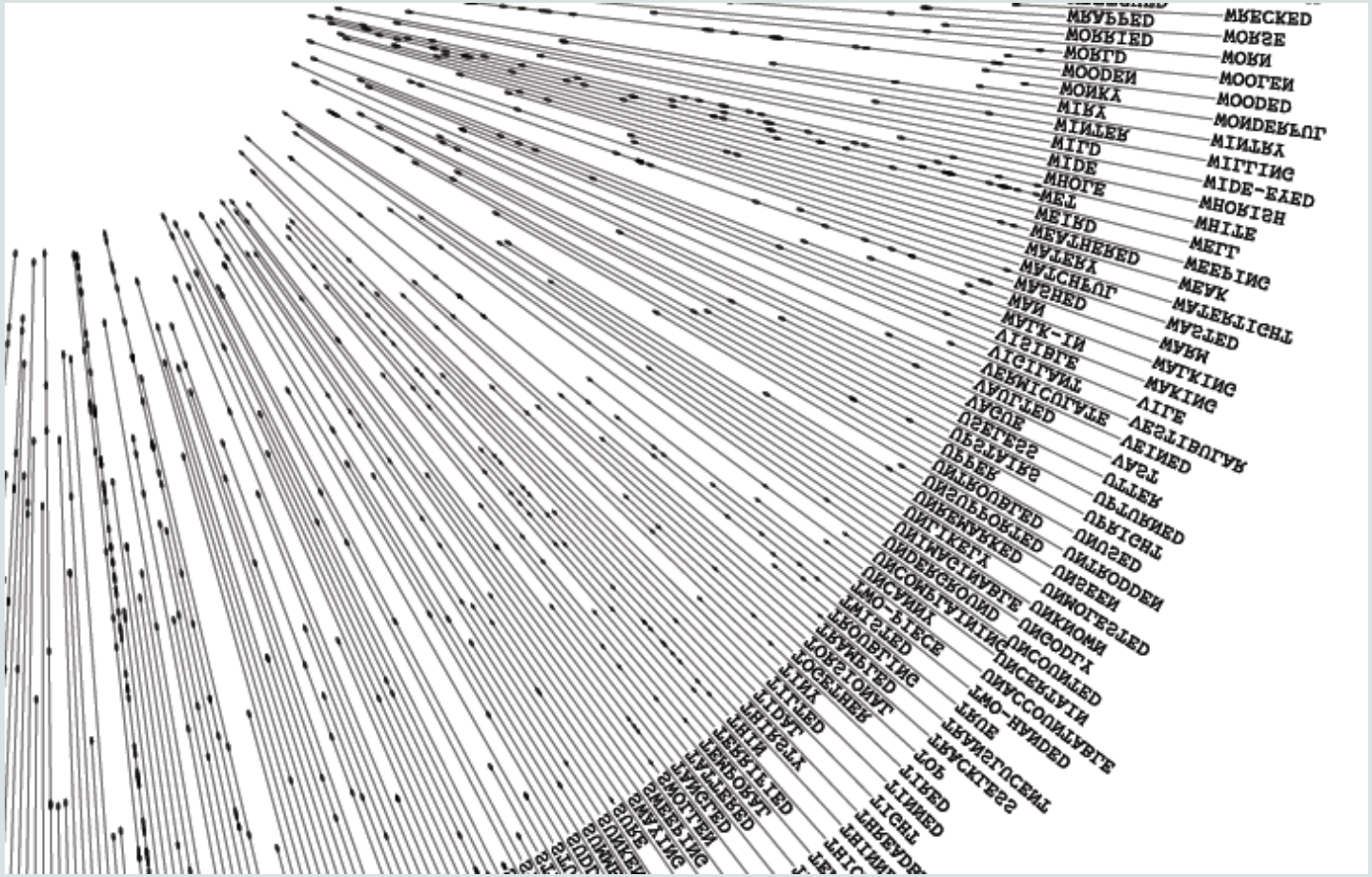
3. Visualization Function

- Convey an **explanatory** portrayal of data to a reader
- Provide an interface to data in order to facilitate visual **exploration**
- Use data as an **exhibition** of self expression

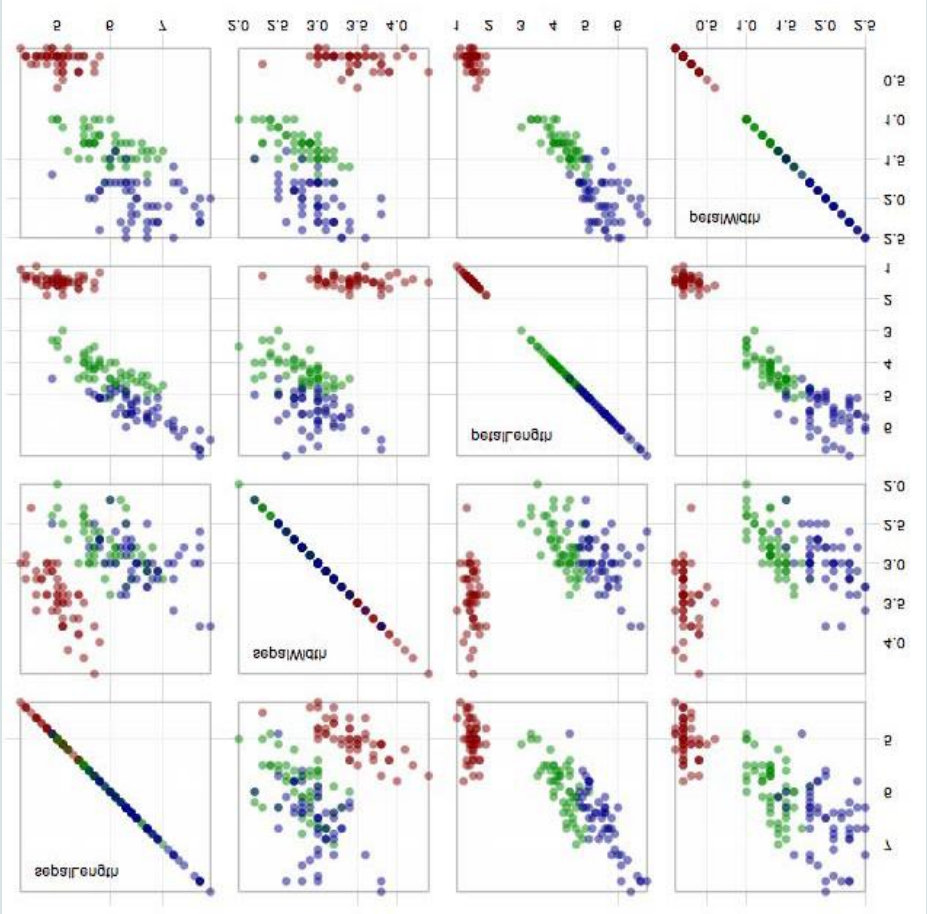
Explanatory



Exhibition



Exploratory



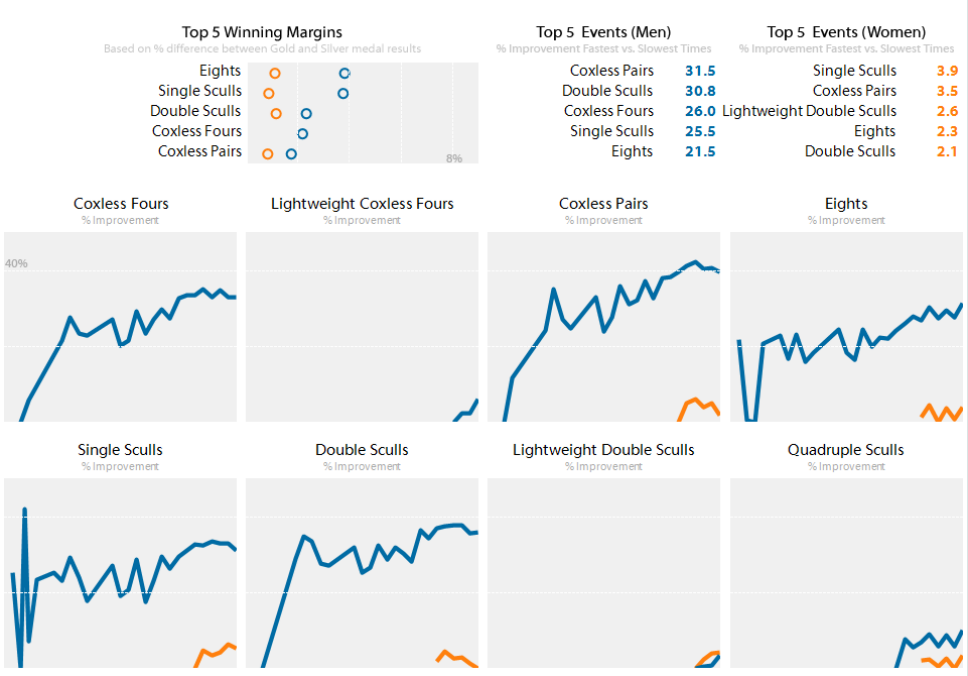
4. Visualizations Tone

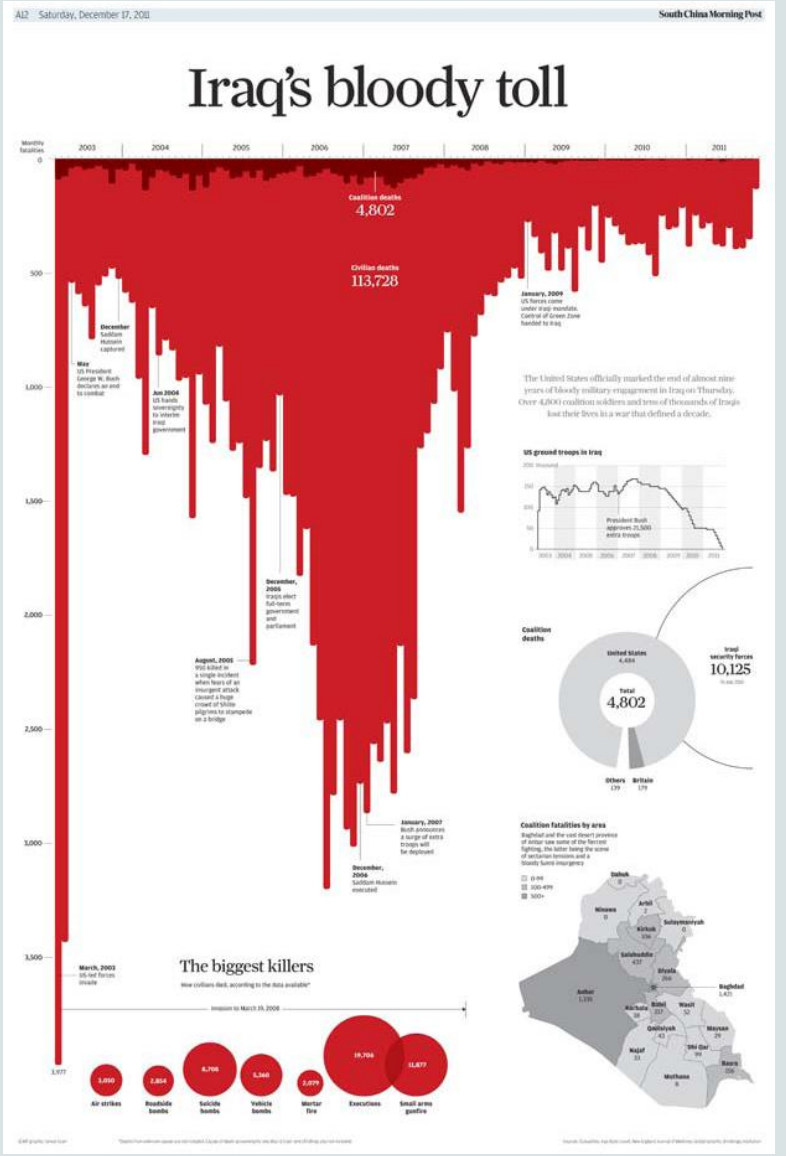
- Pragmatic or Analytic:

  "A visualization is more effective than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other." -- Jock Mackinlay

- Emotive and abstract

  "I have a fear that we aren't feeling enough, we aren't able to digest these huge numbers." – Chris Jordan


Top 5 Winning Margins
Based on % difference between Gold and Silver medal results
Eights
Single Sculls
Double Sculls
Coxless Fours
Coxless Pairs
8%
Top 5 Events (Men)
% Improvement Fastest vs. Slowest Times
Coxless Pairs 31.5
Double Sculls 30.8
Coxless Fours 26.0
Single Sculls 25.5
Eights 21.5
Top 5 Events (Women)
% Improvement Fastest vs. Slowest Times
Single Sculls 3.9
Coxless Pairs 3.5
Lightweight Double Sculls 2.6
Eights 2.3
Double Sculls 2.1
Coxless Fours
% Improvement
40%
Lightweight Coxless Fours
% Improvement
Coxless Pairs
% Improvement
Eights
% Improvement
Single Sculls
% Improvement
Double Sculls
% Improvement
Lightweight Double Sculls
% Improvement
Quadruple Sculls
% Improvement


A12 Saturday, December 17, 2011
South China Morning Post
Iraq's bloody toll
Coalition deaths
4,802
Civilian deaths
113,728
The United States officially marked the end of almost nine years of bloody military engagement in Iraq on Thursday. Over 4,800 coalition soldiers and tens of thousands of Iraqis lost their lives in a war that defined a decade.
US ground troops in Iraq
Coalition deaths
Total
4,802
10,125
The biggest killers
Coalition fatalities by area

Data Visualization Process

Visualization Process

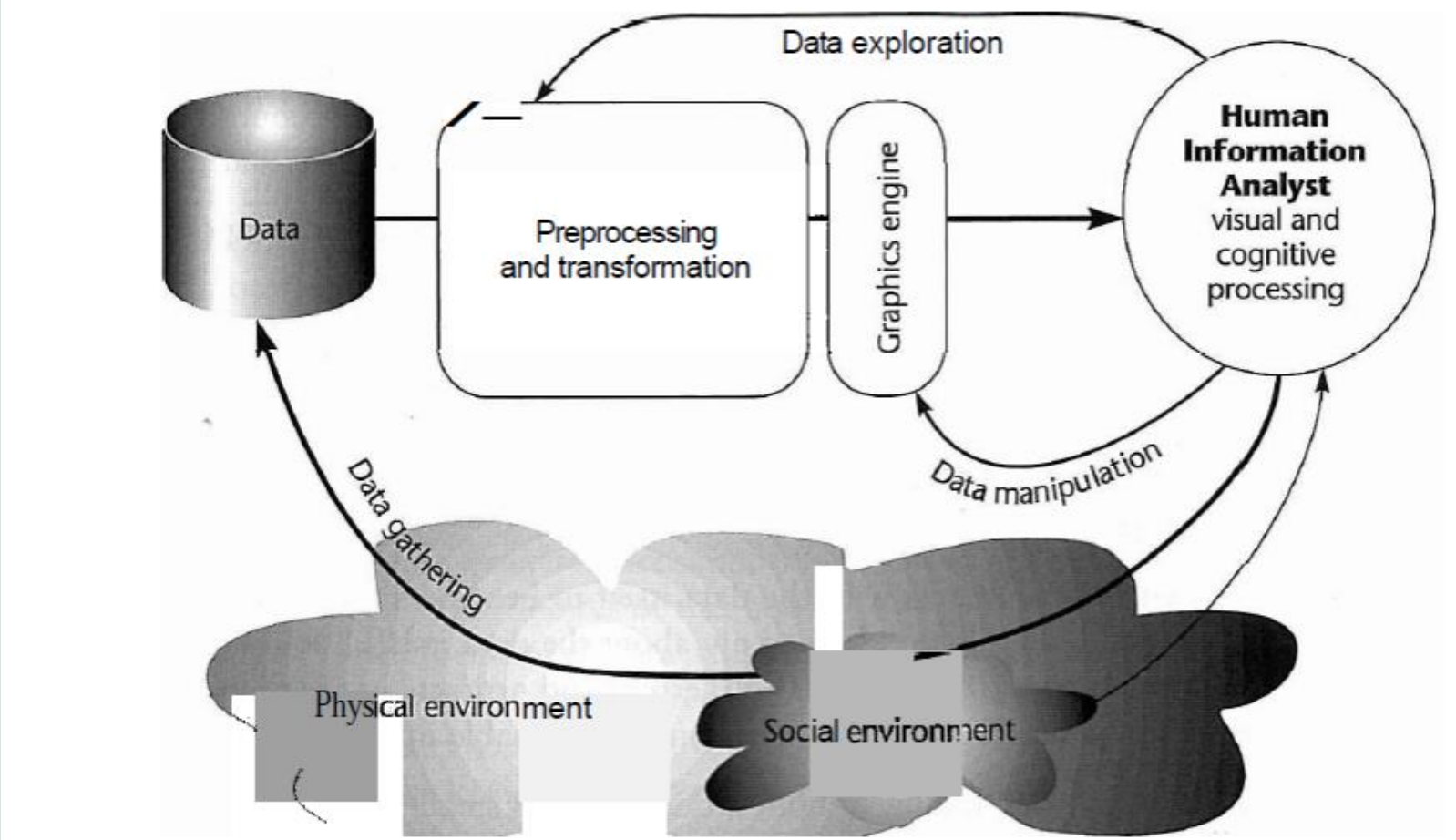
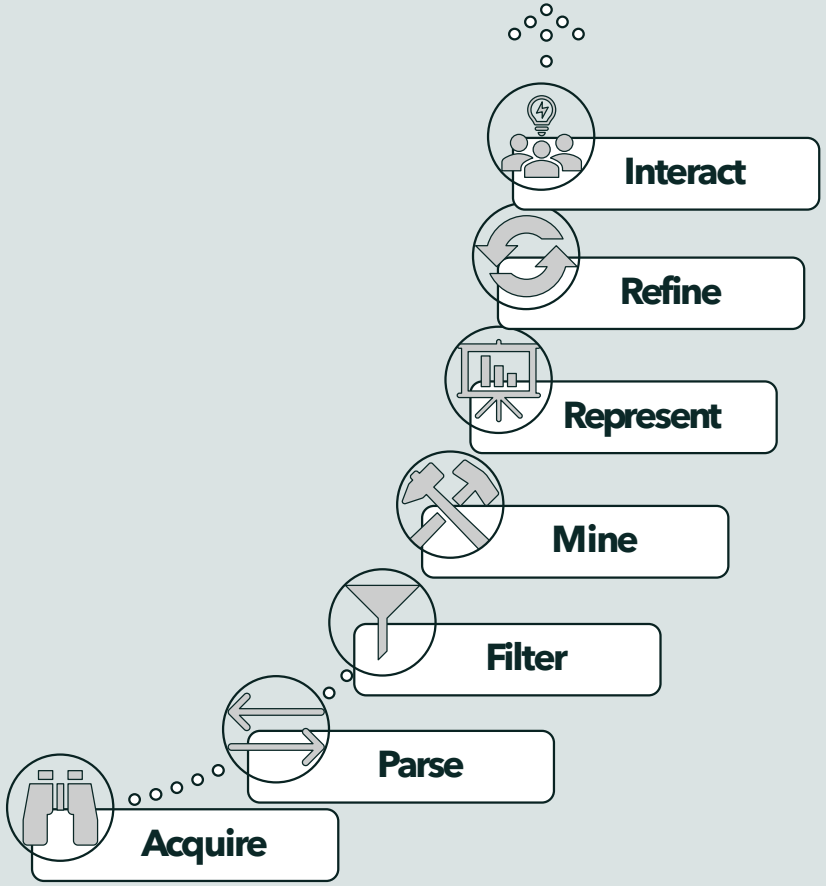
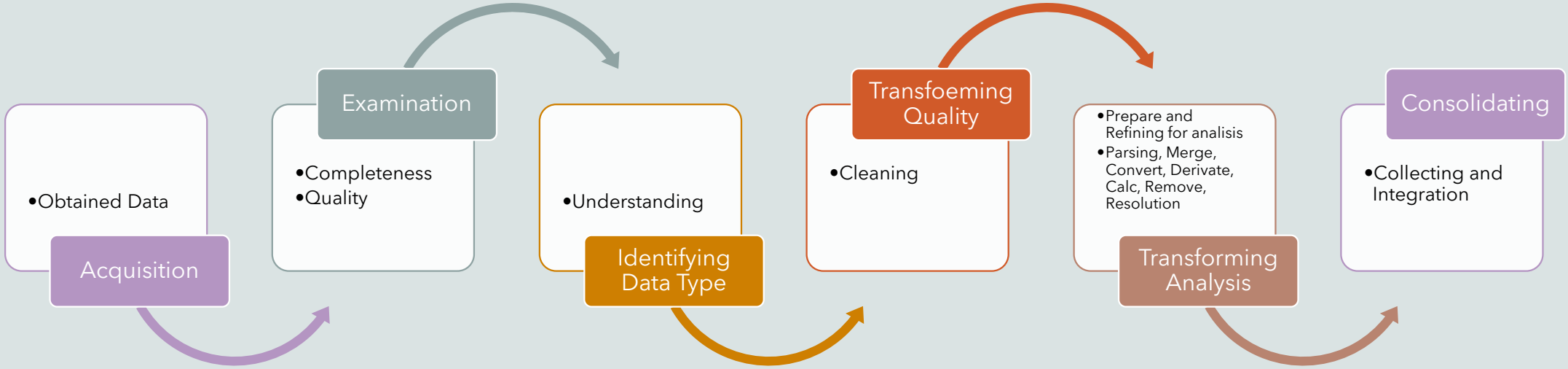


Figure 1.2 A schematic diagram of the visualization process.

