# CSE 564
# Visualization and Visual Analytics

## Visual Design & Aesthetics

Klaus Mueller

Computer Science Department
Stony Brook University

| Lecture | Topic | Projects |
|---|---|---|
| 1 | Intro, schedule, and logistics | |
| 2 | Applications of visual analytics, basic tasks, data types | |
| 3 | Introduction to D3, basic vis techniques for non-spatial data | Project #1 out |
| 4 | Data assimilation and preparation | |
| 5 | Bias in visualization | |
| 6 | Data reduction and dimension reduction | |
| 7 | Visual perception and cognition | Project #1 due |
| 8 | Visual design and aesthetics | Project #2 out |
| 9 | Data mining techniques: clusters, text, patterns, classifiers | |
| 10 | Data mining techniques: clusters, text, patterns, classifiers | |
| 11 | Computer graphics and volume rendering | |
| 12 | Techniques to visualize spatial (3D) data | Project #2 due |
| 13 | Scientific and medical visualization | Project #3 out |
| 14 | Scientific and medical visualization | |
| 15 | Midterm #1 | |
| 16 | High-dimensional data, dimensionality reduction | Project #3 due |
| 17 | Big data: data reduction, summarization | |
| 18 | Correlation and causal modeling | |
| 19 | Principles of interaction | |
| 20 | Visual analytics and the visual sense making process | Final project proposal due |
| 21 | Evaluation and user studies | |
| 22 | Visualization of time-varying and time-series data | |
| 23 | Visualization of streaming data | |
| 24 | Visualization of graph data | Final Project preliminary report due |
| 25 | Visualization of text data | |
| 26 | Midterm #2 | |
| 27 | Data journalism | |
| | Final project presentations | Final Project slides and final report due |

# Three Key Visual Representations

Gestalt Principles:

- the tendency to perceive elements as belonging to a group, based on certain visual properties

Pre-attentiveness:

- certain low level visual aspects are recognized before conscious awareness

Visual variables:

- the different visual aspects that can be used to encode information

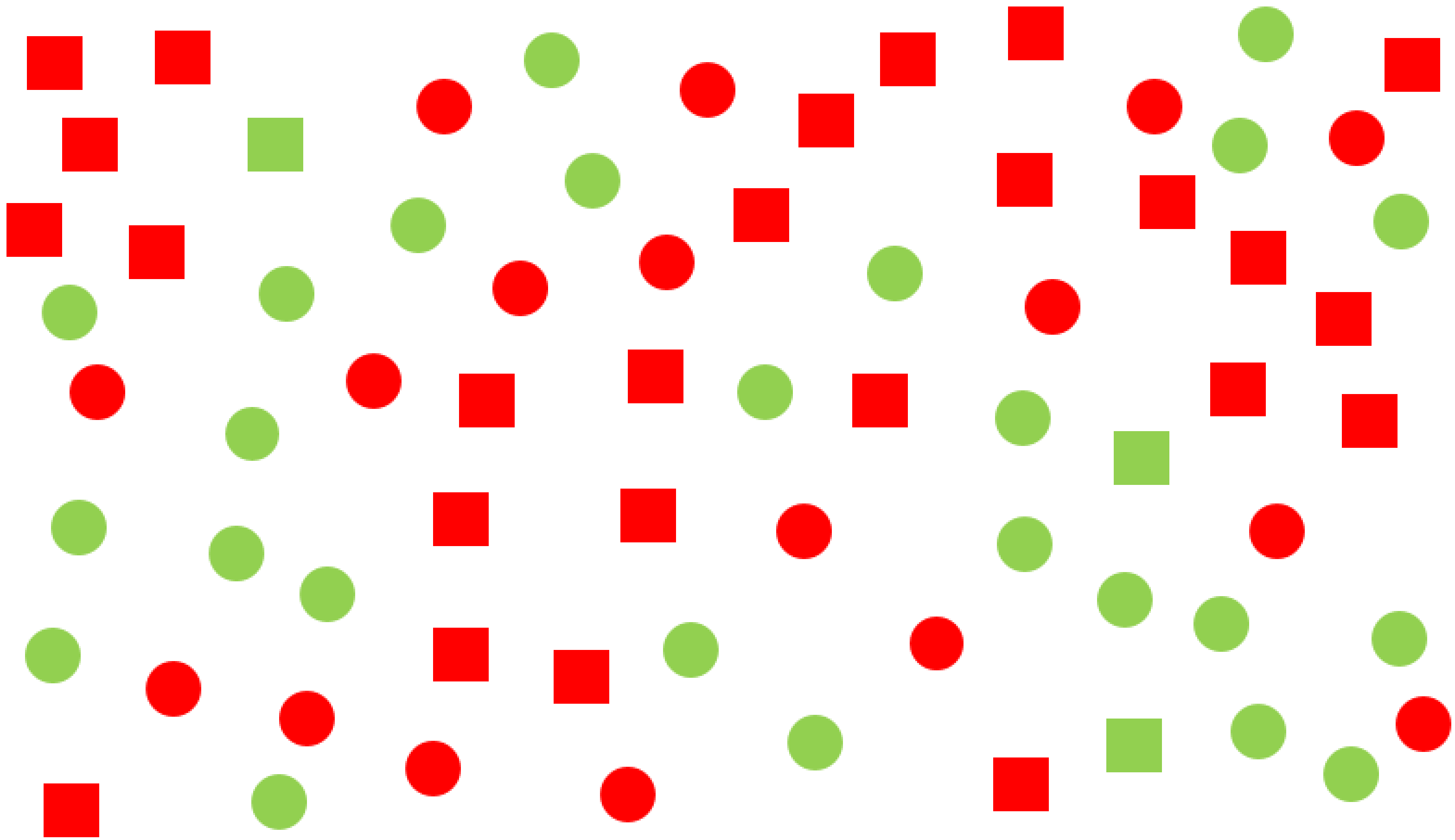
# Gestalt

Concept of totality

- you grasp the "totality" of something before worrying about the details

# Pre-Attentiveness

Also called pop-out (multiple conjunctions shown here):



# Which Popped-out Faster

Color (red vs. green)

Shape (circle vs. square)

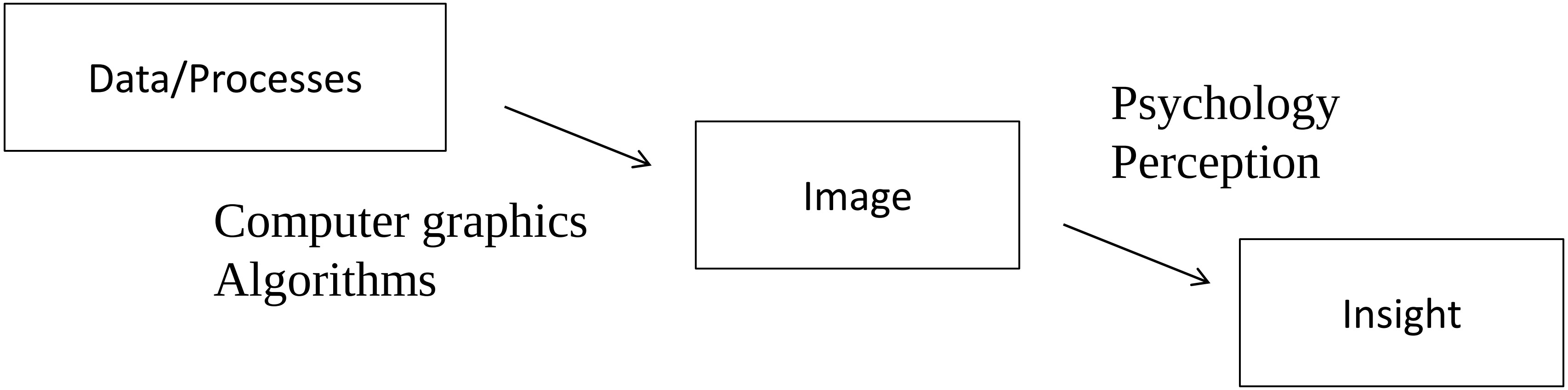
popup

# Visual Variables

Formal theory linking perception to visualization

Established by Jacques Bertin (1967)