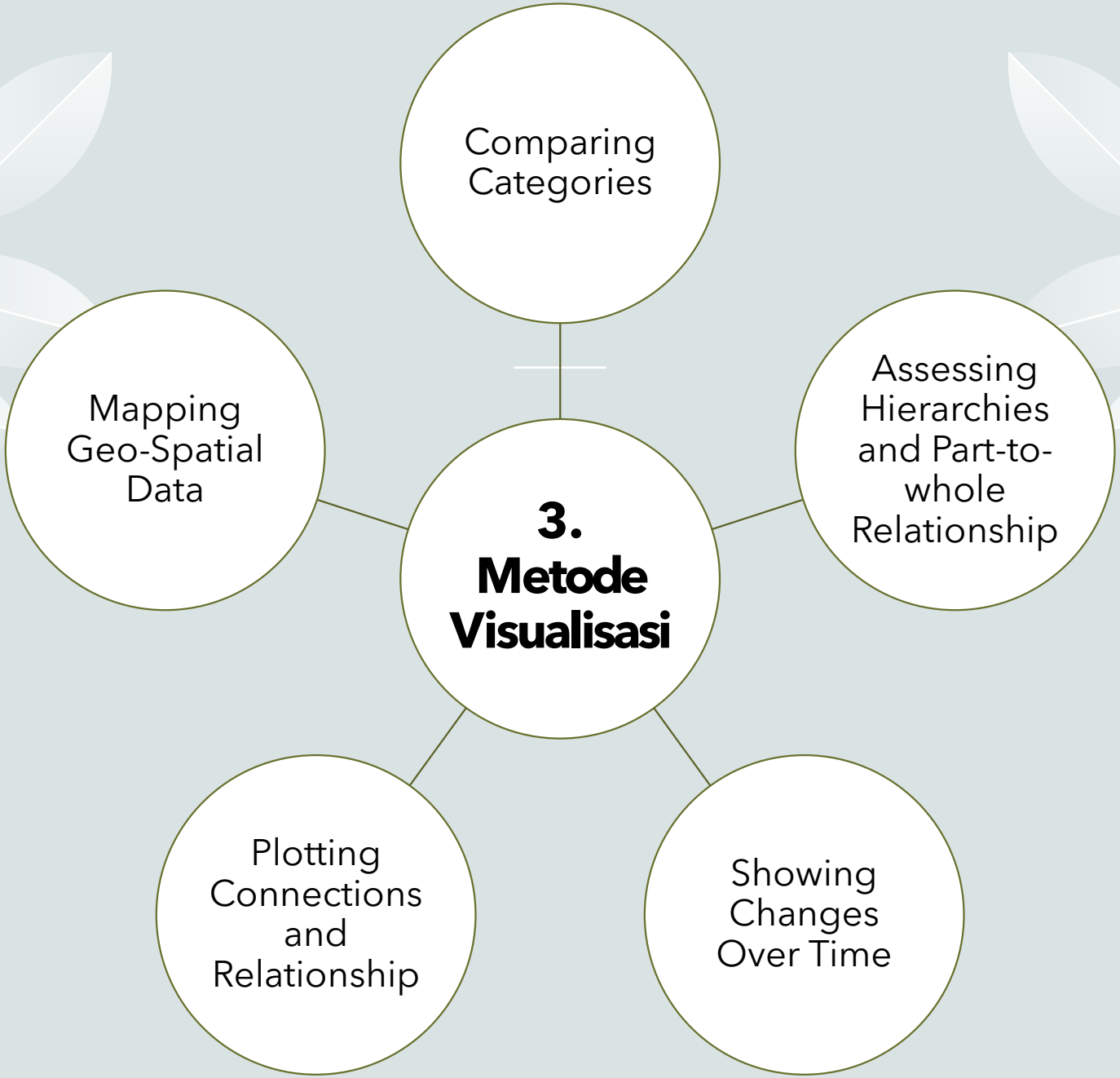
7 Stage to Build Visualization by Fry (2006)



Data Preparation



Metode Visualisasi


Comparing Categories
Mapping Geo-Spatial Data
3. Metode Visualisasi
Assessing Hierarchies and Part-to-whole Relationship
Plotting Connections and Relationship
Showing Changes Over Time

4. Pembangunan Visualisasi

APPLICATION

- Spreadsheet
- PowerBI
- Tableu
- Metabase
- Google Data Studio
- etc

PROGRAMMING AND LIBRARIES

- Python
- Javascript
- Ruby

5. Visual Analytics Technique

Comparison and Proportions

- Range dan Distribution
- Rangking
- Measurement
- Context/Judging

Trend and Pattern

- Direction
- Rate of Change
- Fluctuation
- Significance
- Intersection

Relationship and Connections

- Exception
- Correlation
- Association
- Clusters and Gaps
- Hierarchical Relationship

Thank you

Rahmatika Pratama Santi, M.T

Taxonomy of Viz. Methods and Viz. Design

Rahmatika Pratama Santi, M.T.

Taxonomy of Data Visualization Methods

Metode Visualisasi

Method classification	Communication purpose
Comparing categories	To facilitate comparisons between the relative and absolute sizes of categorical values. The classic example would be the bar chart.
Assessing hierarchies and part-to-whole relationships	To provide a breakdown of categorical values in their relationship to a population of values or as constituent elements of hierarchical structures. The example here would be the pie chart.
Showing changes over time	To exploit temporal data and show the changing trends and patterns of values over a continuous timeframe. A typical example is the line chart.
Plotting connections and relationships	To assess the associations, distributions, and patterns that exists between multivariate datasets. This collection of solutions reflects some of the most complex visual solutions and usually focuses on facilitating exploratory analysis. A common example would be the scatter plot.
Mapping geo-spatial data	To plot and present datasets with geo-spatial properties via the many different mapping frameworks. A popular approach would be the choropleth map.

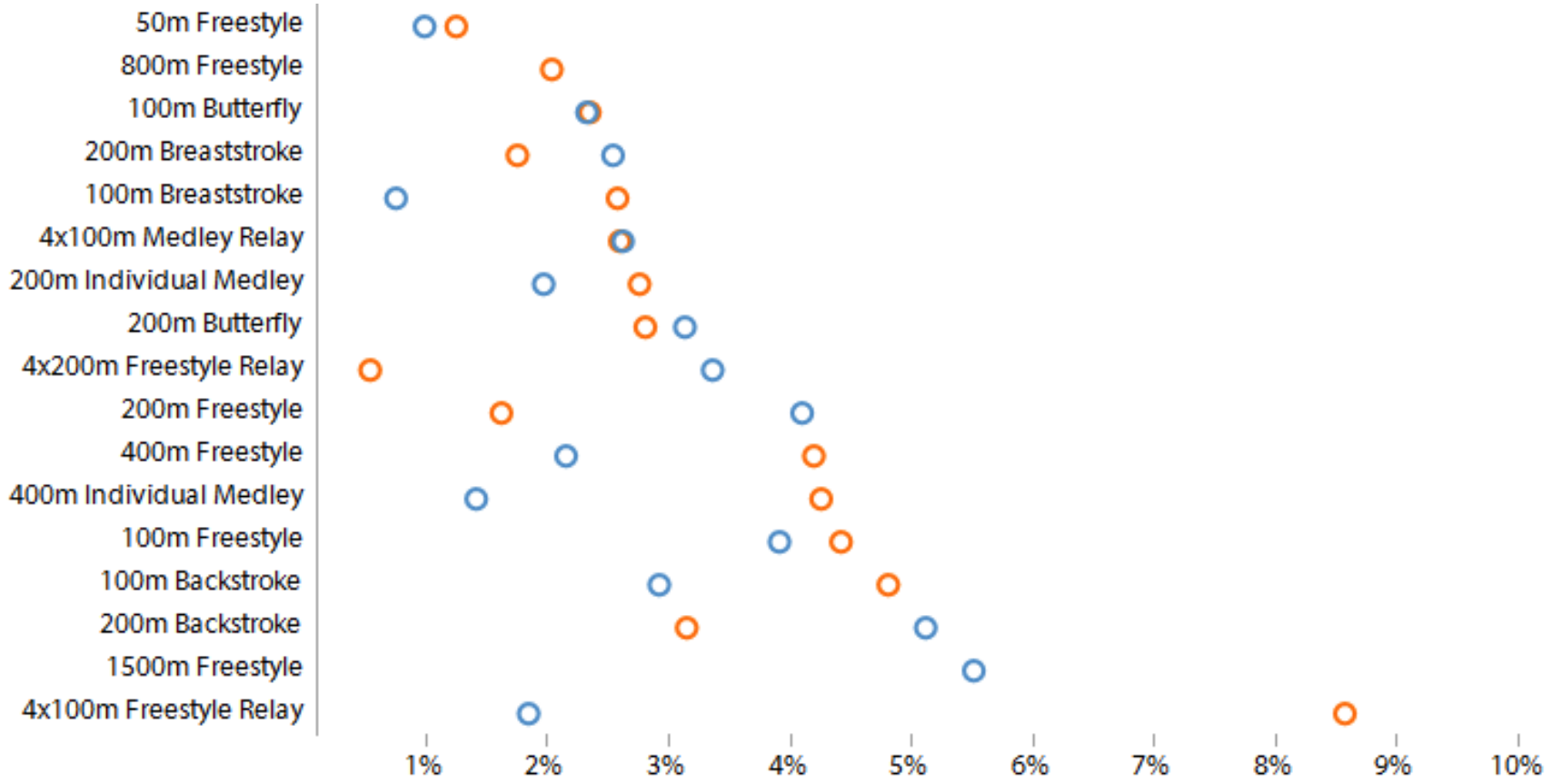
Comparing Categories

- Dot Plot
- Bar Chart
- Floating Bar
- Pixelated Bar Chart
- Histogram
- Slopegraph
- Radial Chart
- Glyph Chart
- Sankey Diagram
- Area Size Chart
- Small Multiples
- Word Cloud

Next

Comparing Categories – Dot Plot

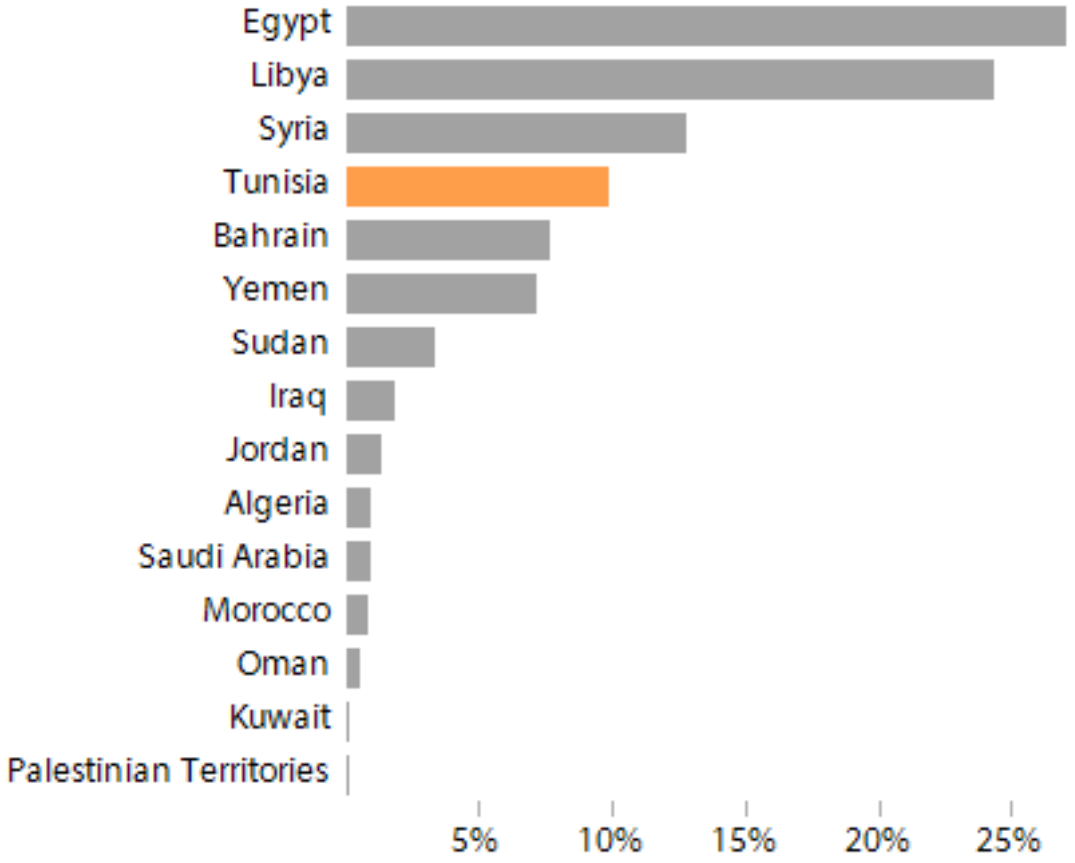
- Clear Range
- Distribution Values



List

Comparing Categories – Bar Chart

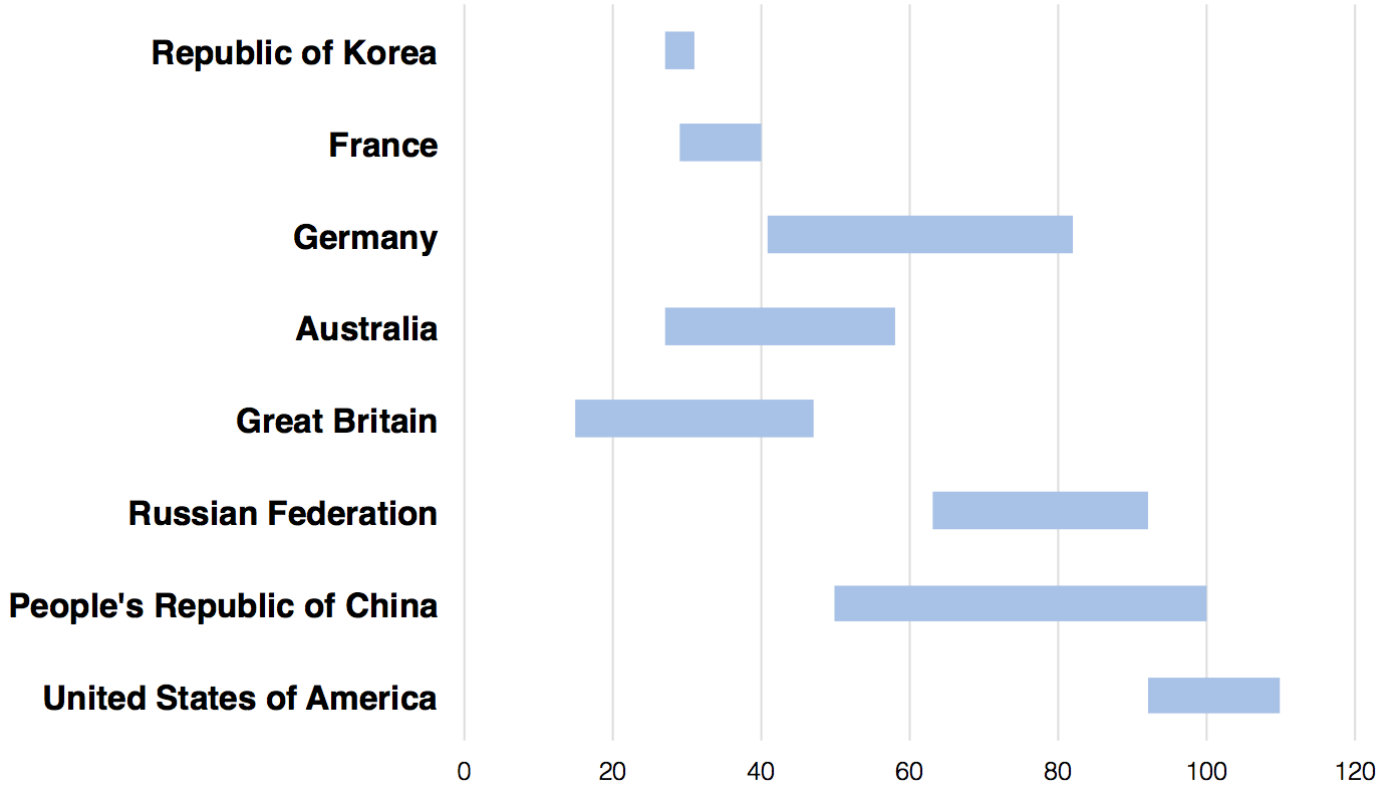
- Show the full extent of this property so always start the bar from the zero point on the axis.
- The use of color can help draw attention to the values of specific categories in accordance with your narrative



List

Comparing Categories – Floating Bar

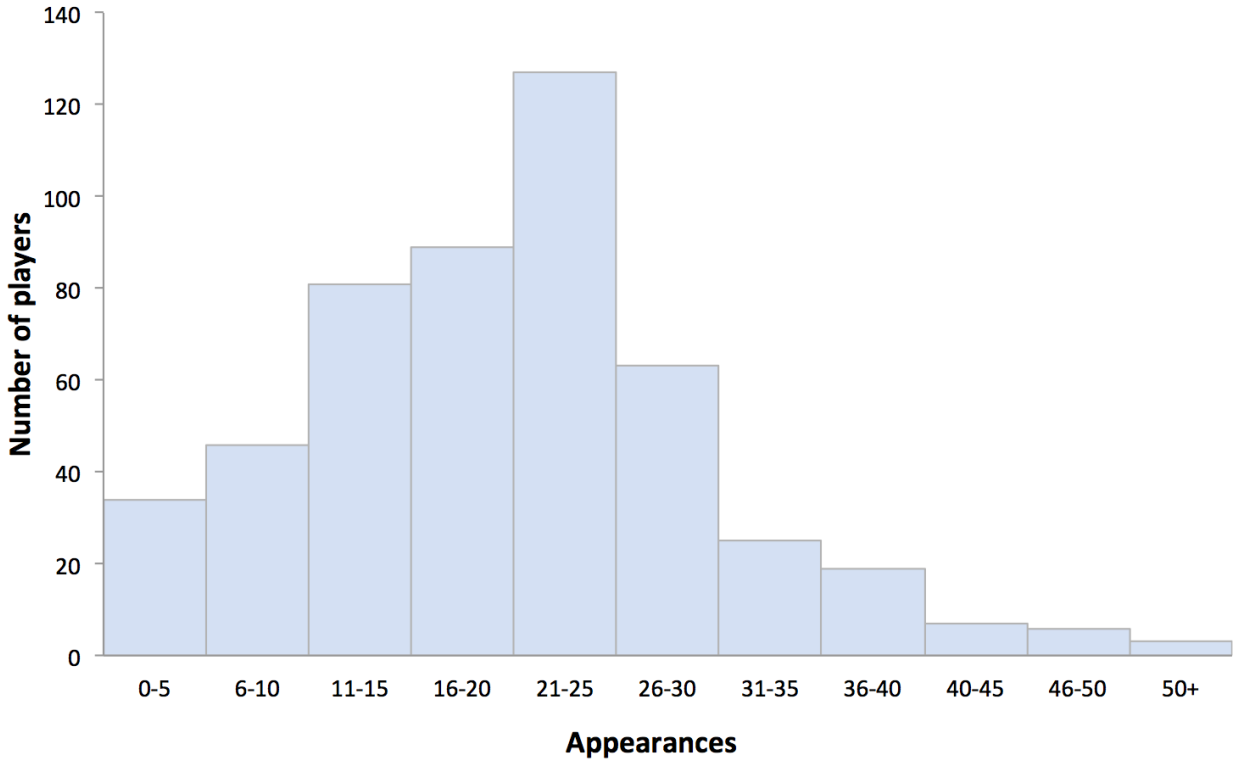
- It presents a bar stretching from the lowest to the highest values
- enables you to identify the diversity of measurements within a category and view overlaps and outliers across all categories.