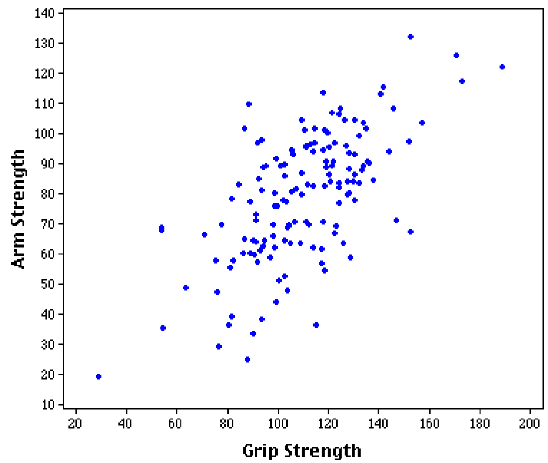
- he called it 'Image Theory'
- original book in French (*Sémiologie Graphique*) translated into English by W. Berg (1983)
- not formally linked to vision research more based on intuition
- but has been shown later by M. Green to be quite accurate



# Visual Variables

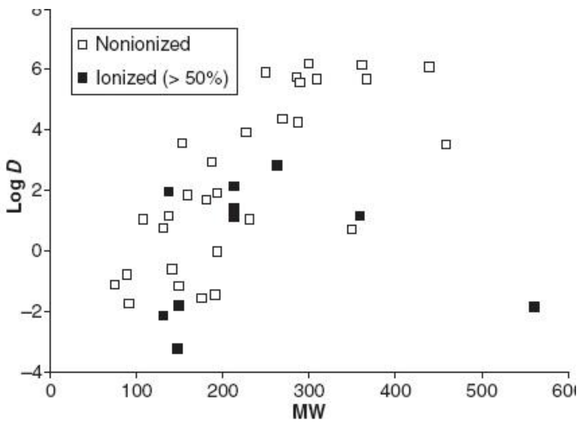
Two planar variables

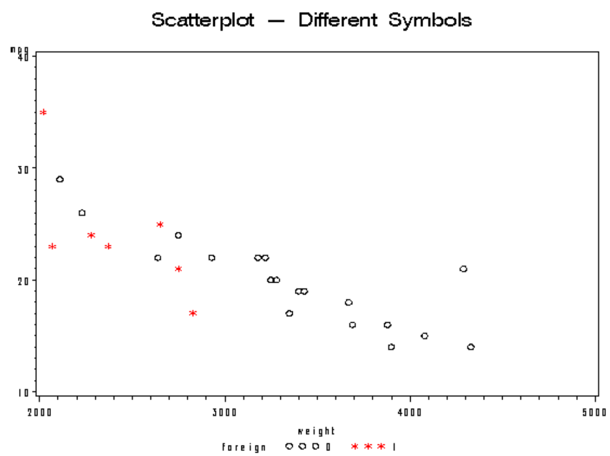
- spatial dimensions
- map (arm, grip) to (x,y)




Six retinal variables

- size
- color
- shape
- orientation
- texture
- brightness




Retinal variables allow for one more variable to be encoded

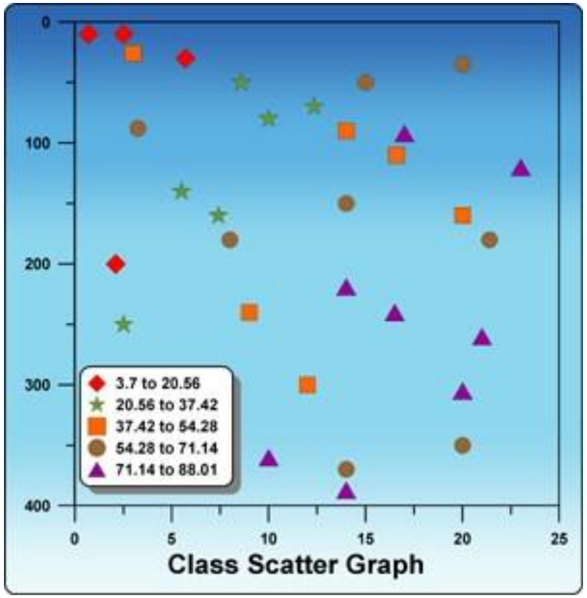
- more than three variables will hamper efficient visual search
- recall low decoding speed of conjunctions

# Associative vs. Selective

Both are nominal qualities

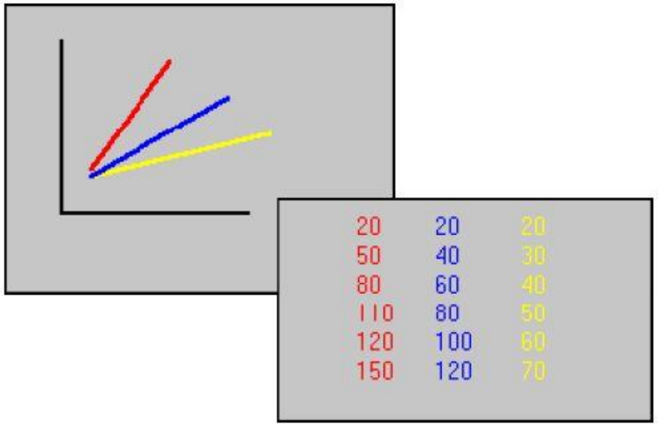
Associative

- lowest organizational level
- enables **visual grouping** of all elements of a variable