List

Comparing Categories – Histogram

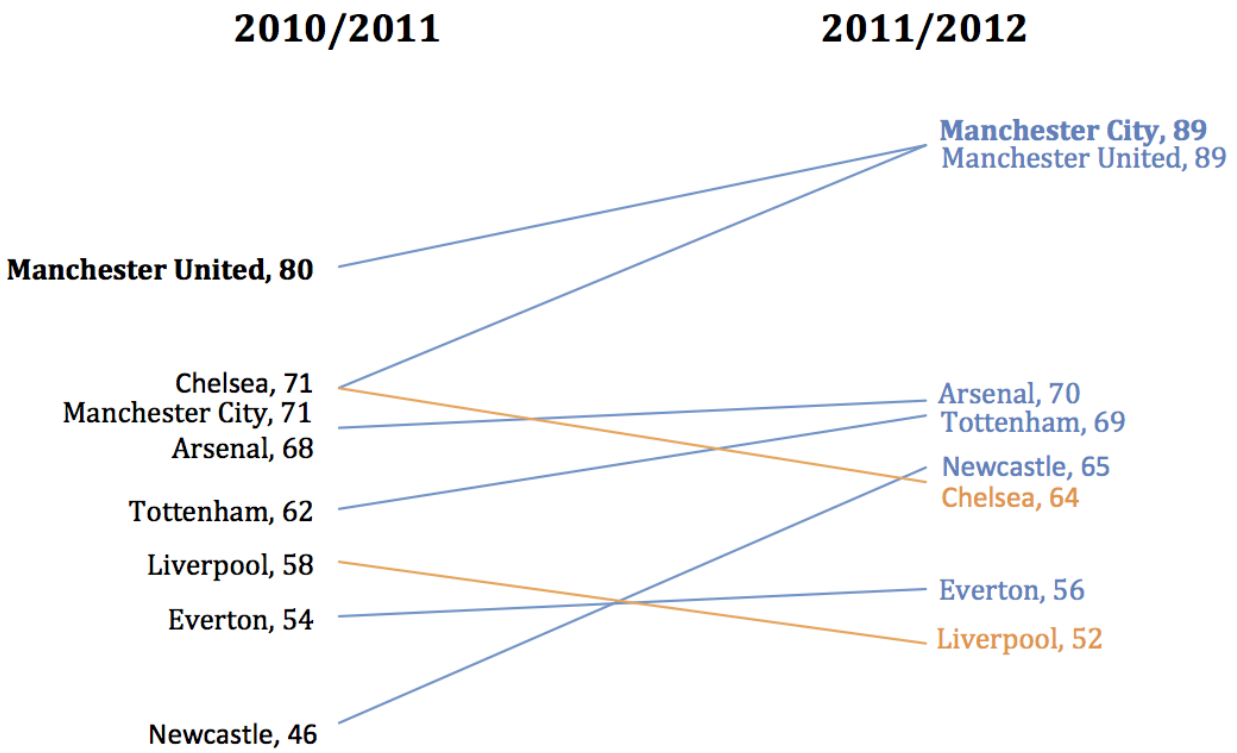
- Histograms are often mistaken for bar charts but there are important differences.
- Histograms show distribution through the frequency of quantitative values (y axis) against defined intervals of quantitative values(x axis).
- By contrast, bar charts facilitate comparison of categorical values.
- One of the distinguishing features of a histogram is the lack of gaps between the bars,



List

Comparing Categories – Slopegraph

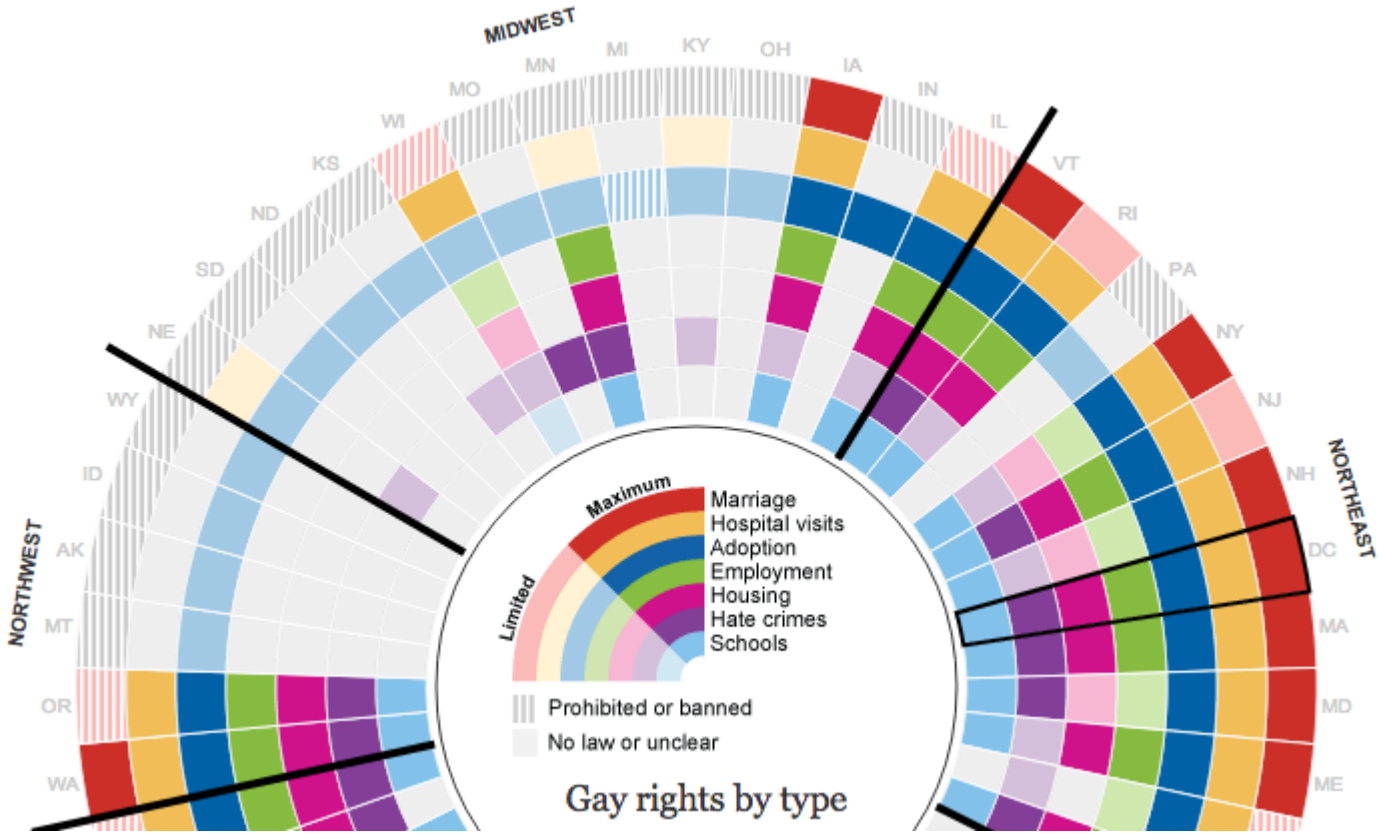
- They especially provide a neat way of showing a before and after view or comparison of two different points in time.



List

Comparing Categories – Radial Chart

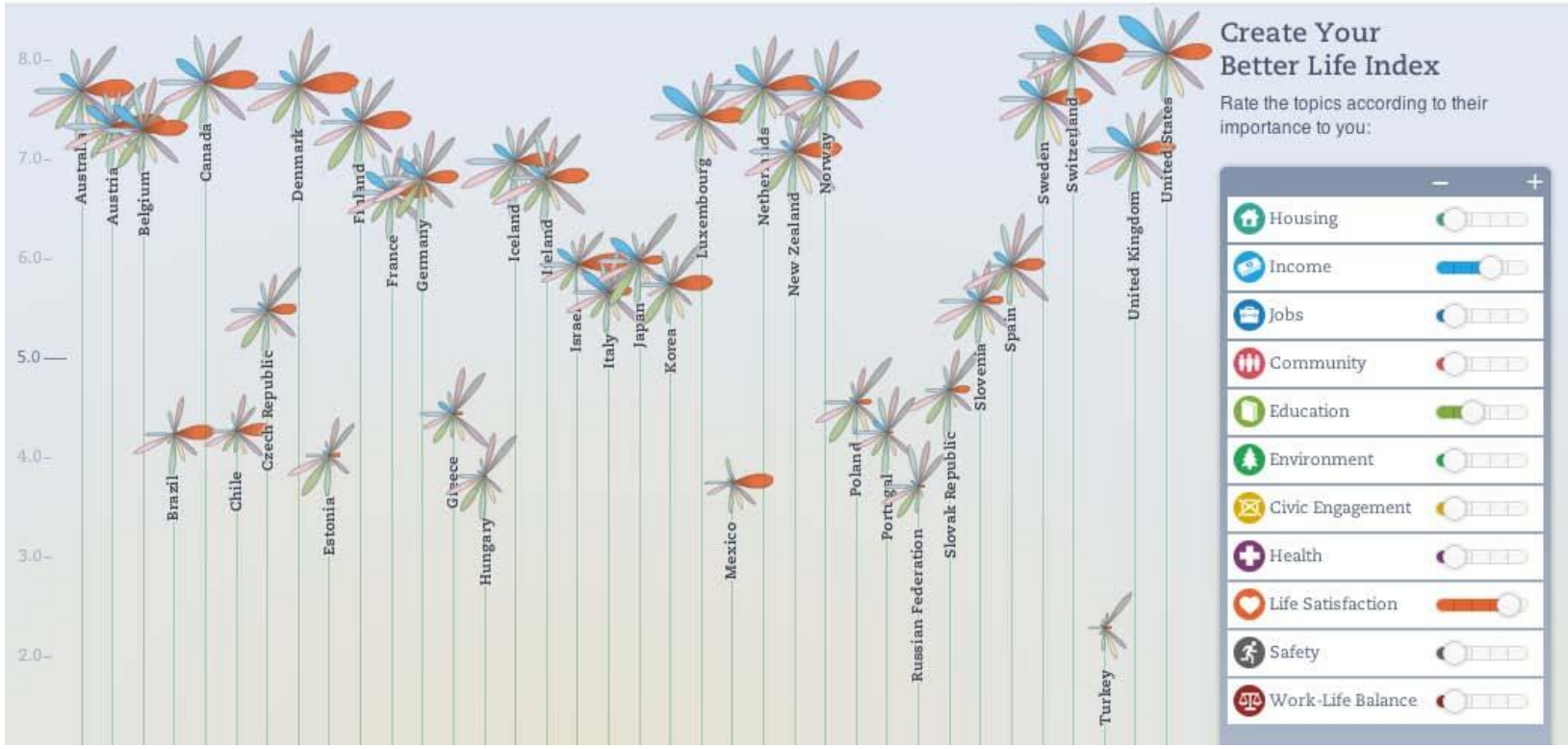
- They especially provide a neat way of showing a before and after view or comparison of two different points in time.



List

Comparing Categories – Glyph

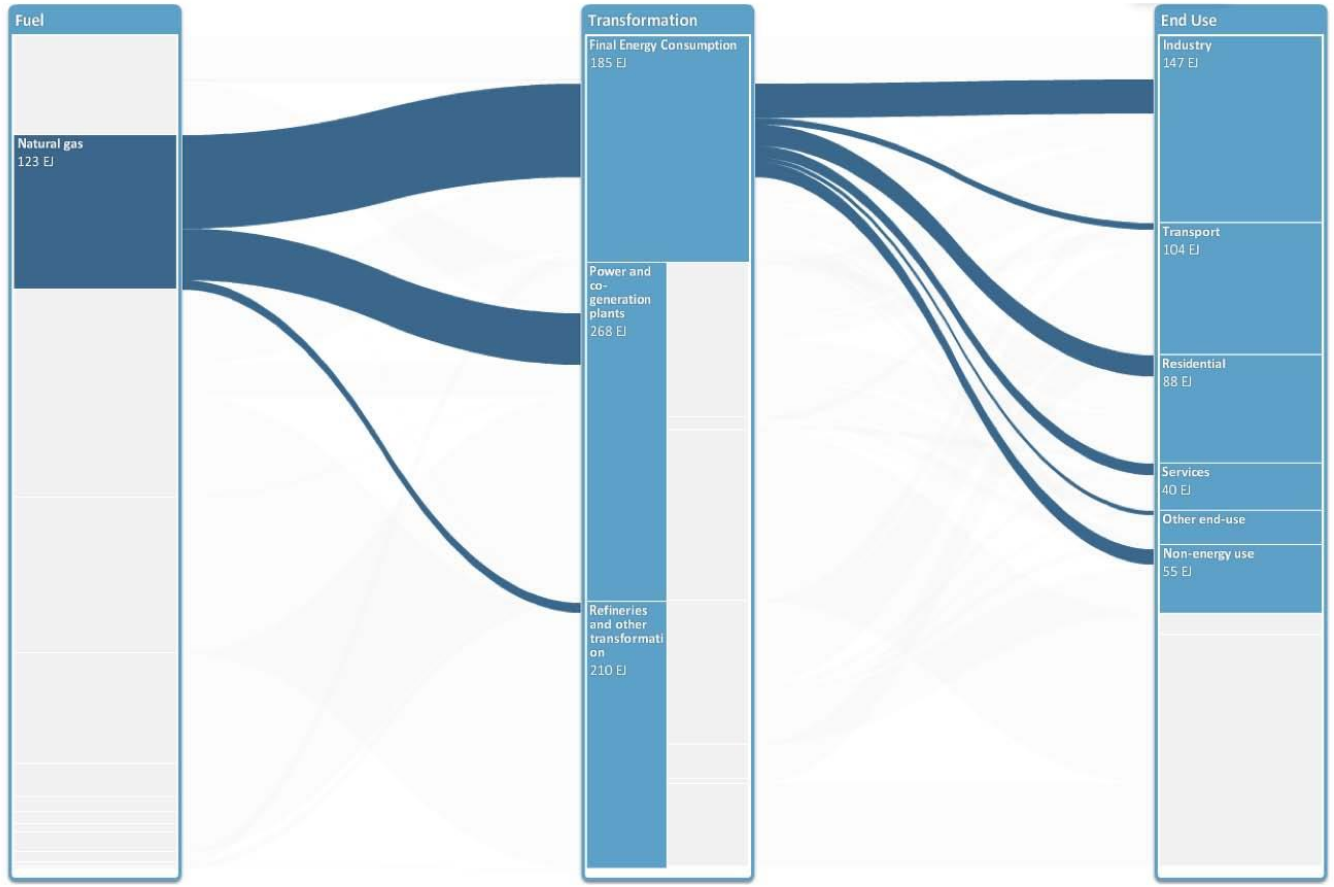
- A glyph chart is based on a shape (in the following example, a flower) being the main artifact of representation.
- The physical properties of the shape (through a feature such as a petal) represent different categorical variables
- They are sized according to the associated quantitative value and distinguished through color.



List

Comparing Categories – Sankey Diagram

- Sankey diagrams are used to convey the idea of flow.
- They portray constituent quantities of a series of associated categorical values, across a number of "stages", with the ongoing associations represented by connecting bands.
- The width of these links indicates the proportional flow from one stage to another.
- They are useful for showing situations where elements transform and divide over key events, as shown here displaying the breakdown of different fuels, how they are transformed and then ultimately used.



List

Comparing Categories – Word Cloud

- Word clouds depict the frequency of words used in a given set of text.
- The font size indicates the quantity of each word's usage. Color is often just used as decoration (which you'll notice actually distorts the visual prominence).
- While it's fair to say they are becoming something of a ubiquitous visual commodity,
- they can be useful for exploring datasets for the first time in order to identify key terms being used.

List

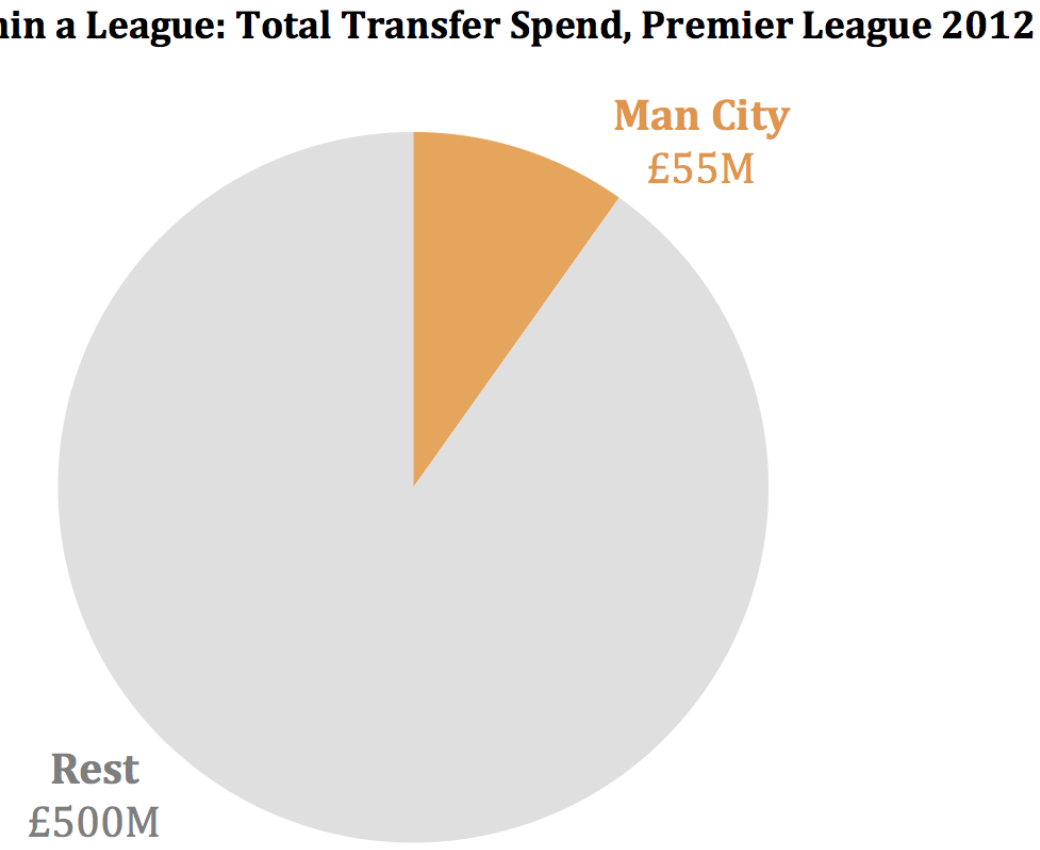
Assessing Hierarchies And Part-to-whole Relationships

- Pie Chart
- Stacked Bar Chart
- Square Pie
- Tree Map
- Circle Packing Diagram
- Bubble Hierarchy
- Tree Hierarchy