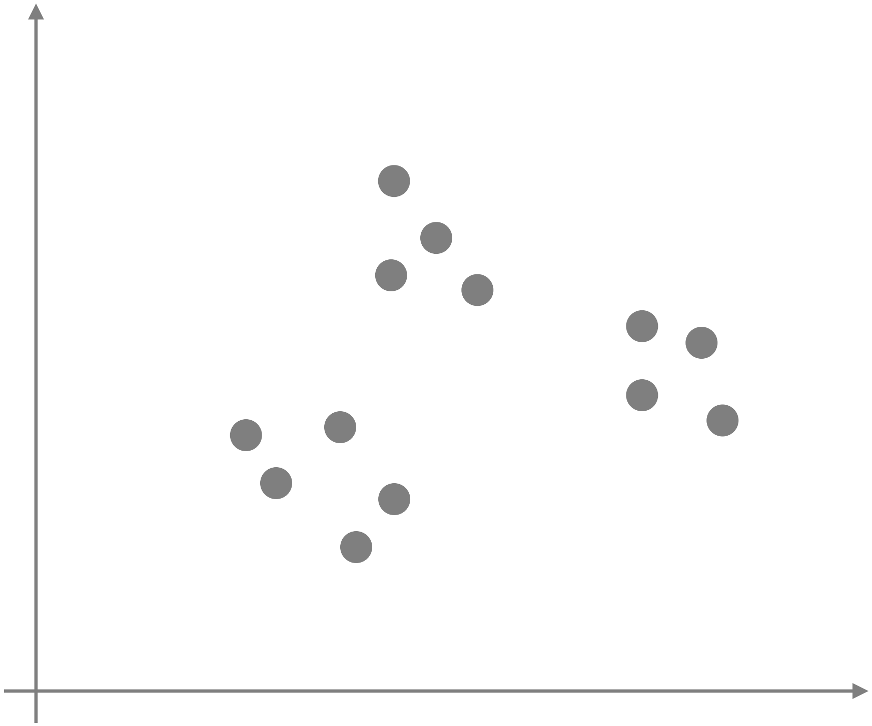

Selective

- next lowest level
- enables viewer to **isolate encoded data** and ignore others



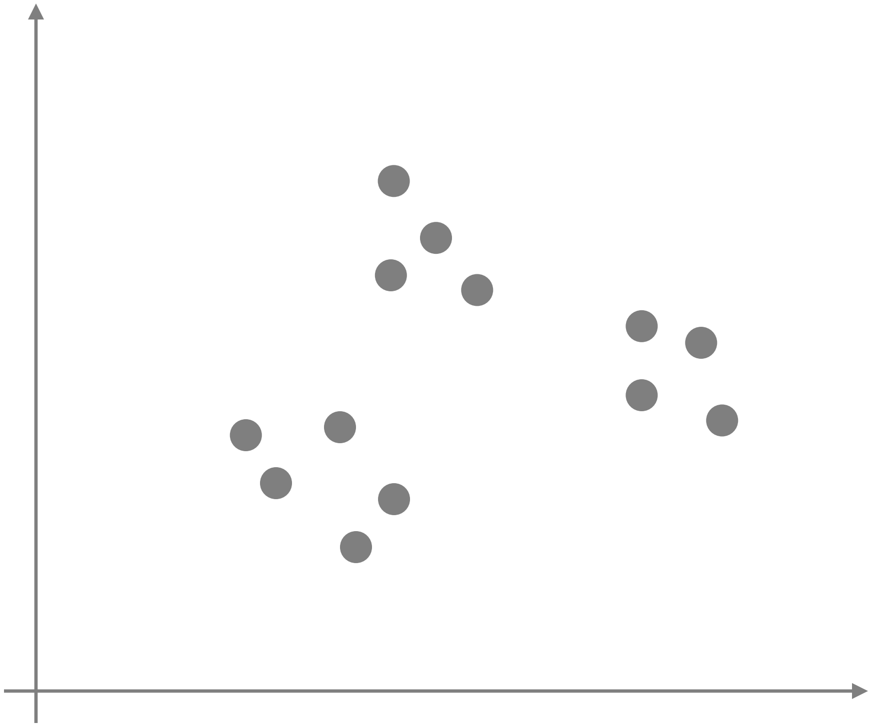
# Visual variable #1 – Planar

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



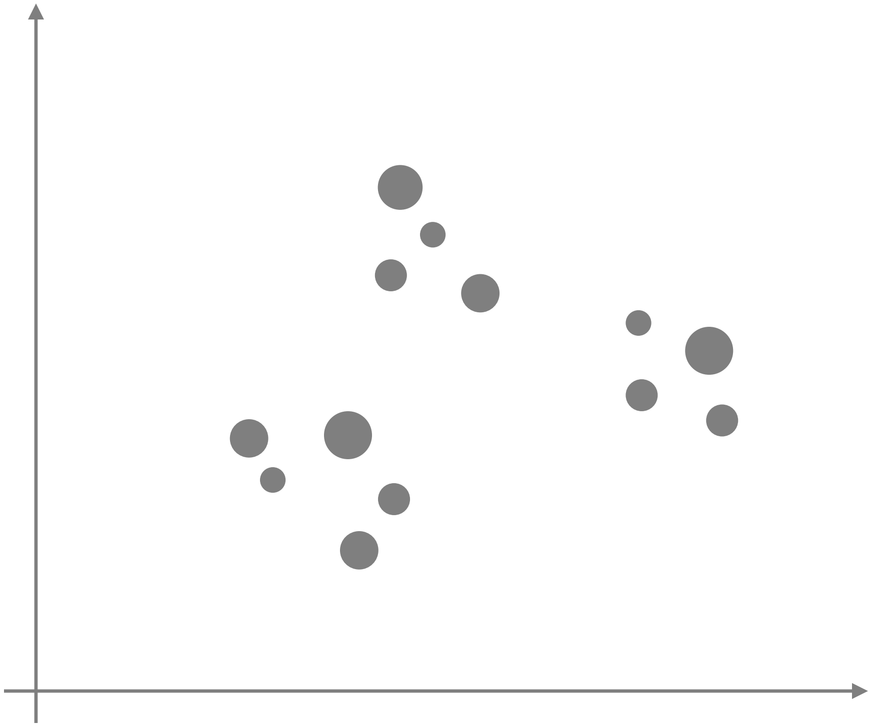
# Visual variable #1 – Planar

| Visual property | Can convey |
|---|---|
| Associative | Y |
| Selective | Y |
| Ordered | Y |
| Quantitative | Y |



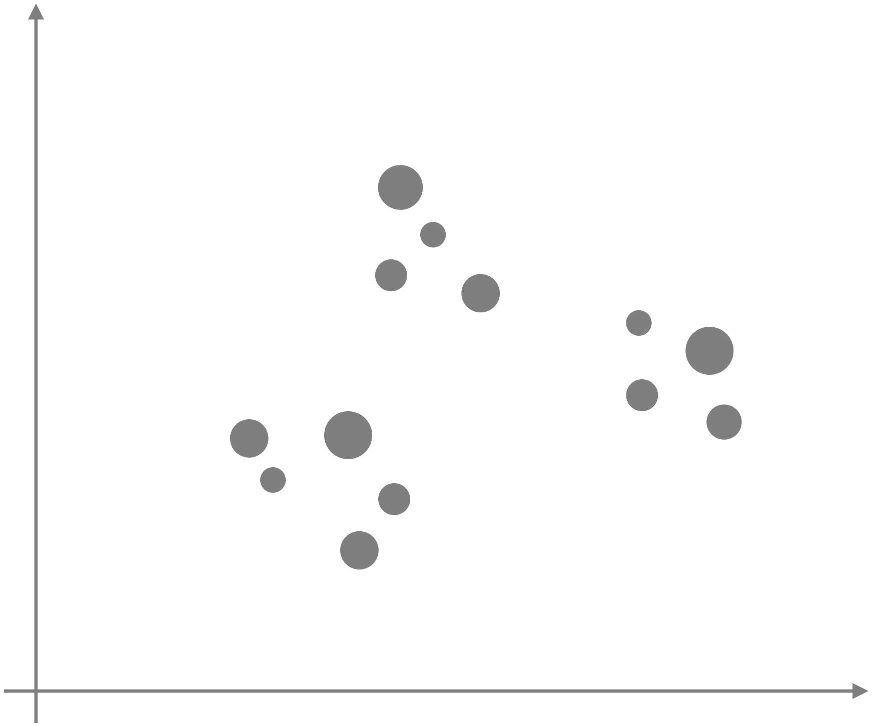
# Visual variable #2 – Size

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



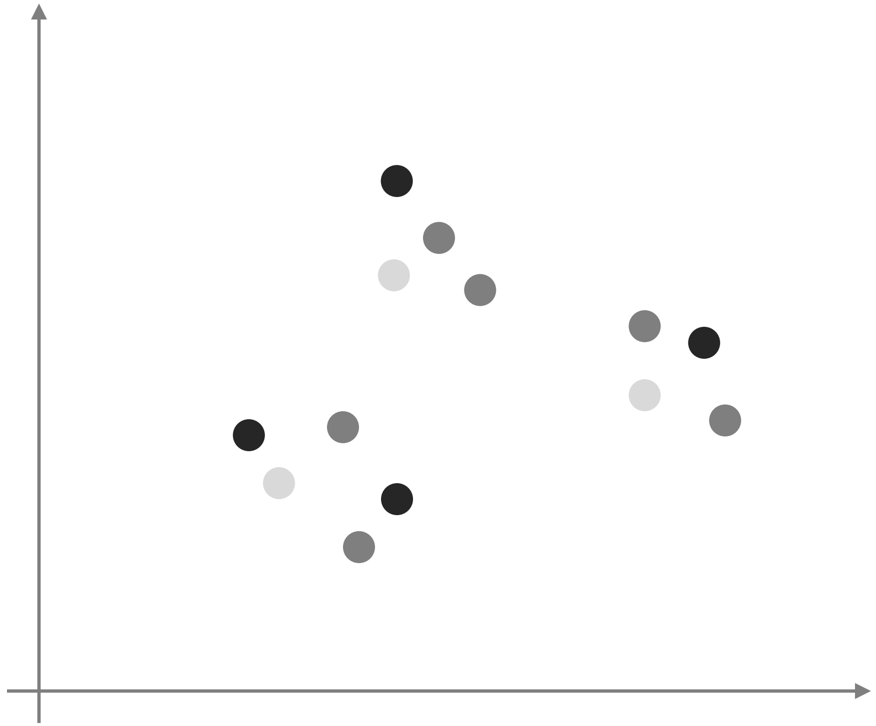
# Visual variable #2 – Size

| Visual property | Can convey |
|---|---|
| Associative | Y |
| Selective | Y |
| Ordered | Y |
| Quantitative | (Y) |



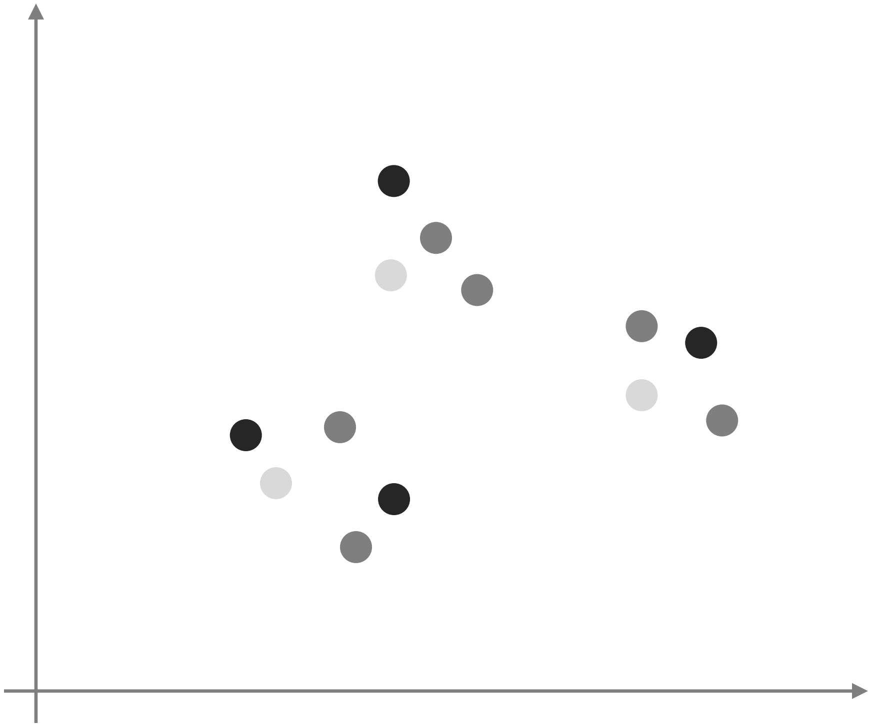
# Visual variable #3 – Brightness

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



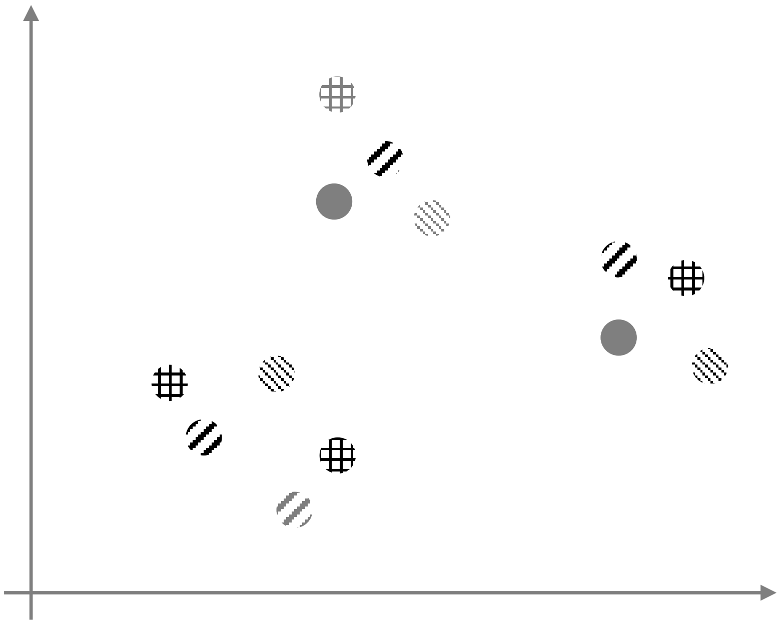
# Visual variable #3 – Brightness

| Visual property | Can convey |
|---|---|
| Associative | Y |
| Selective | Y |
| Ordered | Y |
| Quantitative | |