Next

Assessing Hierarchies – Pie Chart

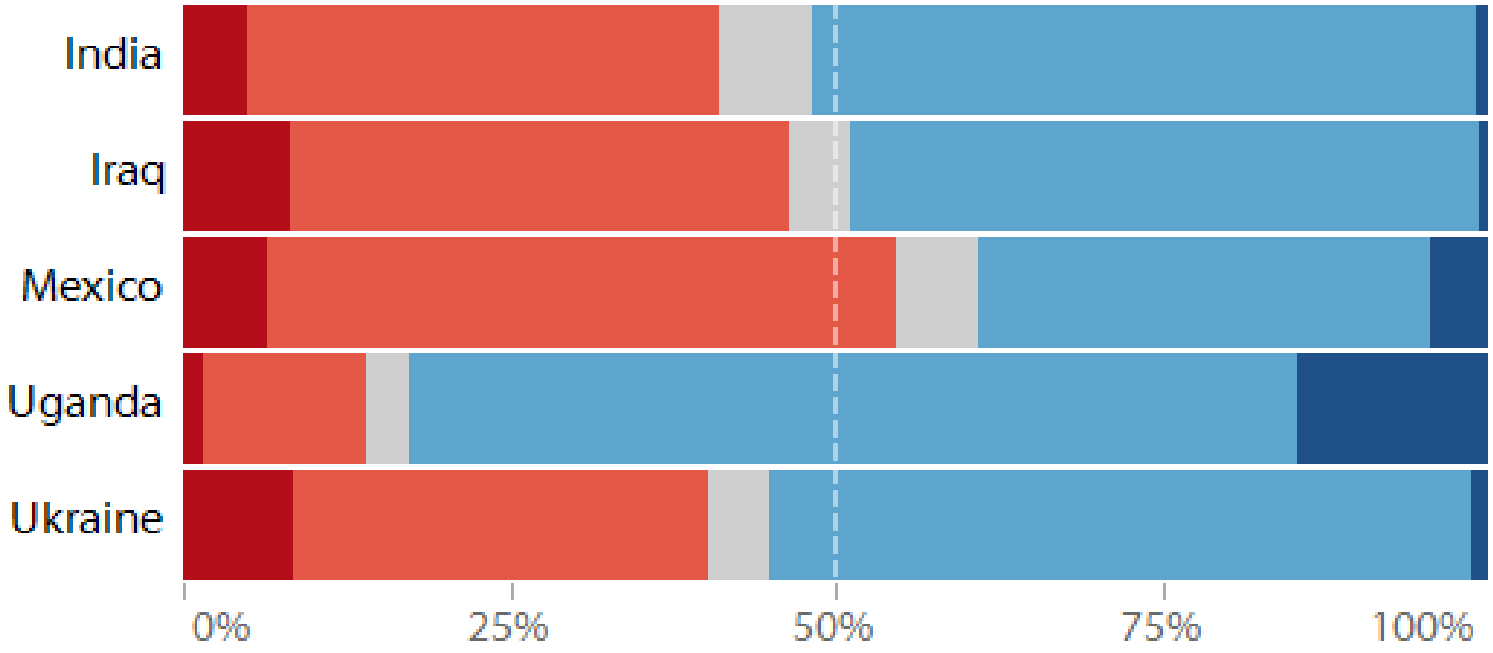
- Text



List

Assessing Hierarchies – Stacked Bar Chart

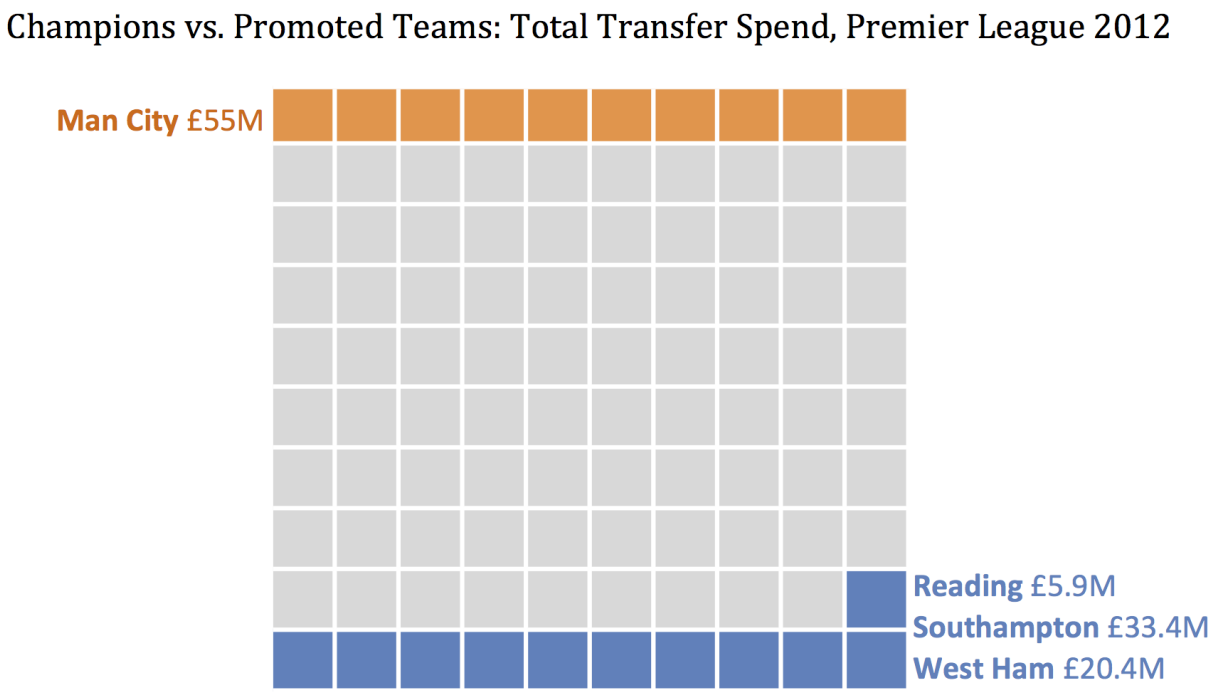
- Colors and position differentiate the value categories.



List

Assessing Hierarchies – Square Pie

- The use of color and symbol establishes the visual composition of the categorical and quantitative values



List

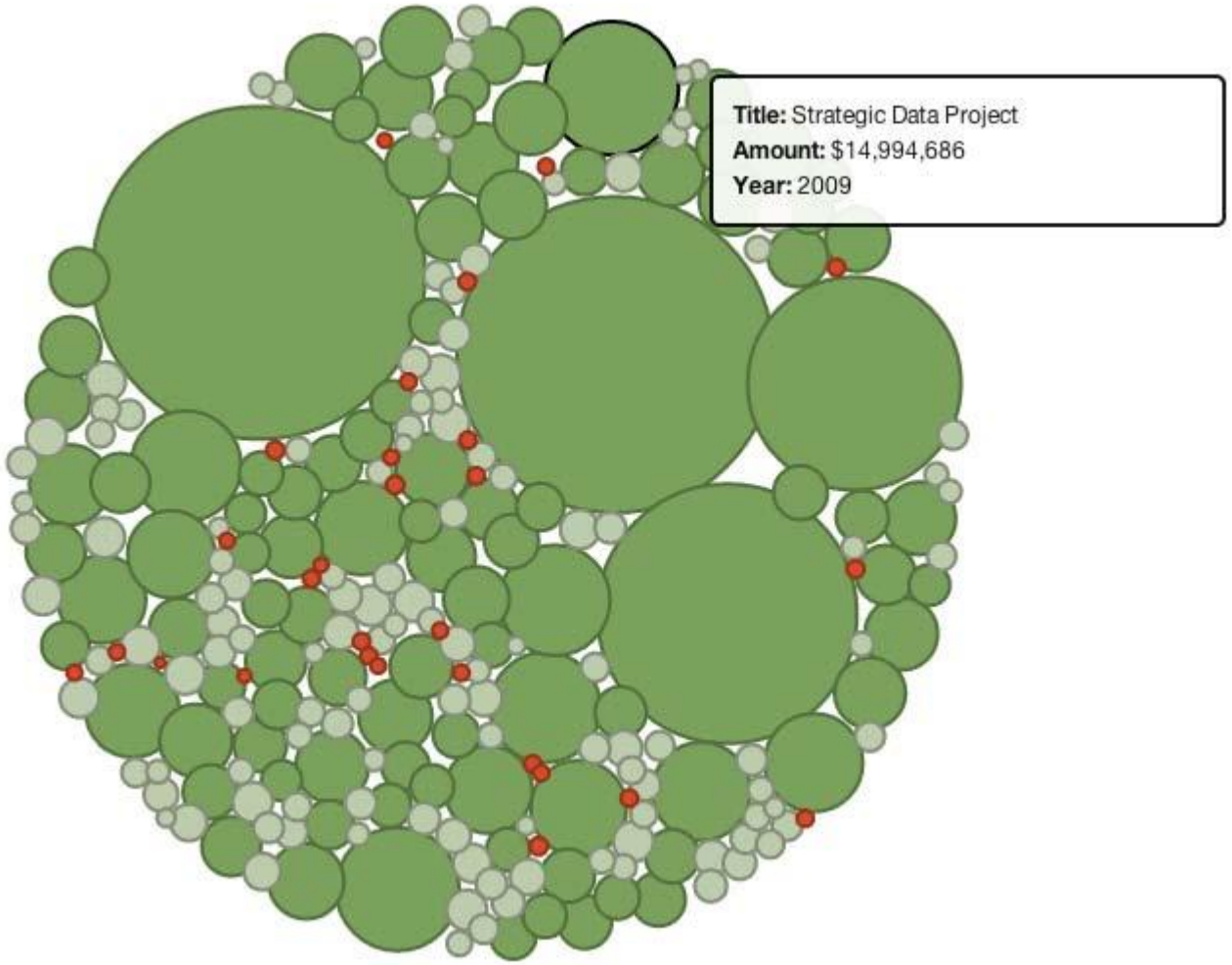
Assessing Hierarchies – Tree Map

- clustered constituent units sized according to their relative value. As well as arrangement
- various properties of color are typically used to provide additional layers of quantitative or categorical insight.

List

Assessing Hierarchies – Circle Packing Diagram

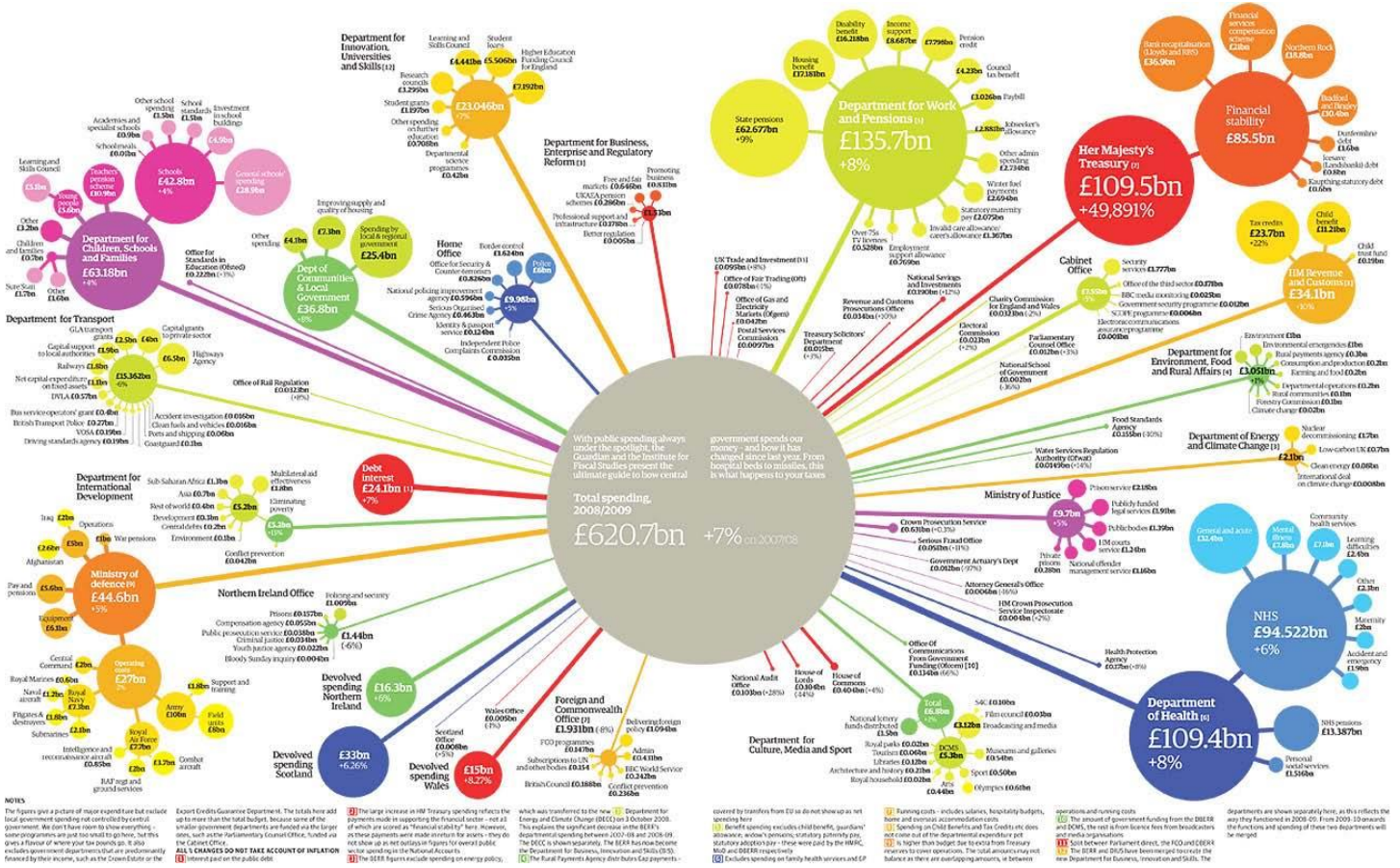
- Each individual circle represents a different category and is sized according to the associated quantitative value.
- Other visual variables, such as color and position, are often incorporated to enhance the layers of meaning of the display.



List

Assessing Hierarchies – Bubble Hierarchy

- This technique is used to portray organization and structure through a hierarchical display.
- In the following example, we see the use of circles to represent the constituent departments, sized according to their quantitative value and colored to visually distinguish the different departments.



List

Assessing Hierarchies – Tree Hierarchy

- Similar to the bubble hierarchy, this technique presents the organization and structure of data through a hierarchical tree network.
- In the following example, portraying the structure of a book, the effect is quite abstract but every visual property is serving the purpose of representing just the data

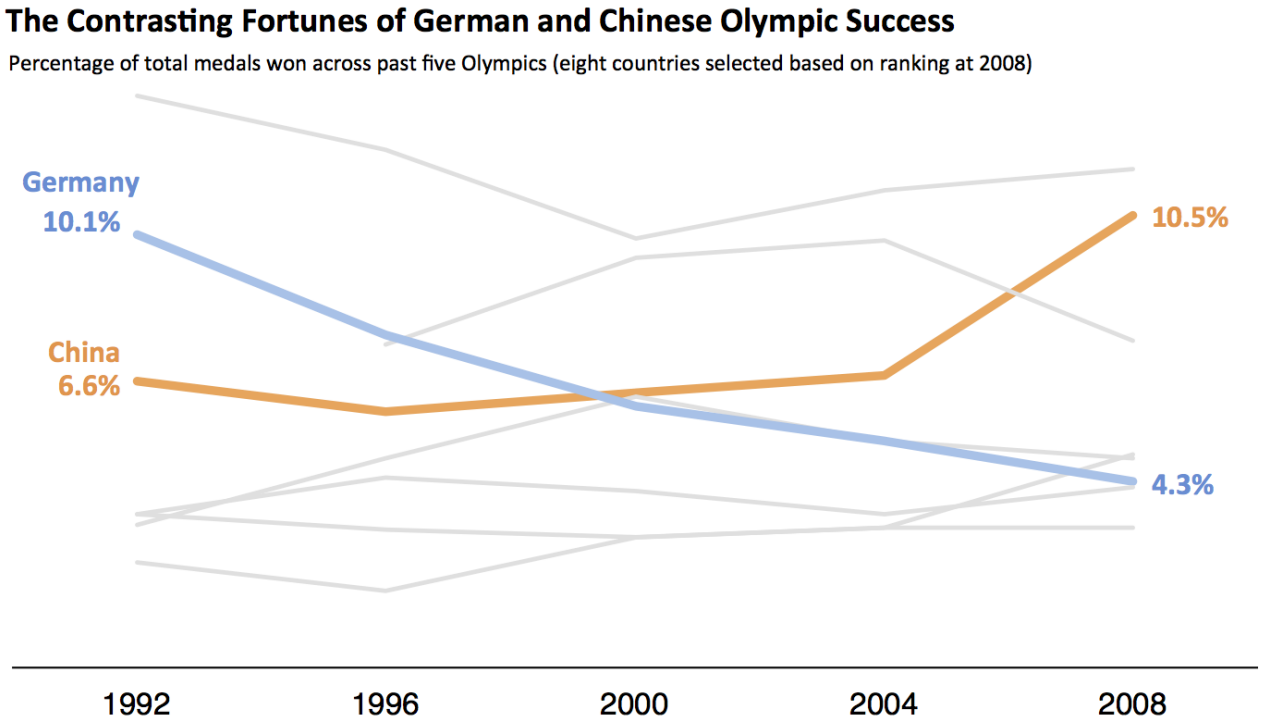
List

Showing Changes Over Time

- Line Chart
- Sparklines
- Area Chart
- Horizon Chart
- Stacked Area Chart
- Stream Graph
- Candlestick Chart
- Barcode Chart
- Flow Map

Change Over Time – Line Chart

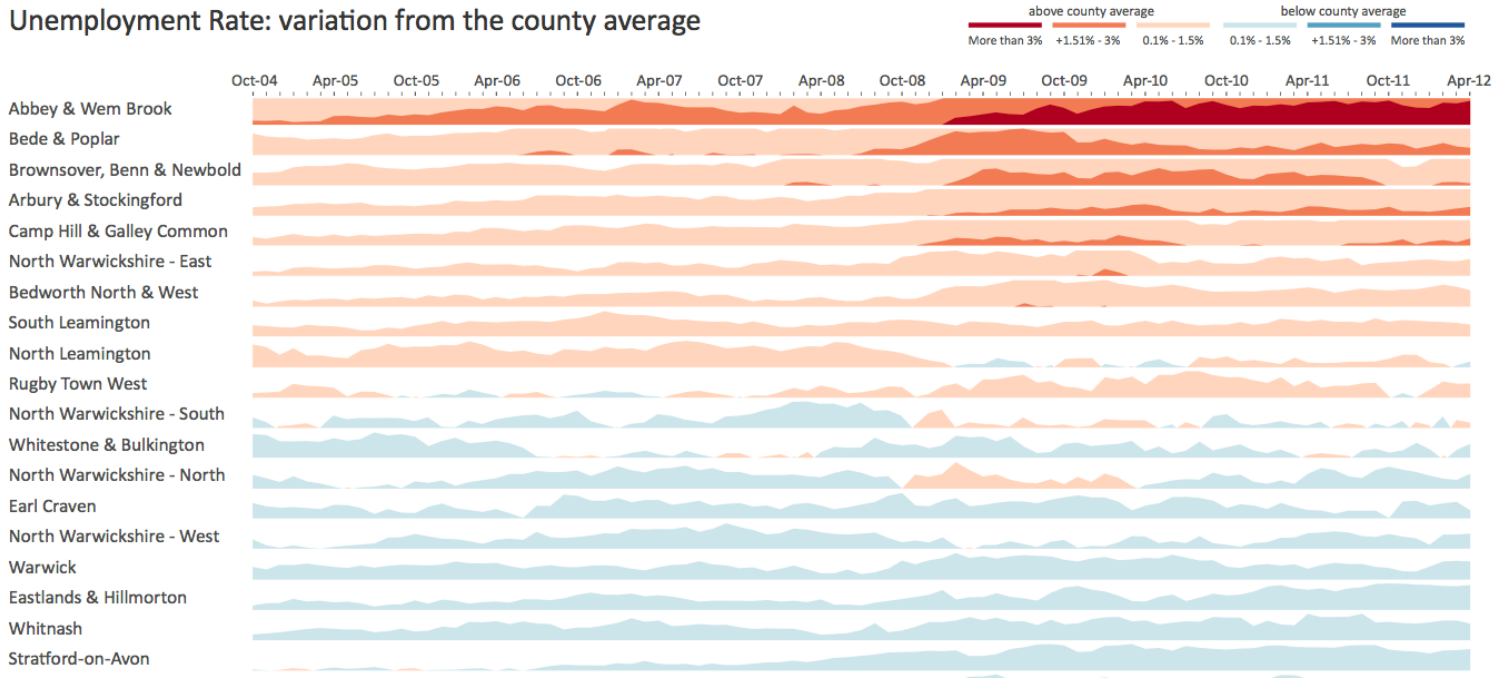
- They are used to compare a continuous quantitative variable on the x axis and the size of values on the y axis.
- The vertical points are joined up using lines to show the shifting trajectory through the resulting slopes.



List

Change Over Time – Horizon Chart

- This is a variation on the area chart, modified to include (and cope with) both positive and negative values.
- Rather than presenting negative values beneath the x axis, the negative area is mirrored on to the positive side and then colored differently to indicate its negative polarity.
- The result is a chart that occupies a single row of space, which helps to accommodate multiple stories onto a single display and facilitates comparison to pick out local and global patterns of change over time.