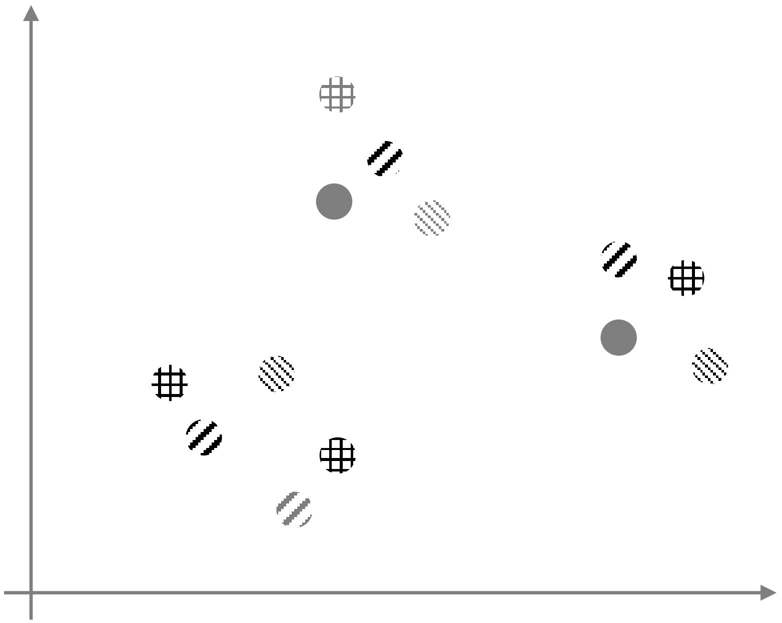
# Visual variable #4 – Texture

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



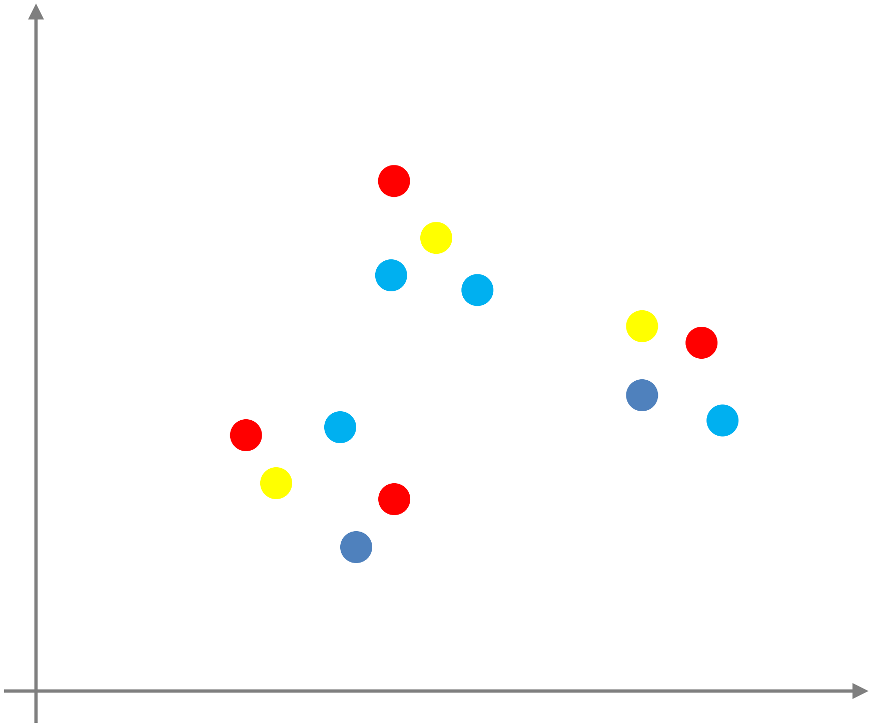
# Visual variable #4 – Texture

| Visual property | Can convey |
|---|---|
| Associative | Y |
| Selective | Y |
| Ordered | |
| Quantitative | |



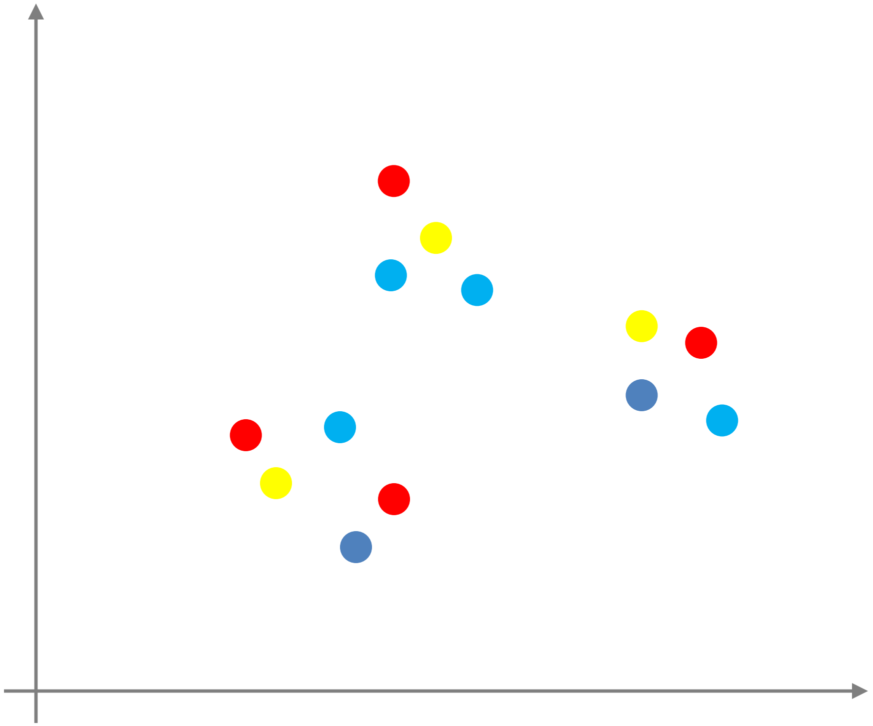
# Visual variable #4 – Color

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



# Visual variable #4 – Color

| Visual property | Can convey |
| --- | --- |
| Associative | Y |
| Selective | Y |
| Ordered | |
| Quantitative | |



# Visual variable #5 – Orientation

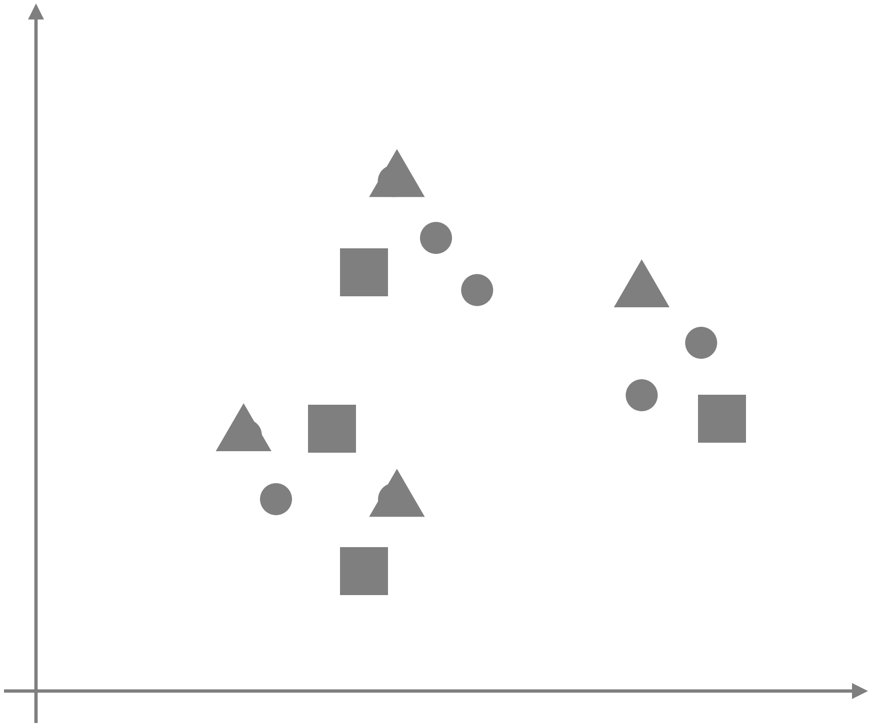
| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |