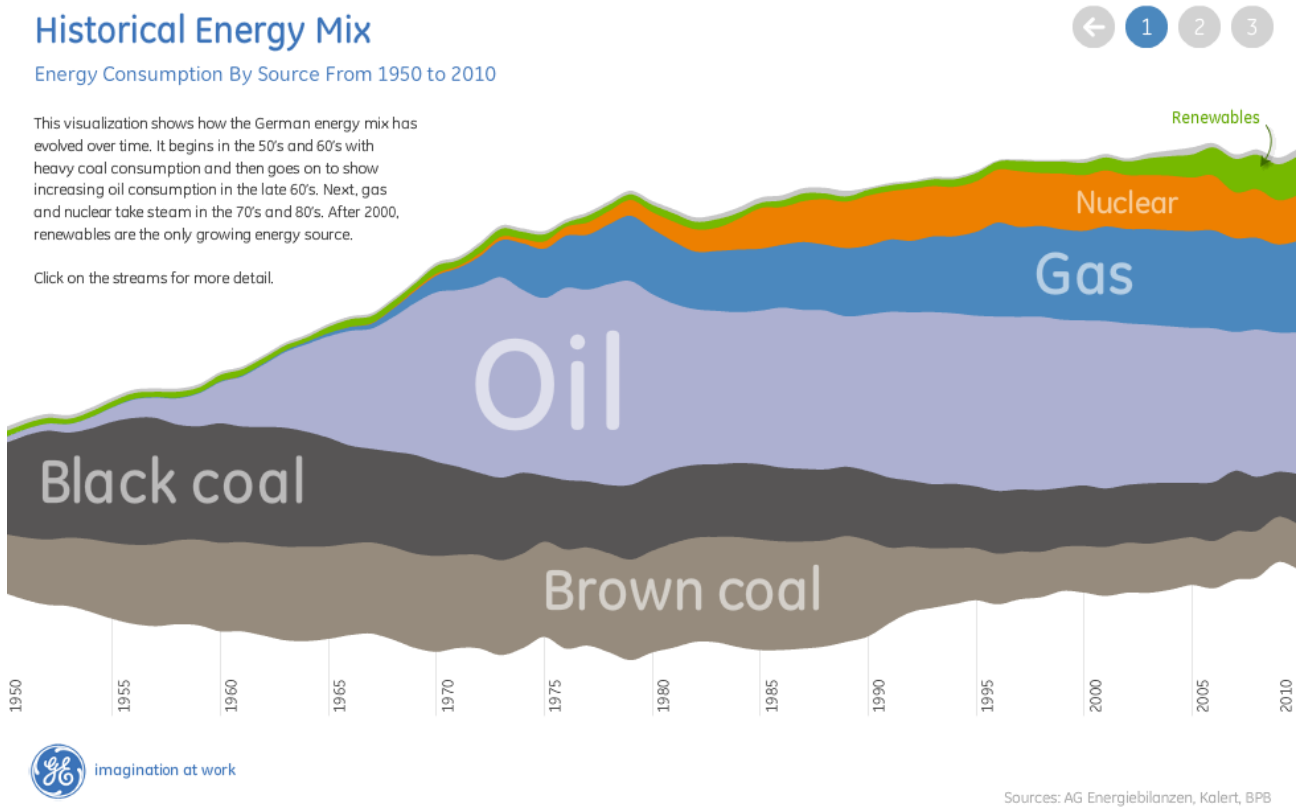


List

Change Over Time – Stream Graph

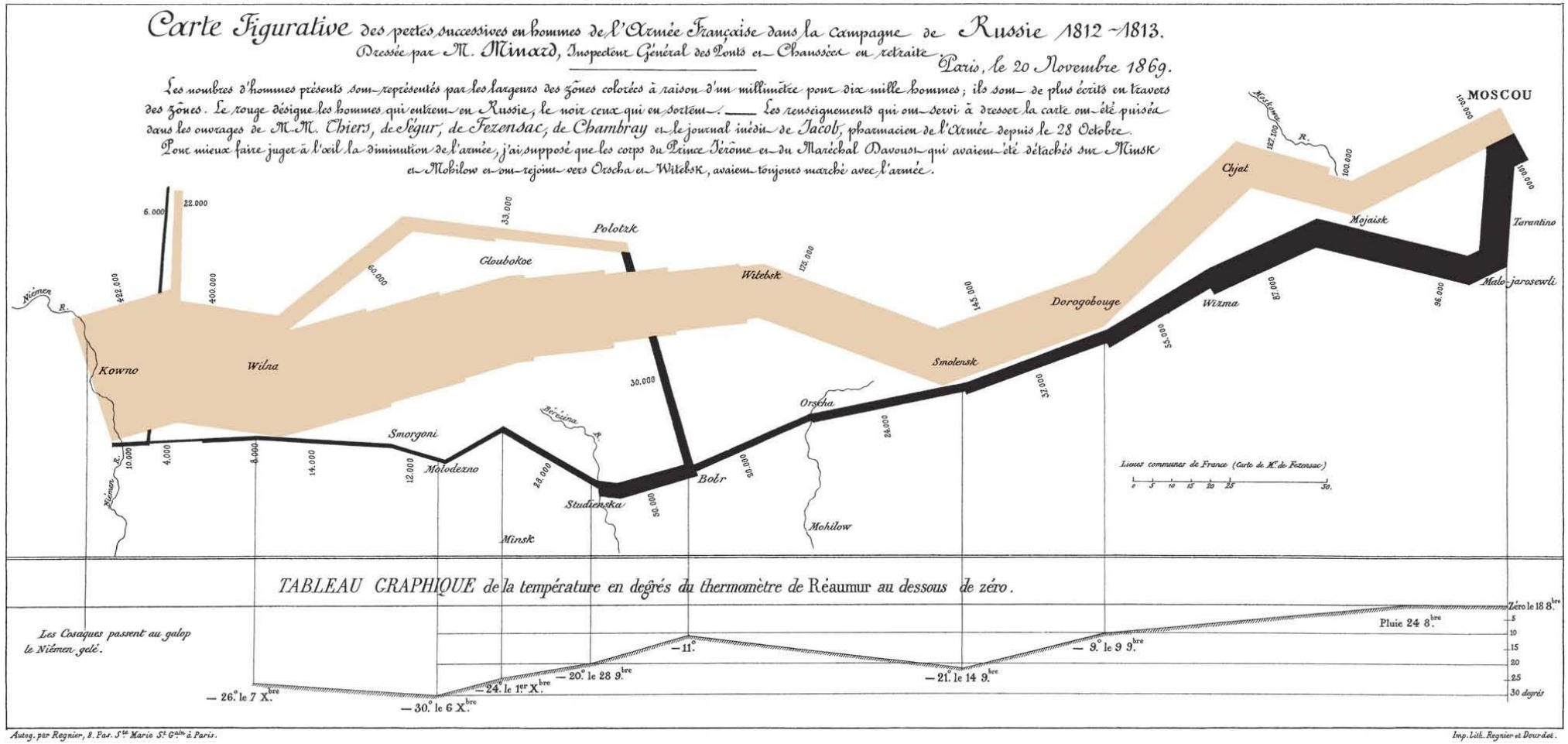
- This is a variation on the area chart, modifed to include (and cope with) both positive and negative values.
- Rather than presenting negative values beneath the x axis, the negative area is mirrored on to the positive side and then colored differently to indicate its negative polarity.
- The result is a chart that occupies a single row of space, which helps to accommodate multiple stories onto a single display and facilitates comparison to pick out local and global patterns of change over time.

List



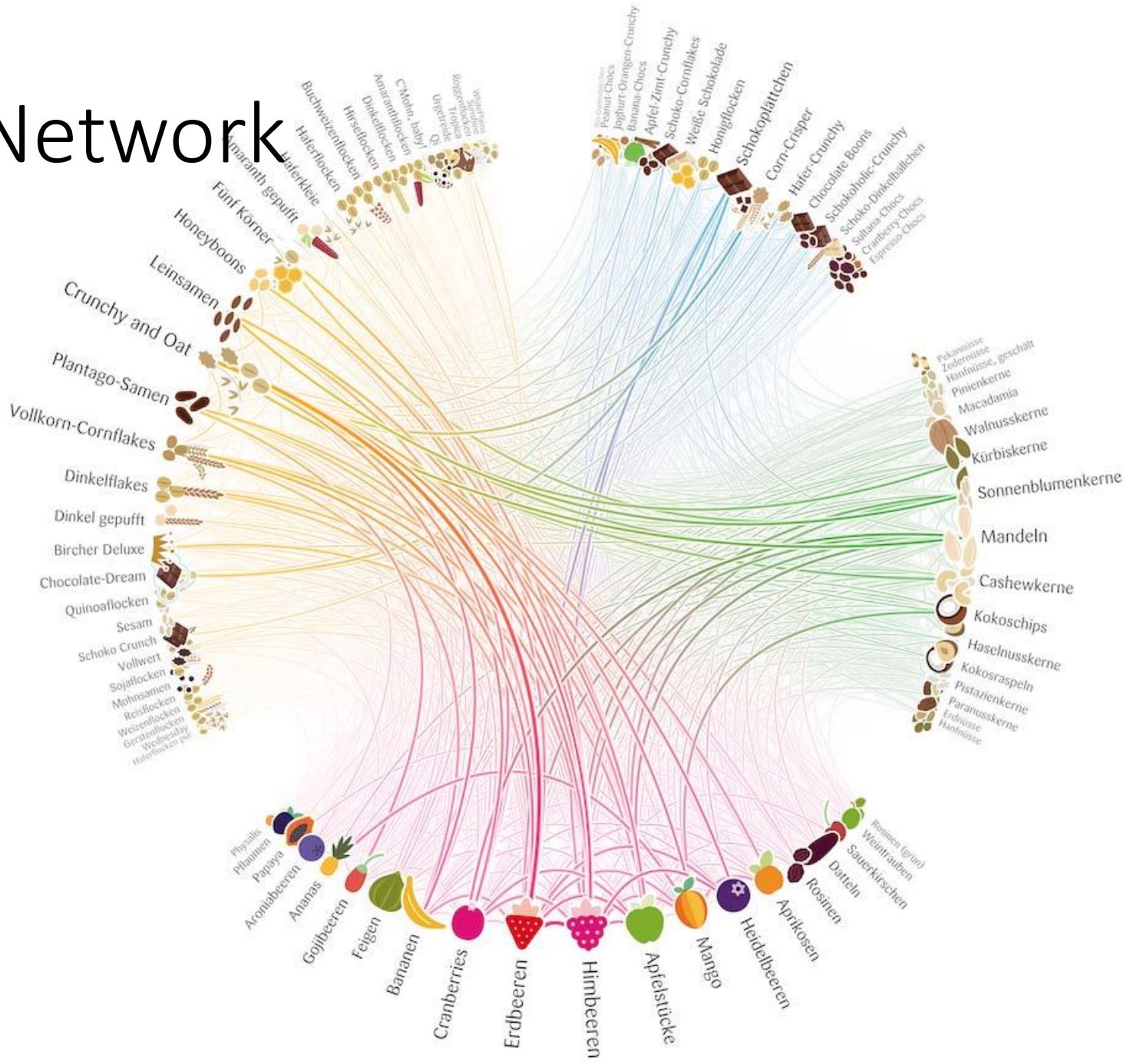
Change Over Time – Flow Map

- Text

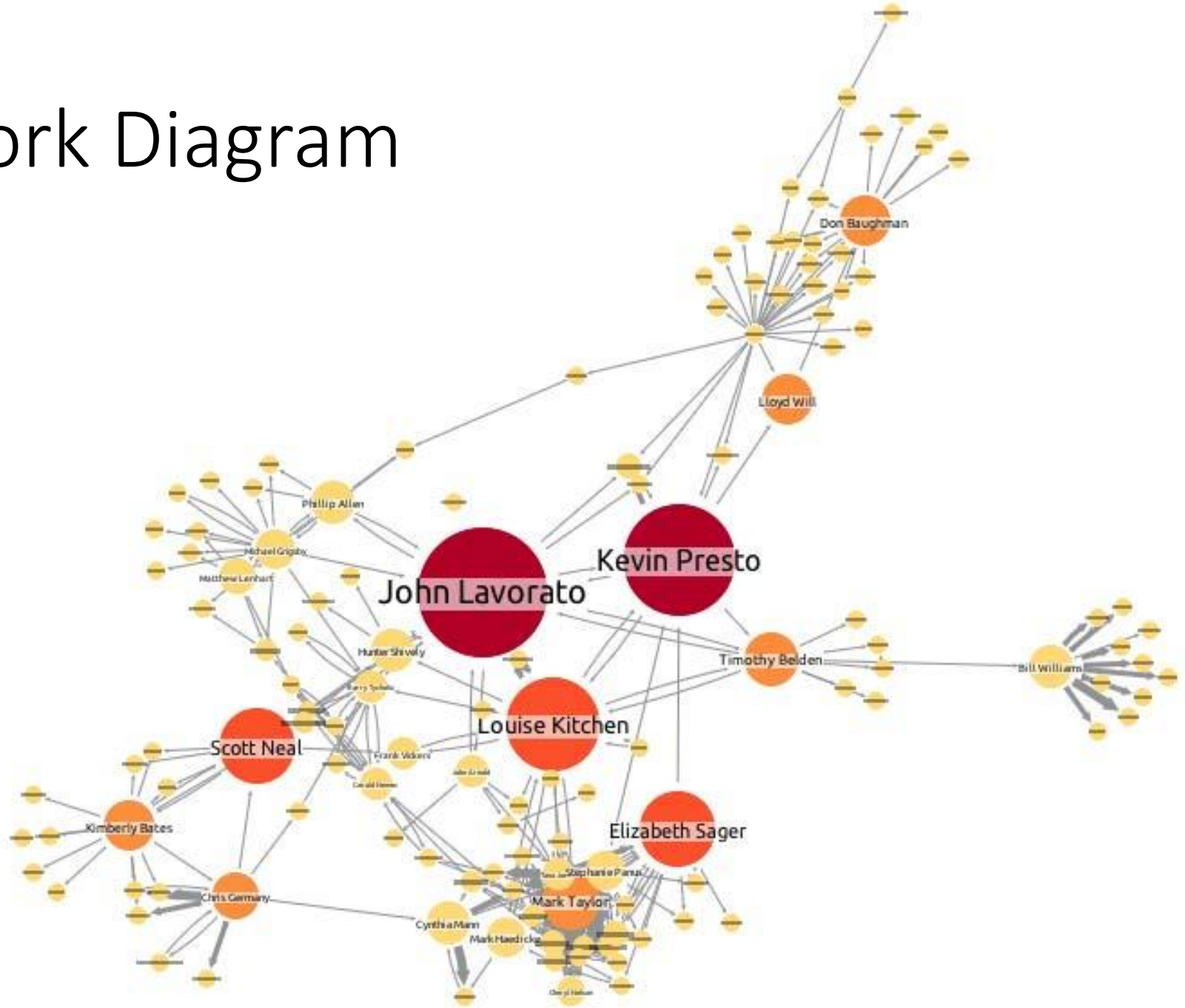


List

Radial Network

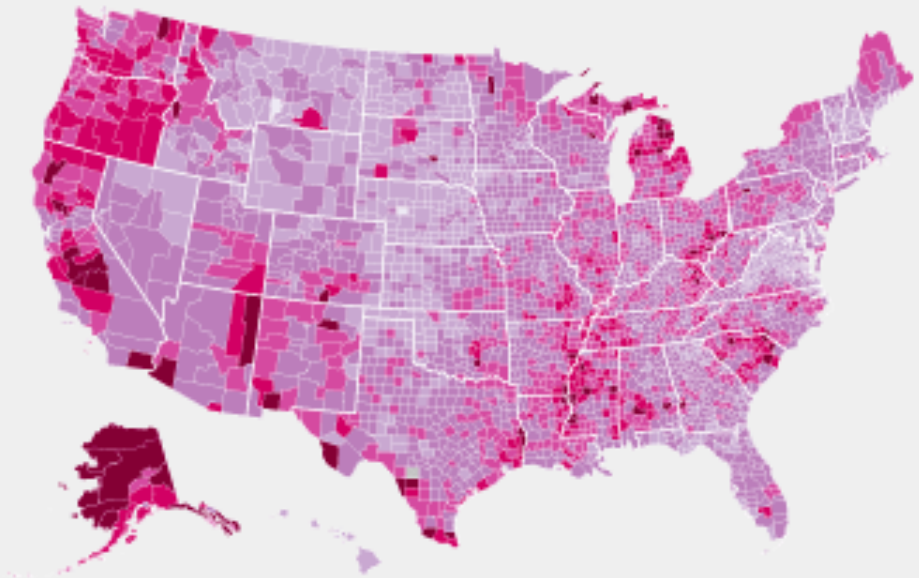


Network Diagram



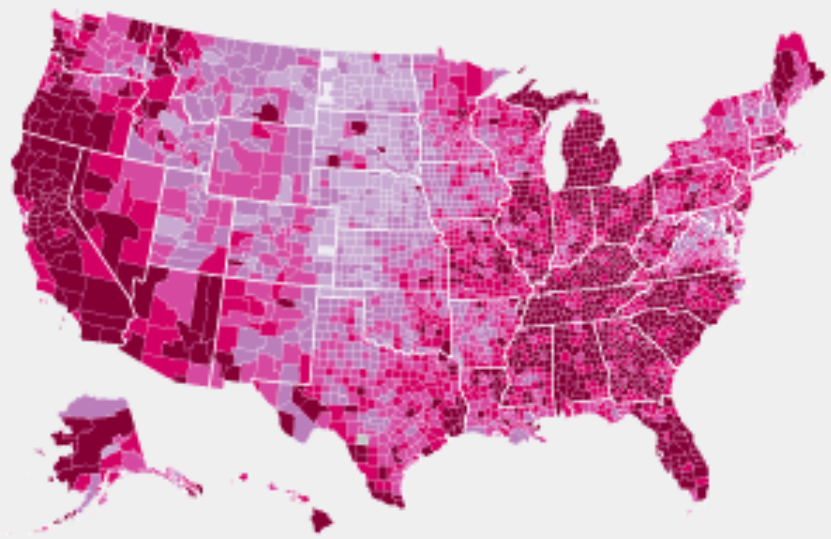
Choropleth Map

2004, 5.5% National Average



Unemployment rose steadily from 2000 to 2004, peaking at 6.3% in June 2003. Rate decreased steadily over the next four years.

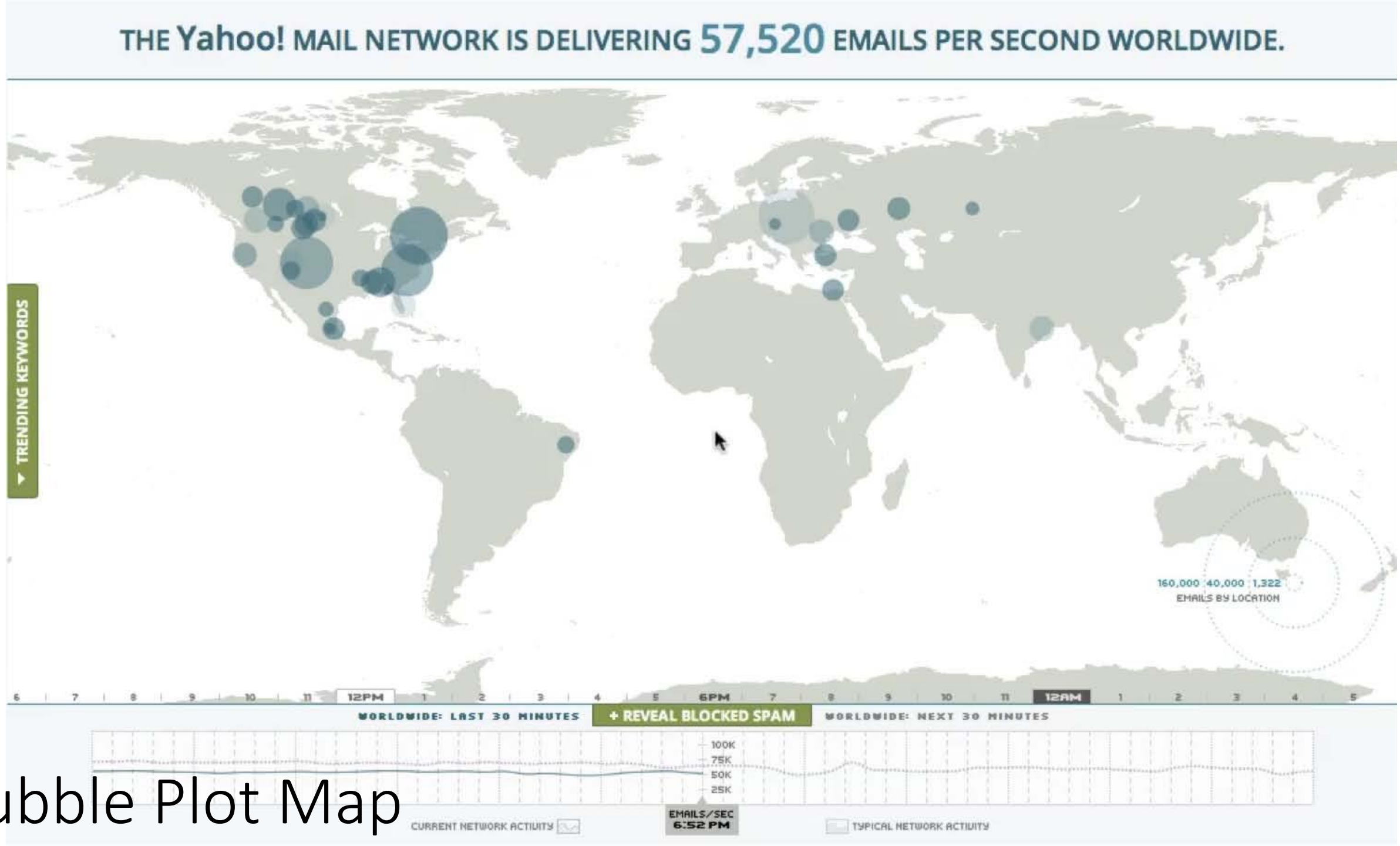
September 2009, 9.8%



The national average rises to the highest it's been since June 1983, when it was 10.1%. Unemployment has increased every month since April 2008 with the exception of one month when it decreased 0.1%.

UNEMPLOYMENT RATE (%)

0 2 4 6 8 10+



Bubble Plot Map

Network Connection Diagram

BIO DIASPORA

The "eight hats" of data visualization design

- Initiator: explorer
- Data Scientist: the data miner, wearing the miner's hat
- Journalist: the storyteller, the person who establishes the narrative approach to the visualization's problem context
- Computer Scientist: the executor, the person who brings the project alive
- Designer: the creative, the one, who, in harmony with the computer scientist, will deliver the solution. Harmony of form vs function
- Cognitive Scientist: the thinker in terms of appreciating the science behind the effectiveness of the technical and designed solutions
- Communicator: naturally, concerned with the communication side of the project
- Project Manager

Numpy

TUGAS BESAR

Pengumpulan di *i-Learn*

TUGAS BESAR

Instruksi

KELAS AVD-A

KELAS AVD-B

Pengumpulan Tugas Besar

Kelas A AVD SI

PENGUMPULAN TUGAS BESAR KELAS A

Tugas Besar untuk Kelas A matakuliah AVD dikumpulkan disini dengan file yang dikumpulkan adalah sebagai berikut:

1. Laporan Akhir (Soft Copy)
2. PPT
3. File Coding
4. Laporan Akhir Hard Copy (Boleh menyusul maksimal pada Pertemuan 4)

Pastikan ada informasi Nama-nama anggota dari kelompoknya dan topik saat dikumpulkan untuk mempermudah untuk pencarian data.

Batas Pengumpulan : 13 November 2024 Pukul 23.59 WIB (Tidak Ada Toleransi)

Kelas B AVD SI

PENGUMPULAN TUGAS BESAR KELAS B

Tugas Besar untuk Kelas B matakuliah AVD dikumpulkan disini dengan file yang dikumpulkan adalah sebagai berikut:

1. Laporan Akhir (Soft Copy)
2. PPT
3. File Coding
4. Laporan Akhir Hard Copy (Boleh menyusul maksimal pada Pertemuan 4)

Pastikan ada informasi Nama-nama anggota dari kelompoknya dan topik saat dikumpulkan untuk mempermudah untuk pencarian data.

Batas Pengumpulan : 14 November 2024 Pukul 23.59 WIB (Tidak Ada Toleransi)

Installation of NumPy

- Install using command: `C:\Users\Your Name>pip install numpy`
- Or using import file in IDE
- Need **import** keyword to use
- Create an alias with keyword **as**

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)
```

Numpy and Array

- Numpy is a Python Library
- Numpy is used for working with arrays
- Numpy is short for Numerical Python

```
arr = np.array([1, 2, 3, 4], ndmin=5)
```

Imort library

Create 1-D array using numpy

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr)
print(type(arr))
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
```

NumPy Array Indexing

- Indexing is accessing an array element
- Indexing started with 0
- For Accessing 2-D Array, using **arr[row,column]**
- Accessing array from the end using negative

```
import numpy as np

arr = np.array([1, 2, 3, 4])

print(arr[1])
```

Matplotlib

Introduction

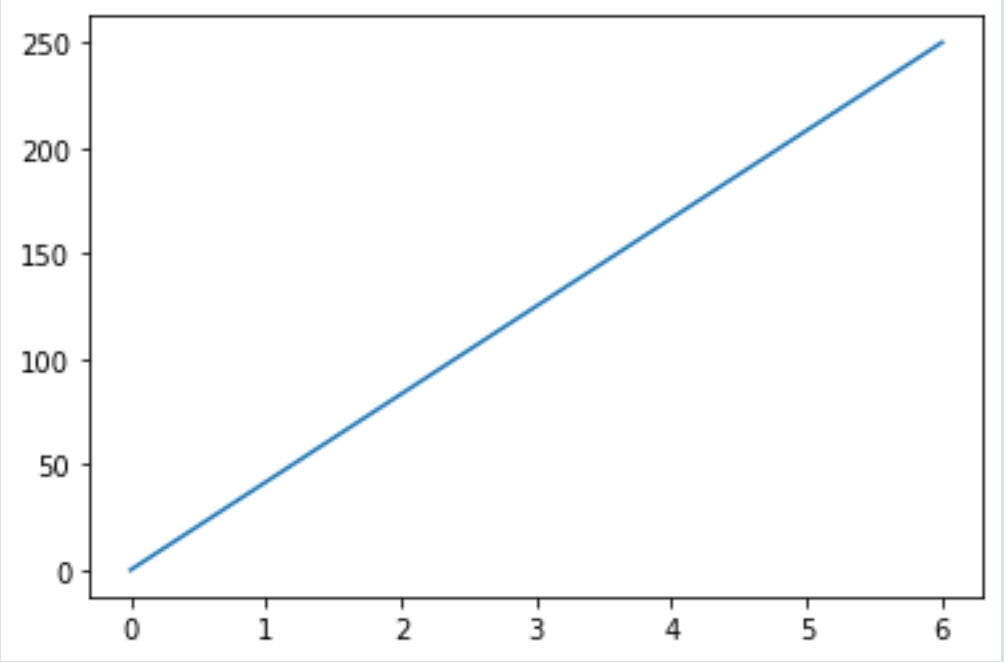
- Matplotlip is a low level graph plotting library for visualization
- Matplotlib created by John D. Hunter and open source so we can use it freely
- Where is the Matplotlib Codebase?
- The source code for Matplotlib is located at this github repository https://github.com/matplotlib/matplotlib
- Installation of Matplotlib can use command or installed on IDE

```
C:\Users\Your Name>pip install matplotlib
```

Using Matplotlib

- Use import keyword
- Matplotlib lies under submodul ***pyplot*** under alias ***plt***

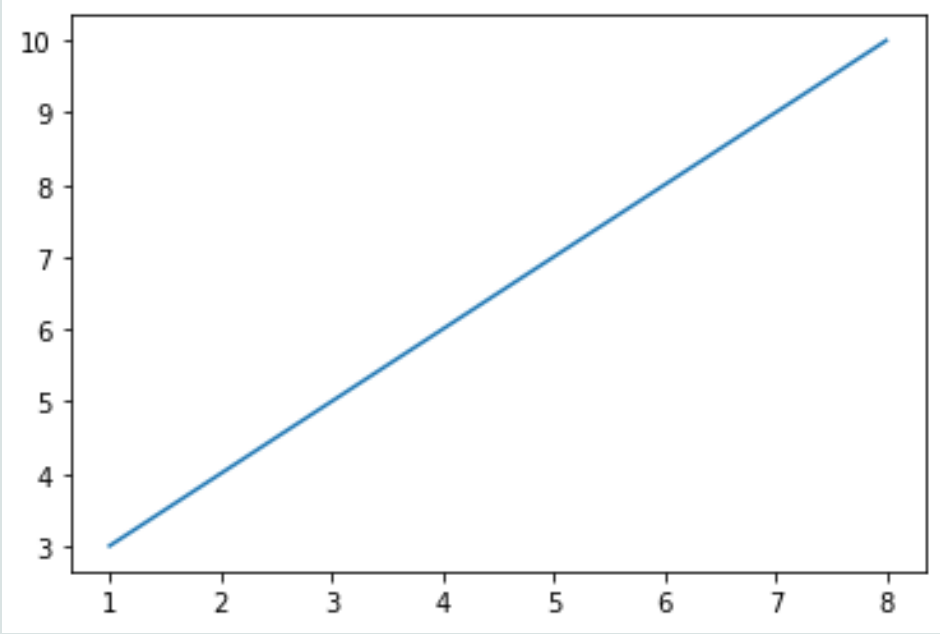
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```



Plotting

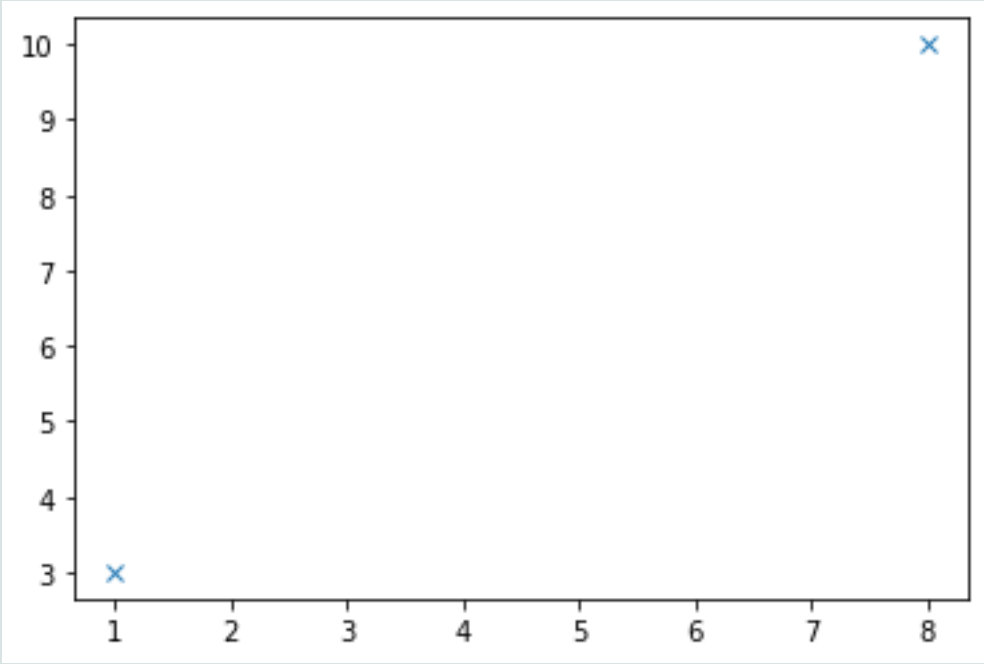
- Using ***plot()*** function to draw points in diagram
- The function ***plot()*** takes parameter that contain of ***x-axis*** and ***y-axis***
- If we want to plot a line from (1,3) to (8,10), then:

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()
```



Plotting Without Line

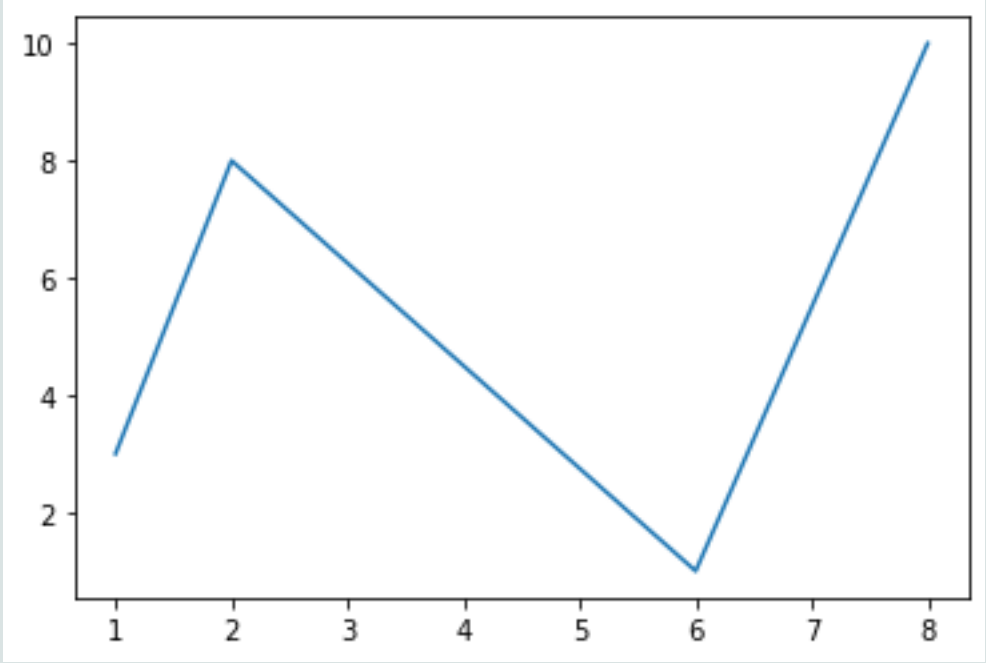
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints, 'x')
plt.show()
```



Draw diagram without line can use any character
S = square, O = circle, D = diamond, H = hexagon, P = pentagon, V = triangle, X = x

Plotting Multiple Points

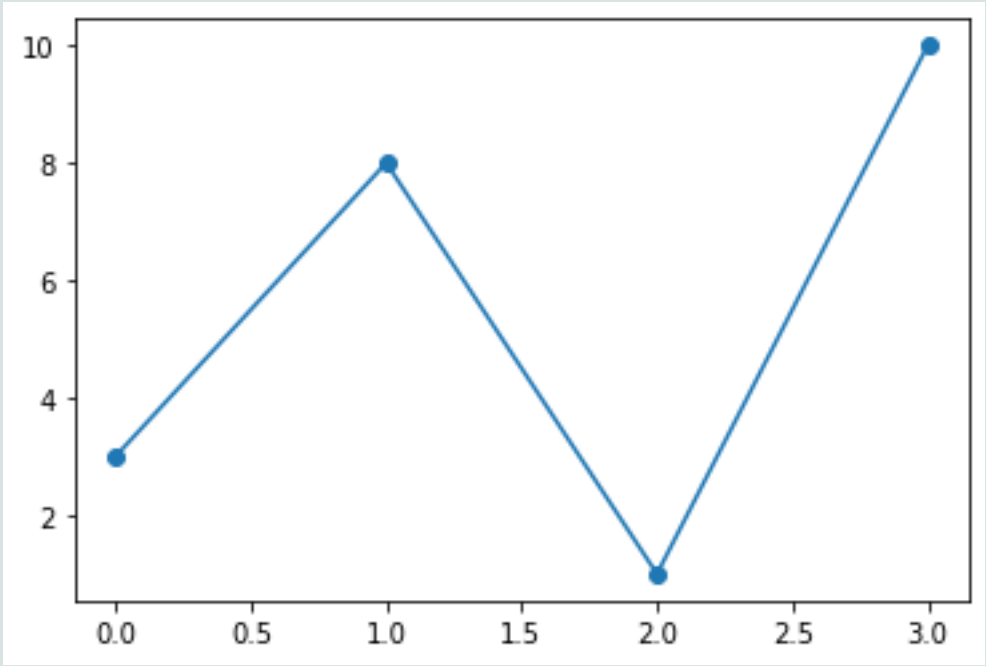
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])

plt.plot(xpoints, ypoints)
plt.show()
```



If we do not specify the points on the x-axis, they will get the default values 0, 1, 2, 3

Markers

```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')
plt.show()
```



Marker References

Marker	Description	
'o'	Circle	Try it »
'*'	Star	Try it »
'.'	Point	Try it »
','	Pixel	Try it »
'x'	X	Try it »
'X'	X (filled)	Try it »
'+'	Plus	Try it »
'P'	Plus (filled)	Try it »
's'	Square	Try it »
'D'	Diamond	Try it »
'd'	Diamond (thin)	Try it »
'p'	Pentagon	Try it »
'H'	Hexagon	Try it »
'h'	Hexagon	Try it »
'v'	Triangle Down	Try it »
'^'	Triangle Up	Try it »
'<'	Triangle Left	Try it »
'>'	Triangle Right	Try it »
'1'	Tri Down	Try it »
'2'	Tri Up	Try it »
'3'	Tri Left	Try it »
'4'	Tri Right	Try it »
'|'	Vline	Try it »
'_'	Hline	

Format Strings

- To specify the marker
- Also called fmt
- Parameter in syntax marker|line|color
- Syntax: **plot(xpoints, ypoints, 'marker|line|color')**

Line Reference

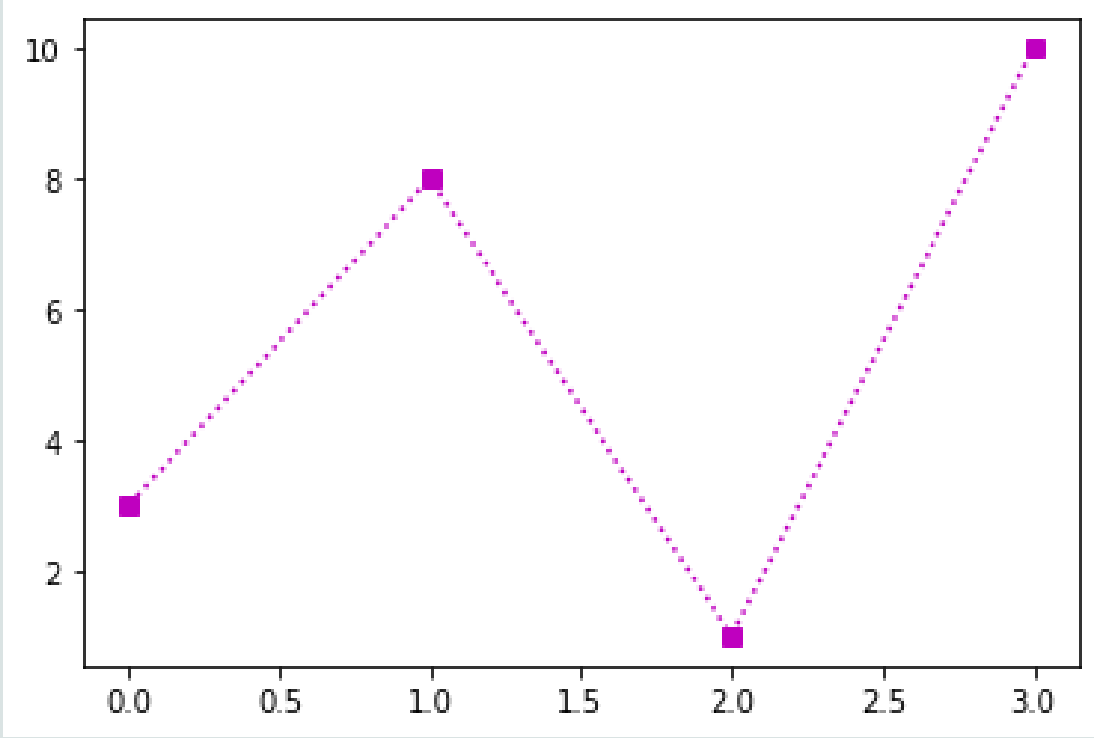
Line Syntax	Description
'-'	Solid line
':'	Dotted line
'--'	Dashed line
'-.'	Dashed/dotted line

Color Reference

Color Syntax	Description
'r'	Red
'g'	Green
'b'	Blue
'c'	Cyan
'm'	Magenta
'y'	Yellow
'k'	Black
'w'	White

Format Strings Example

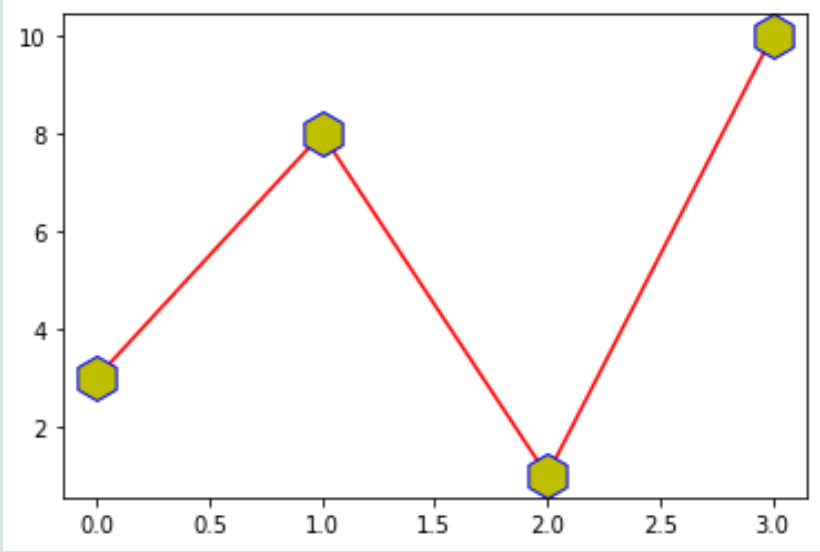
```
import matplotlib.pyplot as plt
import numpy as np

ypoints =
np.array([3, 8, 1, 10])

plt.plot(ypoints, ‘s:m’)
plt.show()
```



Size & Color of Marker & Line

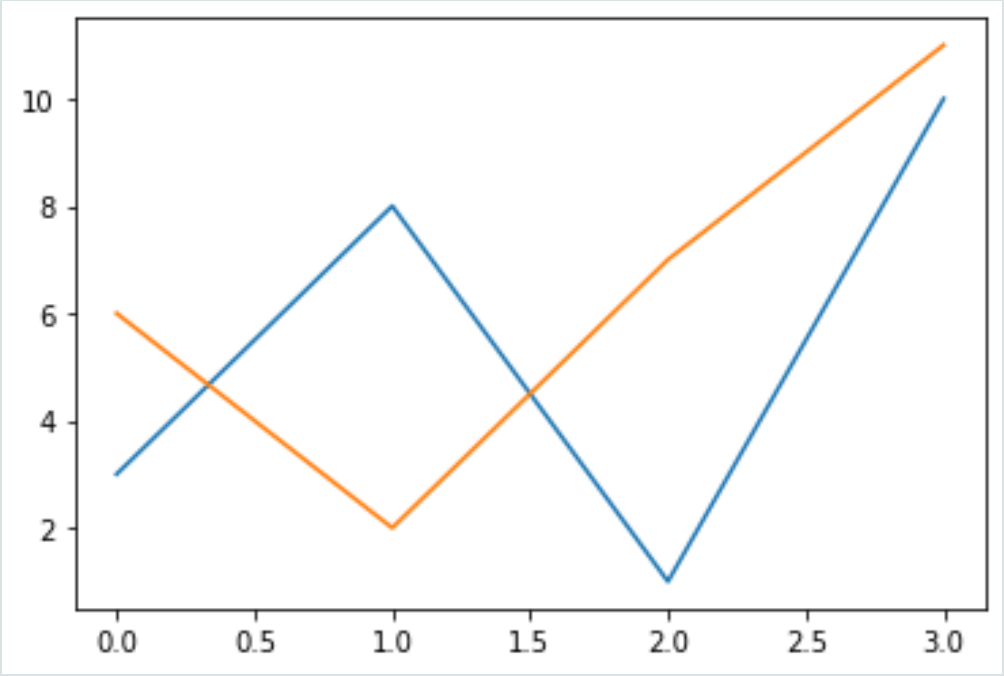
- Use keyword markersize (**ms**) to set the markers, and **mec** (marker edge color) for marker outline color, **mfc** for entire marker color
- You can also use Hexadecimal color values Or any of the 140 supported color names.
- You can use the keyword argument **color** or the shorter **c** to set the color of the line
- You can use the keyword argument **linewidth** or the shorter **lw** to change the width of the line.