# Visual variable #5 – Orientation

| Visual property | Can convey |
|---|---|
| Associative | (Y) |
| Selective | (Y) |
| Ordered | |
| Quantitative | |

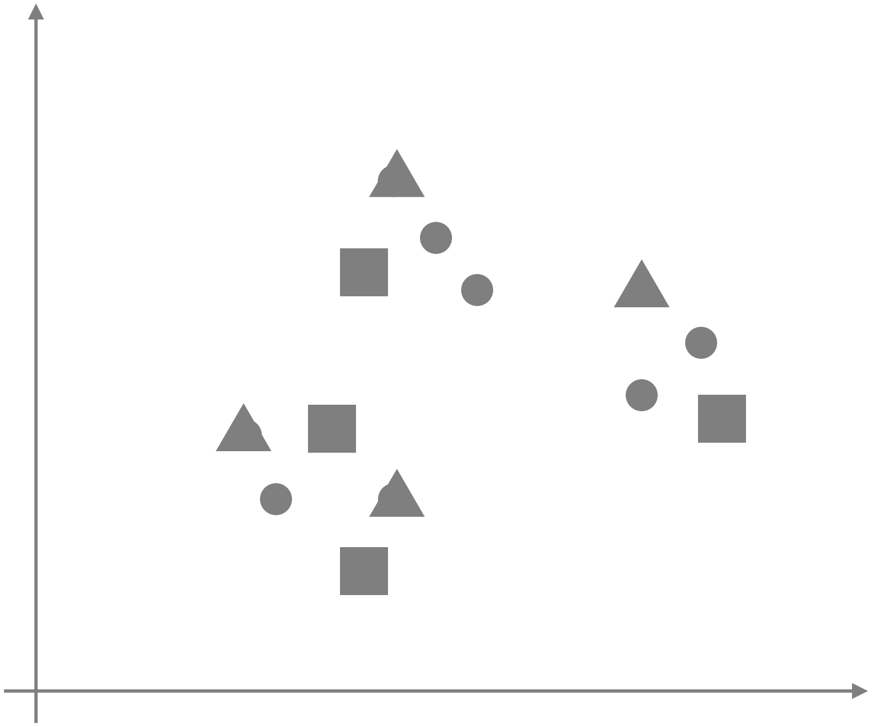
# Visual variable #6 – Shape

| Visual property | Can convey |
|---|---|
| Associative | |
| Selective | |
| Ordered | |
| Quantitative | |



# Visual variable #6 – Shape

| Visual property | Can convey |
| --- | --- |
| Associative | (Y) |
| Selective | (Y) |
| Ordered | |
| Quantitative | |



# Levels of Organization

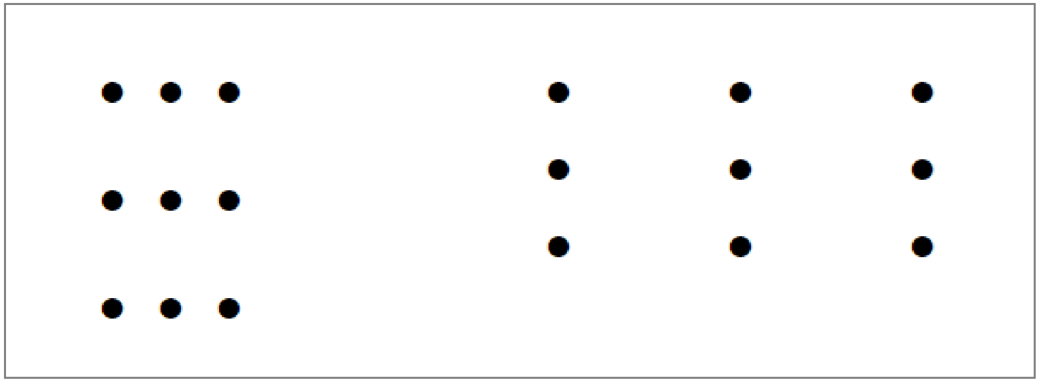
Visual variables differ in what data properties they can convey

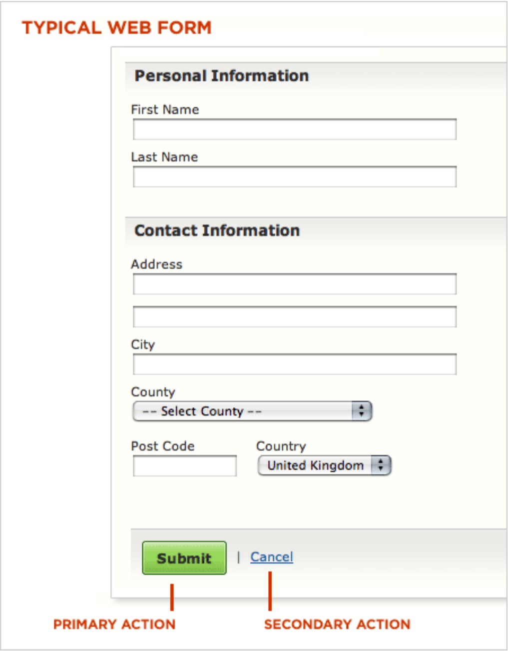
| | Associative | Selective | Ordered | Quantitative |
|---|---|---|---|---|
| Planar | yes | yes | yes | yes |
| Size | yes | yes | yes | (yes) |
| Brightness (Value) | yes | yes | yes | |
| Texture | yes | yes | | |
| Color (Hue) | yes | yes | | |
| Orientation | (yes) | (yes) | | |
| Shape | (yes) | (yes) | | |

# Take-Aways (1)

Planar variable is the single most strongest visual variable

- maps to proximity
- provides an intuitive organization of information
- things close together are perceptually grouped together





# Take-Aways (2)

Size and brightness are good secondary visual variables to encode *relative* magnitude

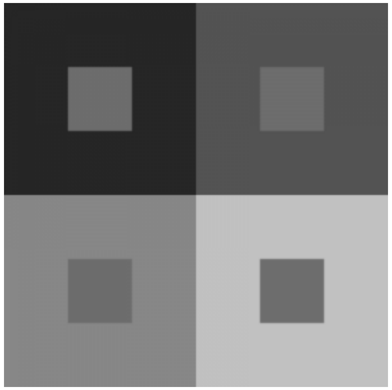
- size appeals to spatial perceptive channels

What are the advantages and disadvantages of brightness

+ brightness does not consume extra space (bigger disks do)

- brightness depends on environmental lighting (size does not)

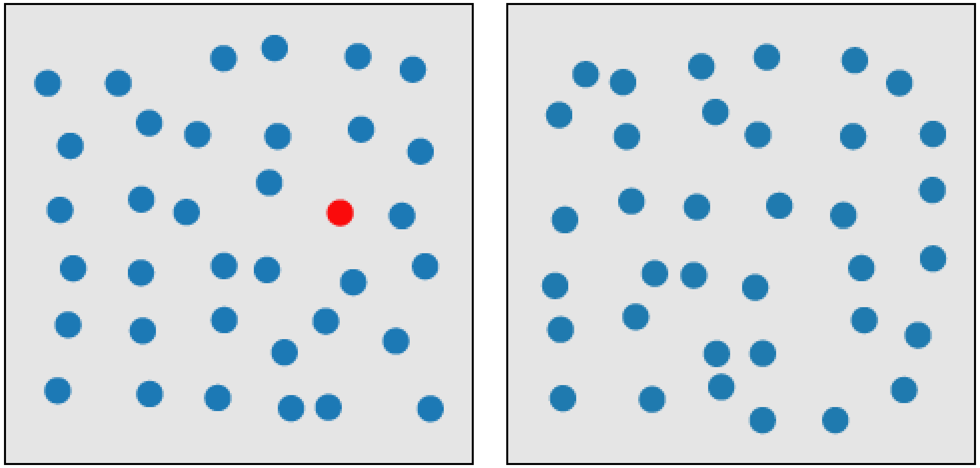
  where do you view the visualization (office, outdoors, night or day?)