```
import matplotlib.pyplot as plt
import numpy as np
ypoints = np.array([3, 8, 1, 10])
plt.plot(ypoints, 'h-r', c='g',
lw='20.5', ms = 20, mec='b', mfc='y')
plt.show()
```



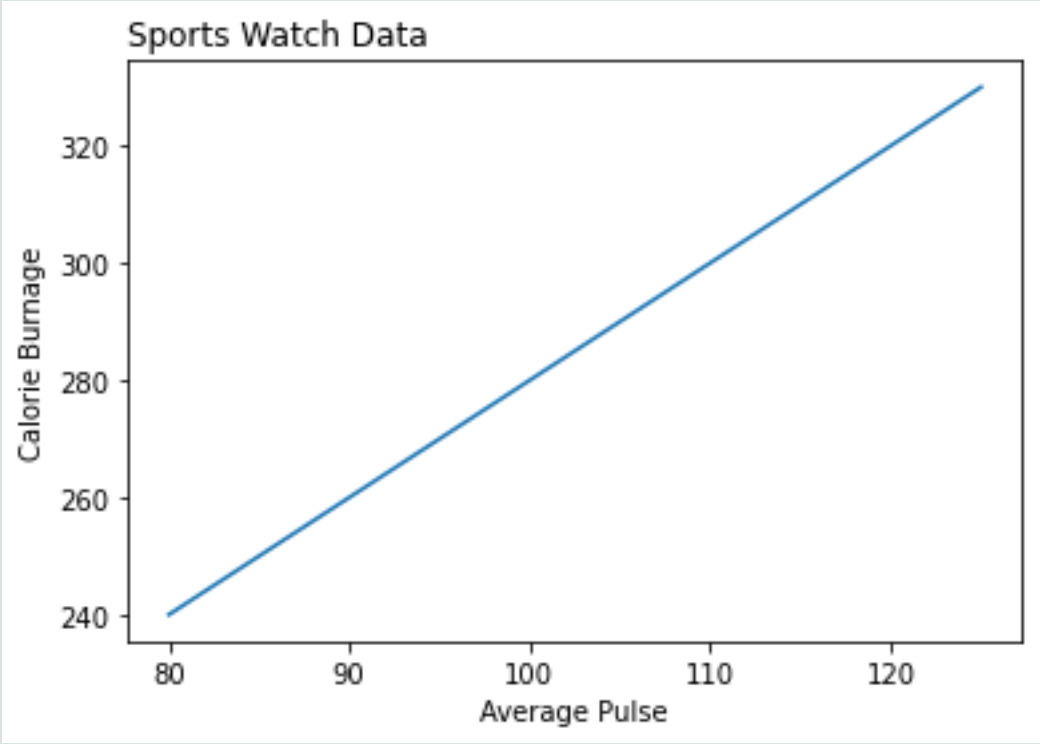
Multiple Line

```
import matplotlib.pyplot as plt
import numpy as np

y1 = np.array([3, 8, 1, 10])
y2 = np.array([6, 2, 7, 11])

plt.plot(y1)
plt.plot(y2)

plt.show()
```



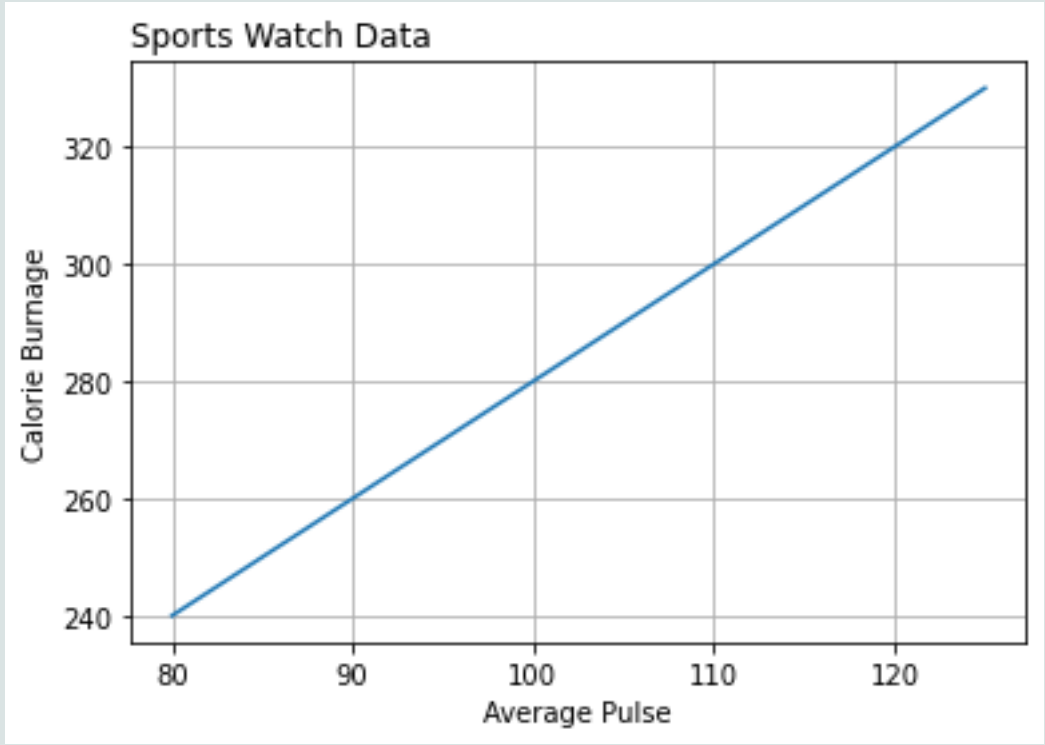
Labels & Title

```
import numpy as np
import matplotlib.pyplot as plt

x =
np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y =
np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 3
30])

plt.title("Sports Watch Data", loc = 'left')
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.plot(x, y)
plt.show()
```



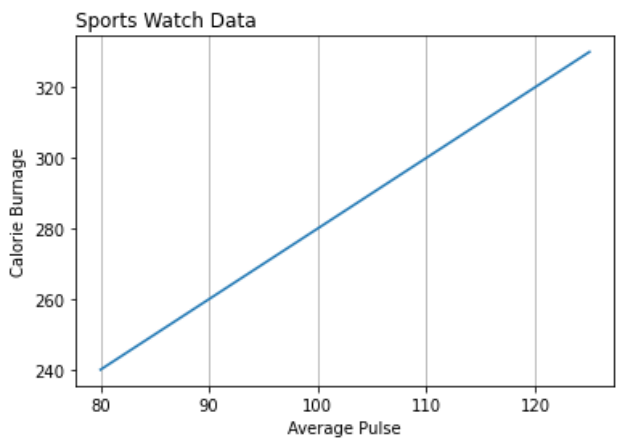
Grid Line

```
import numpy as np
import matplotlib.pyplot as plt

x =
np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y =
np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 3
30])

plt.title("Sports Watch Data", loc = 'left')
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.grid()

plt.plot(x, y)
plt.show()
```

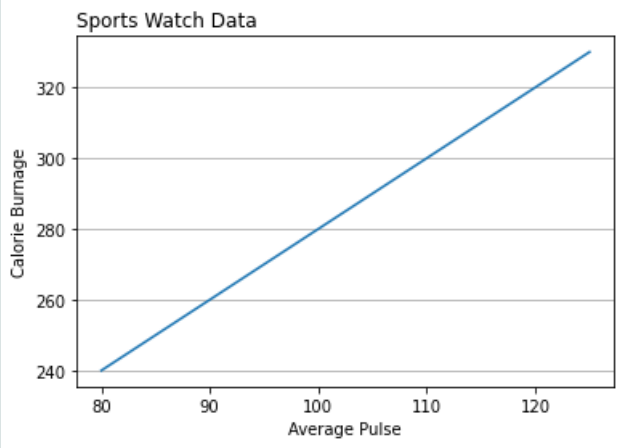


Grid Lines

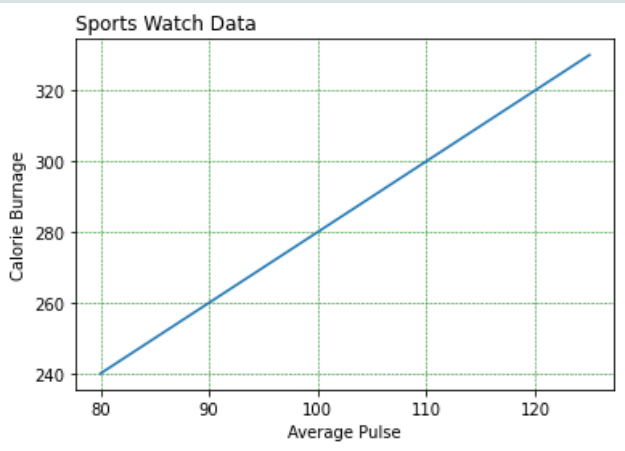
```
plt.grid(axis = 'x')
```




```
plt.grid(axis = 'y')
```



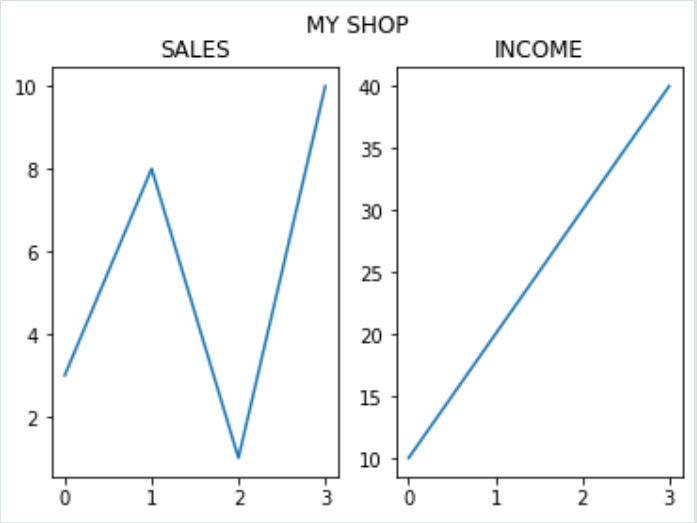

```
plt.grid(color
= 'green', linestyle
= '--', linewidth = 0.5)
```



Subplot

```
import matplotlib.pyplot as plt
import numpy as np

#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(1, 2, 1)
plt.plot(x,y)
plt.title("SALES")

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)
plt.title("INCOME")

plt.suptitle("MY SHOP")
plt.show()
```

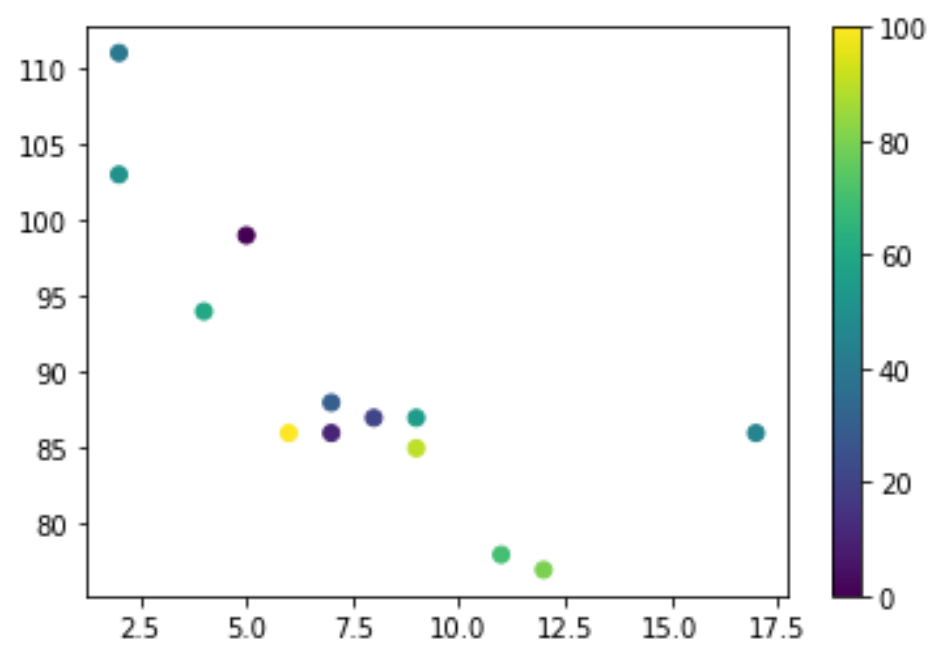
- The subplot() function takes three arguments that describes the layout of the figure.
- The layout is organized in rows and columns, which are represented by the first and second argument.
- The third argument represents the index of the current plot.



Scatter

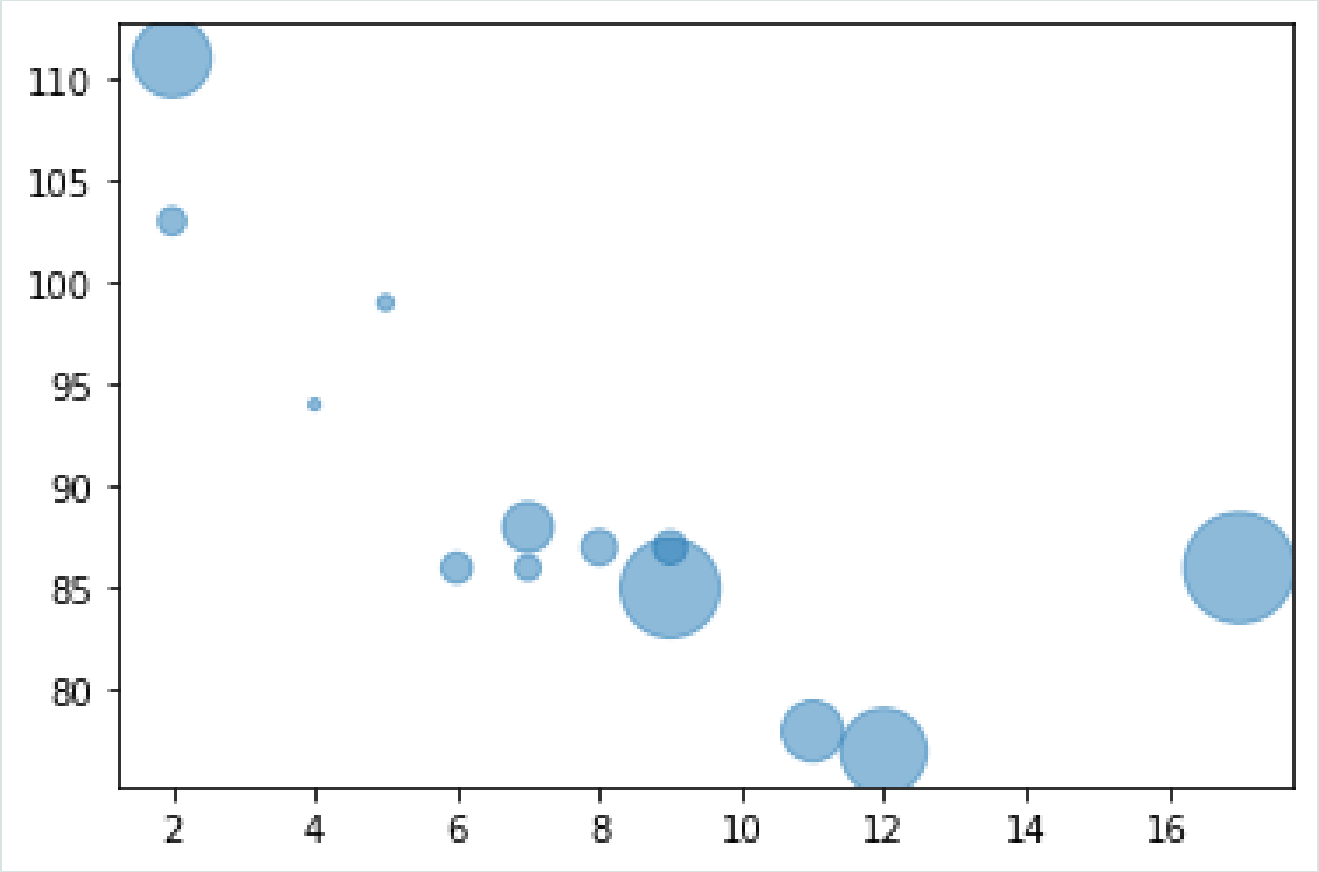
```
import matplotlib.pyplot as plt
import numpy as np

x =
np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y =
np.array([99,86,87,88,111,86,103,87,94,78,
77,85,86])
colors =
np.array([0, 10, 20, 30, 40, 45, 50, 55, 6
0, 70, 80, 90, 100])

plt.scatter(x, y, c=colors,
cmap='viridis')

plt.colorbar()

plt.show()
```

- With Pyplot, you can use the scatter() function to draw a scatter plot.
- The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis



Size

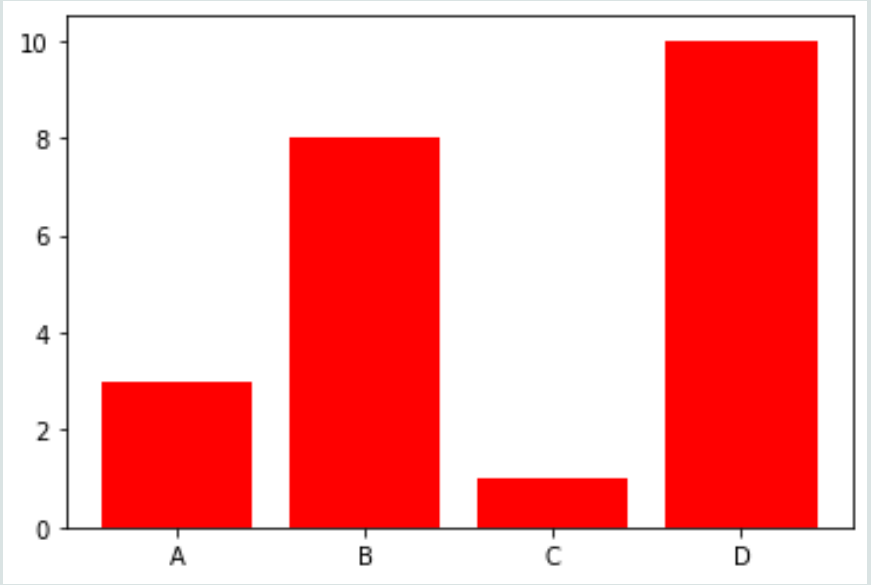
```
import matplotlib.pyplot as plt
import numpy as np

x =
np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y =
np.array([99,86,87,88,111,86,103,87,94,78,
77,85,86])
sizes
= np.array([20,50,100,200,500,1000,60,90,1
0,300,600,800,75])

plt.scatter(x, y, s=sizes, alpha=0.5)

plt.show()
```



Bar

- If you want the bars to be displayed horizontally instead of vertically, use the barh()
- The bar() takes the keyword argument width to set the **width** of the bars
- The barh() takes the keyword argument height to set the **height** of the bars

```
import matplotlib.pyplot as plt
import numpy as np

x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])

plt.bar(x,y, color = "red")
plt.show()
```



Challenges!!

Learn how to show Histogram and Pie Chart in Python!!

3D CHART
VISUALIZATION USING PYTHON & MYSQL

3D VISUALIZATION USING PYTHON

Rahmatika Pratama Santi

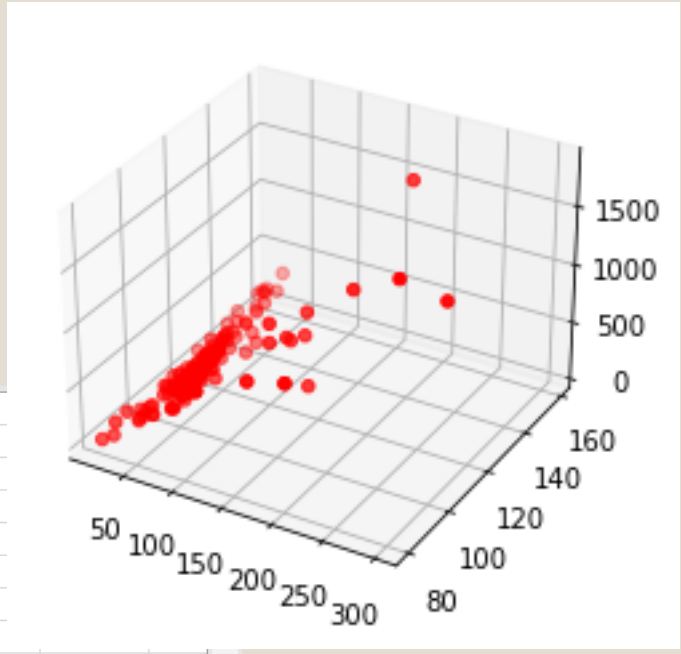
Library

- Matplotlib
- Pandas

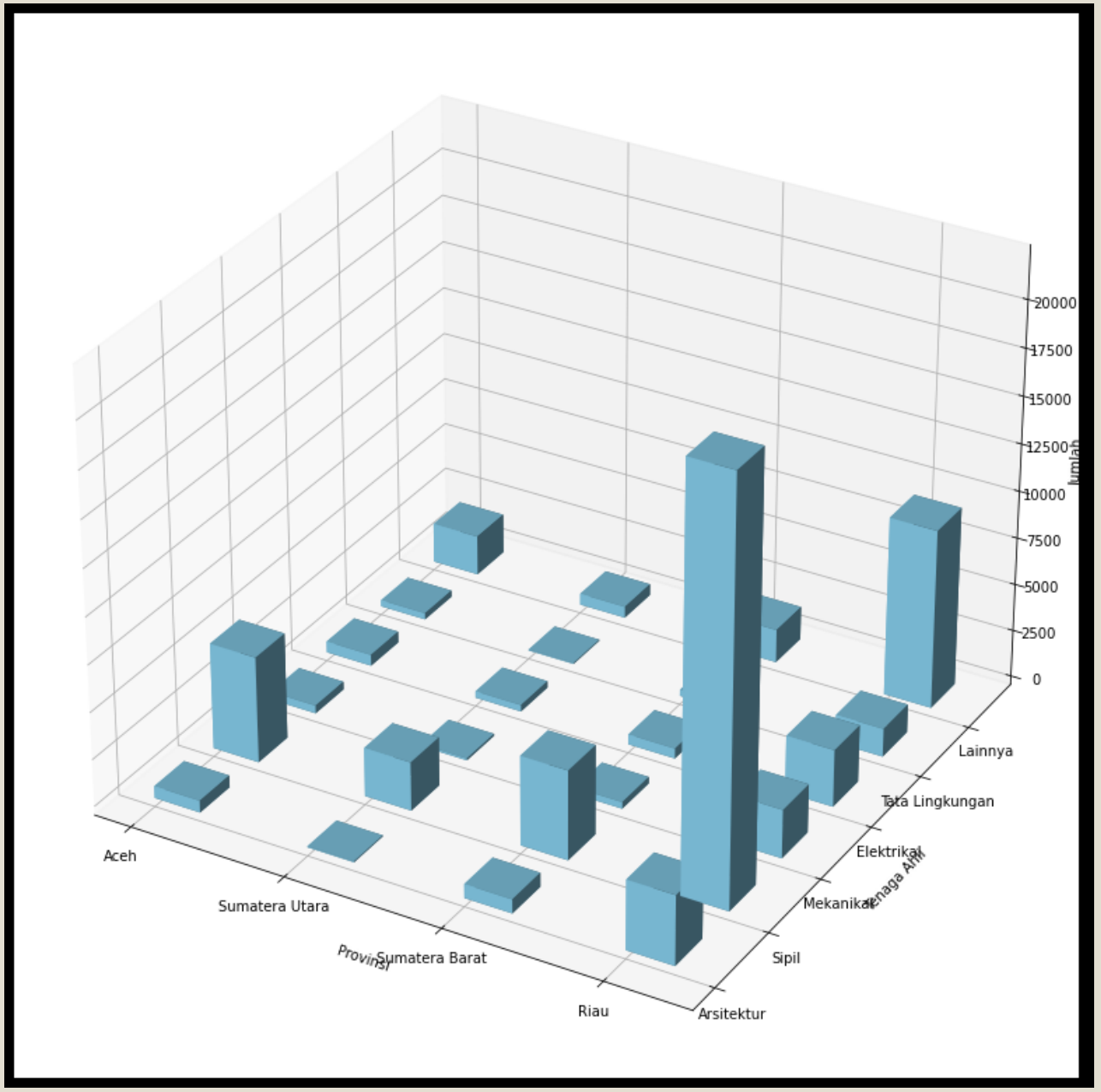
Scatter, Praktikkan di Laptop

```
import matplotlib.pyplot as plt
import pandas
from mpl_toolkits.mplot3d import Axes3D

data =
pandas.read_csv(r'D:\...\data.csv')

fig = plt.figure()
ax = fig.add_subplot(111,
projection='3d')

x = data['Duration'].values
y = data['Pulse'].values
z = data['Calories'].values

ax.scatter(x, y, z, c='r', marker='o')

plt.show()
```

	Duration	Pulse	Maxpulse	Calories
2	60	110	130	409.1
3	60	117	145	479
4	60	103	135	340
5	45	109	175	282.4
6	45	117	148	406
7	60	102	127	300
8	60	110	136	374
9	45	104	134	253.3
10	30	109	133	195.1
11	60	98	124	269
12	60	103	147	329.3
13	60	100	120	250.7
14	60	106	128	345.3
15	60	104	132	379.3
16	60	98	123	275
17	60	98	120	215.2
18	60	100	120	300
19	45	90	112	
20	60	103	123	323
21	45	97	125	243
22	60	108	131	364.2
23	45	100	119	282



Bar


20000
17500
15000
12500
10000
7500
5000
2500
0
Jumlah
Aceh
Sumatera Utara
Sumatera Barat
Riau
Provinsi
Arsitektur
Sipil
Mekanikal
Elektrikal
Tata Lingkungan
Lainnya

Import Library

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
import pandas as pd
```

Create Chart

```
fig = plt.figure(facecolor="Black",
figsize=(20,14))

ax = plt.axes(projection="3d")
```

Import and Create Data

```
data = pd.read_csv(r’D:\...\tenagaahli.csv')
df=pd.DataFrame(data)

daerah=['Aceh','Sumatera Utara','Sumatera Barat','Riau']
ahli=['Arsitektur','Sipil','Mekanikal','Elektrikal', 'Tata
Lingkungan', 'Lainnya']
```

Create Coordinate Position

```
numOfCols = 4
numOfRows = 6

xpos = np.arange(0, numOfCols, 1)
ypos = np.arange(0, numOfRows, 1)
xpos, ypos = np.meshgrid(xpos + 1, ypos + 1)

xpos = xpos.flatten()
ypos = ypos.flatten()
zpos = np.zeros(numOfCols * numOfRows)
```

Create Bar Size and Value

```
dx = np.ones(numOfRows * numOfCols) * 0.3
dy = np.ones(numOfCols * numOfRows) * 0.5
zsize = df['Jumlah']
```

Customize The Chart

```
ax.bar3d(xpos, ypos, zpos, dx, dy, zsize, color='skyblue')

ticks_x = np.arange(1, 5, 1)
ax.set_xticks(ticks_x)
ticks_y=np.arange(1,7,1)
ax.set_yticks(ticks_y)

ax.set_xlabel('Provinsi')
ax.set_ylabel('Tenaga Ahli')
ax.set_zlabel('Jumlah')
ax.set_xticklabels(daerah)
ax.set_yticklabels(ahli)
plt.show()
```

PYTHION & MYSQL

Connection

```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
    )

print(mydb)
```

Check DB

```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
    )

mycursor = mydb.cursor()
mycursor.execute("SHOW DATABASES")
for x in mycursor:
  print(x)
```

Create DB

```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
    )

mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE
mydatabase")
```

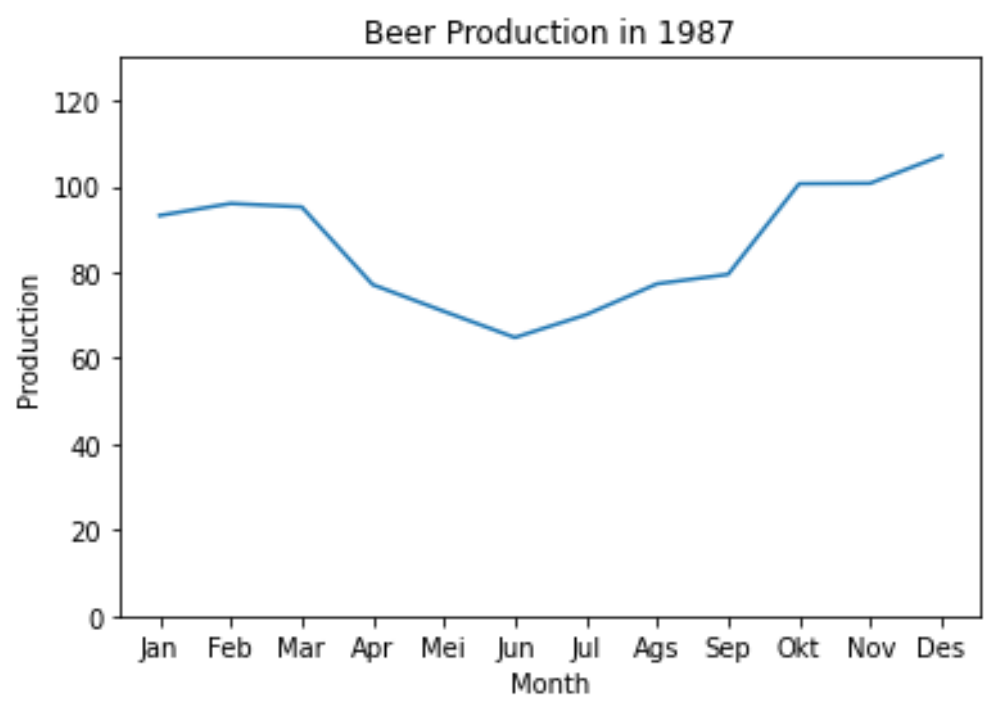
Create Table

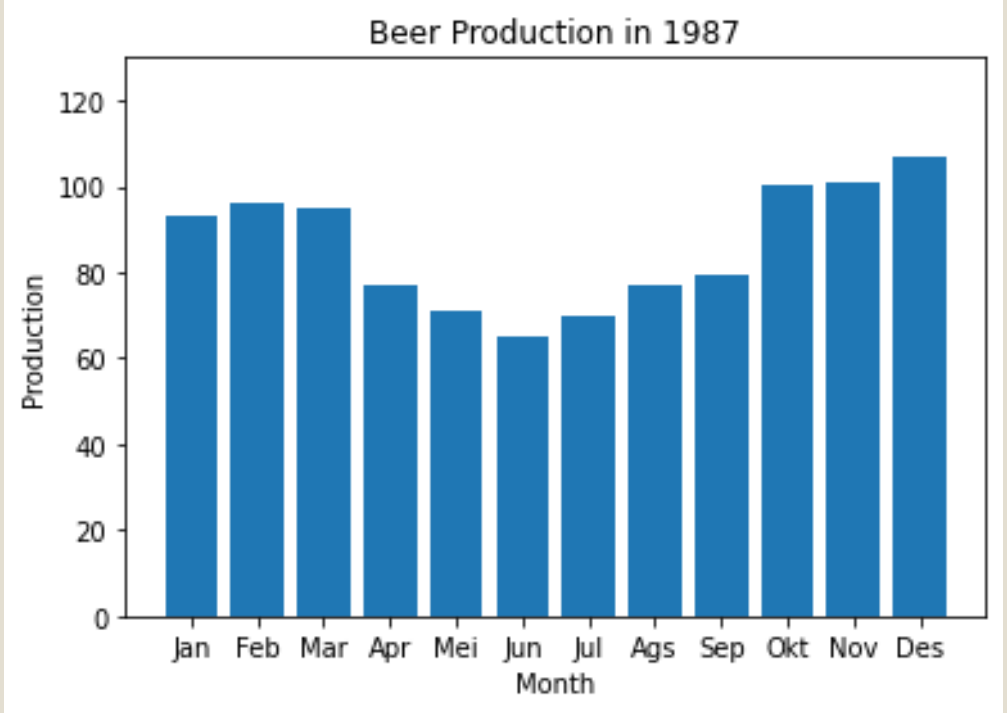
```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password=""
    )

mycursor = mydb.cursor()
mycursor.execute(""CREATE TABLE customers (name
VARCHAR(255), address VARCHAR(255))")
```

SQL for Manipulating

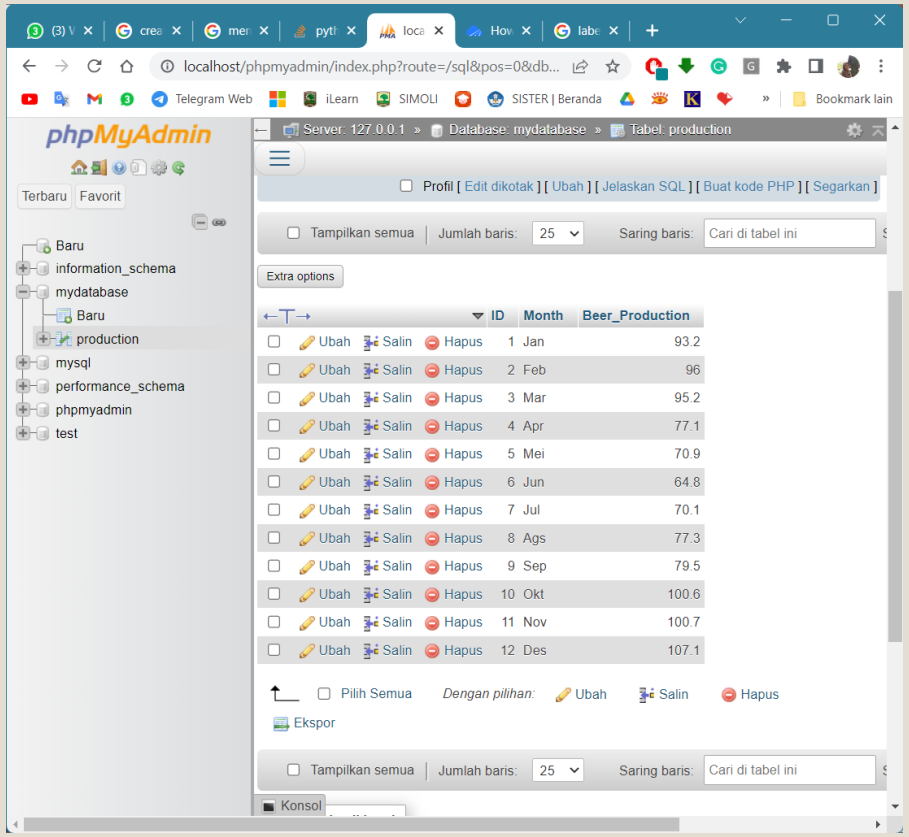
- INSERT
- SELECT
- UPDATE
- DELETE

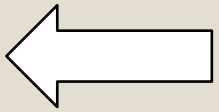
Visualize The Data

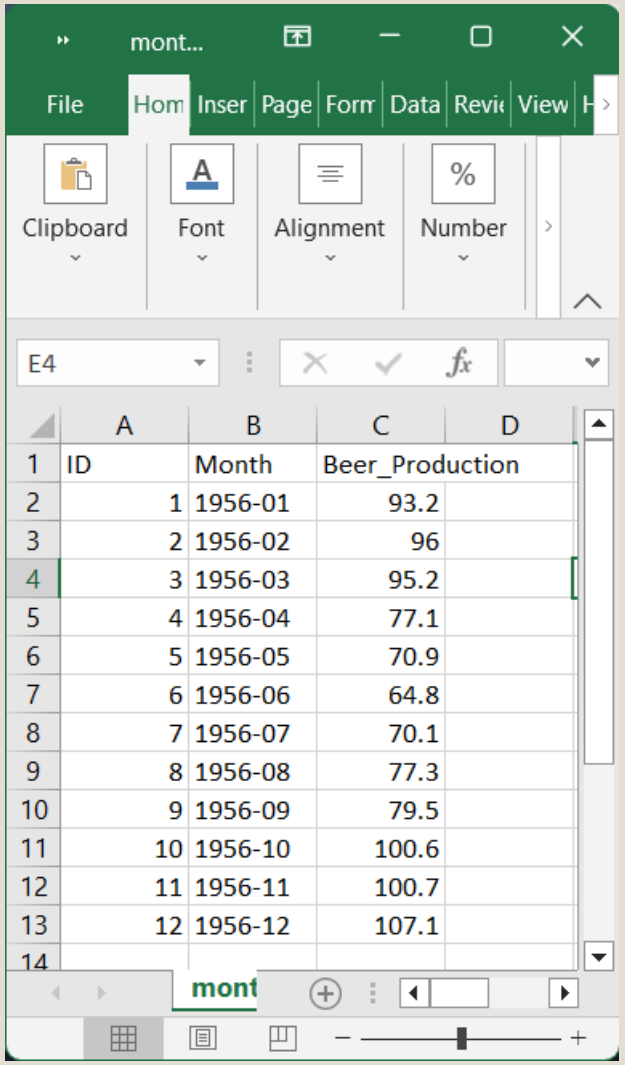




Insert Data Into DB







	A	B	C	D
1	ID	Month	Beer_Production	
2	1	1956-01	93.2	
3	2	1956-02	96	
4	3	1956-03	95.2	
5	4	1956-04	77.1	
6	5	1956-05	70.9	
7	6	1956-06	64.8	
8	7	1956-07	70.1	
9	8	1956-08	77.3	
10	9	1956-09	79.5	
11	10	1956-10	100.6	
12	11	1956-11	100.7	
13	12	1956-12	107.1	
14				

How?

1. Import Library
2. Create Connection
3. Access Data

```
mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM production")

myresult = mycursor.fetchall()
```

How?

4. Append to Array

```
Month = []
Production = []

for x in myresult:
  Month.append(x[1])
  Production.append(x[2])
```

How?

5. Create Visualization

```
plt.plot(Month, Production)
plt.ylim(0, 130)
plt.xlabel("Month")
plt.ylabel("Production")
plt.title("Beer Production in
1987")
plt.show()
```

STUDI KASUS

INSTRUKSI

1

Lakukan Preprocessing
See more

2

Bangun visualisasi dengan Power BI berdasarkan data yang sudah di bersihkan

3

Lakukan analisis visual

4

Paparkan di depan kelas

Instruksi 1

- Pada data yang diberikan, lakukan
  - Seleksi:
    - Pilih 5 Data setiap Tahun dari data tahun 2001 hingga 2020
    - Total akhir data adalah 50 row
    - Coloumn yang digunakan hanya List Year, Town, Address, Assessed Value, Sale Amount, Sales Ratio
  - Cleaning:
    - Lakukan cleaning pada data
    - Jika ada data kosong maka lakukan pengisian data (sesuai kategori tersedia), contoh:
      - Column Properies yang bernilai kosong di isi dengan Other
  - Transformation
    - Transformasikan data jika da format yang belum seragam dan sesuai

STUDI KASUS (LAGI)

Visualisasi IP (Tugas Individu)

- Buat data IP semester Anda sejak semester 1 hingga semester ini pada database

Semester	IP
Semester 1	3.47
Semester 2	3.35
Semester …	…

- Visualisasikan IP anda per semester menggunakan python dalam bentuk visualisasi 3D
- Import data dari Database bukan menggunakan Numpy ataupun array lainnya
- Pengumpulan: 13 Juni 2023
- Print Out berisikan:
  - Data
  - Codingan
  - Grafik