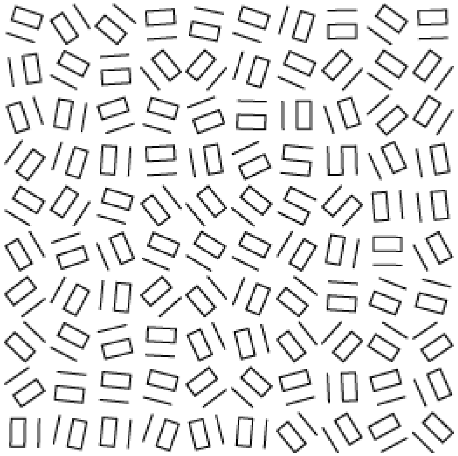
# Take-Aways (3)

Color is a good visual variable for labeling

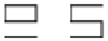
- texture can do this as well, but it does not support pop-out much



color pop-out

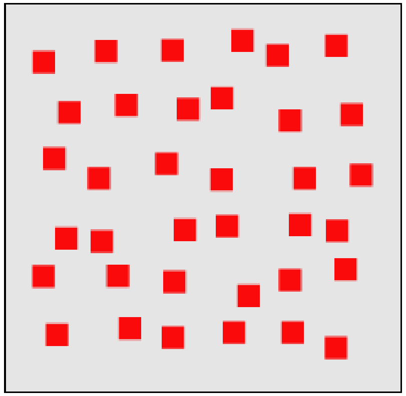


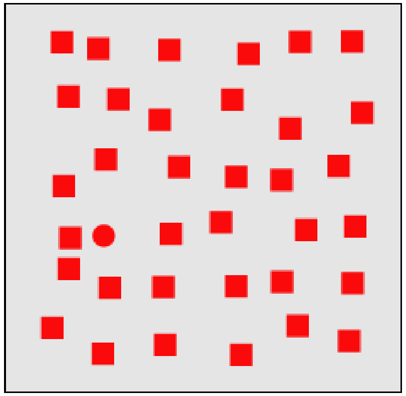
texture pop-out?



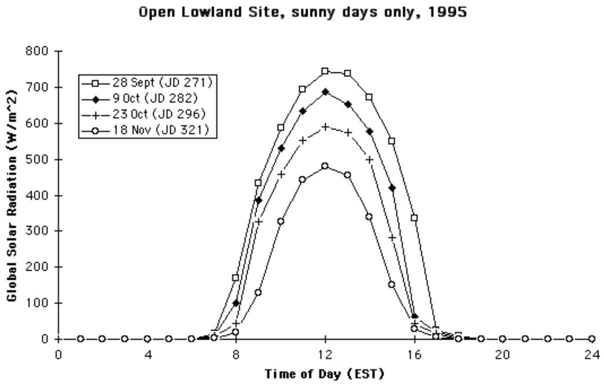
# TAKE AWAYS (4)

*Shape* provides only limited pop-out

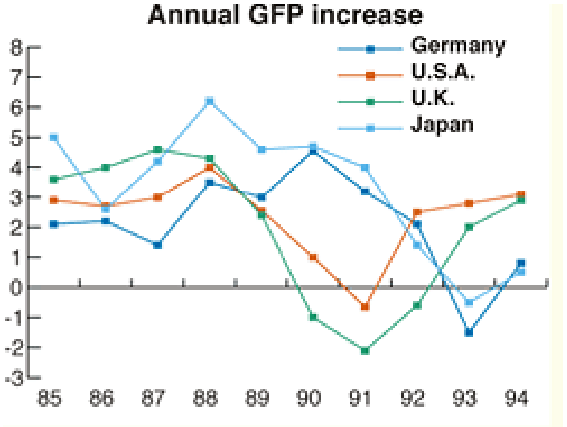




- compare with color pop-out on the previous slide
- another example: coloring of graphs




vs.



# Color and Contrast



Background with same-colored object at the same brightness

- can you see the shape?
- can you count the number of gaps?

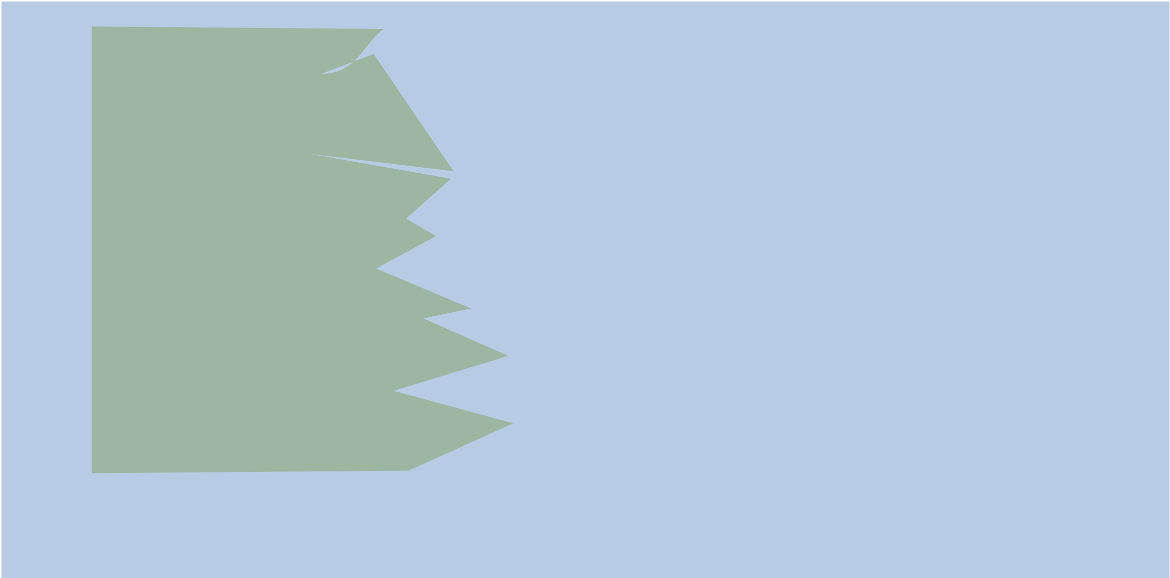
# Color and Contrast



Background with different-colored object at similar brightness

- can you see the shape?
- can you count the number of gaps?

# Color and Contrast



Background with different-colored object at lower brightness

- can you see the shape?
- can you count the number of gaps?

# Color and Contrast



Background with different-colored object at higher brightness

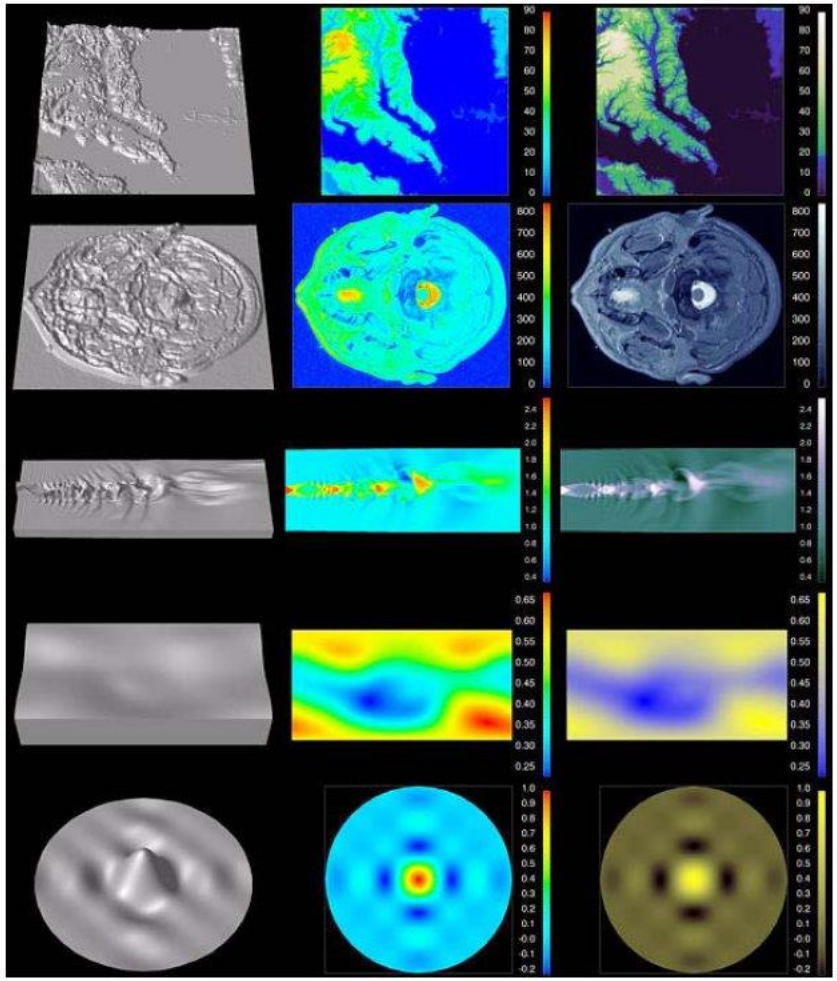
- can you see the shape?
- can you count the number of gaps?

# What Did we Learn from that Experiment?

Color is for …

Brightness (intensity, luminance) is for …

# Luminance and Hue



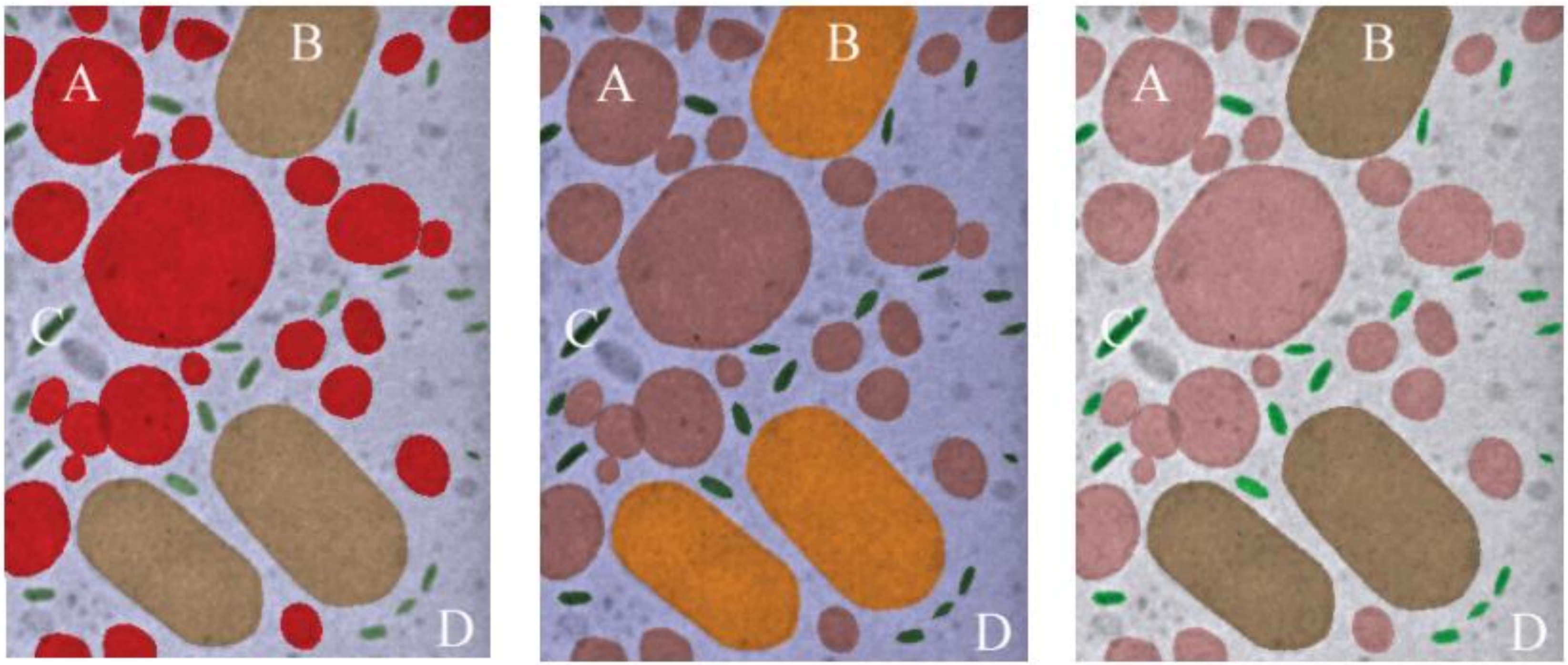
luminance mapped to height    just hue    hue and luminance

encode high frequency information by L

from Bergman/Rogowitz/Treinish Vis'95

# Role of Saturation

# Color Tagging for Importance

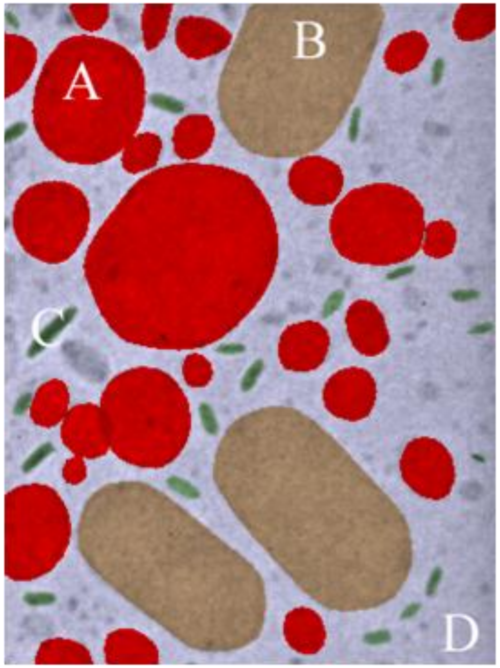



Which is the most important structure in each (as intended by the author)

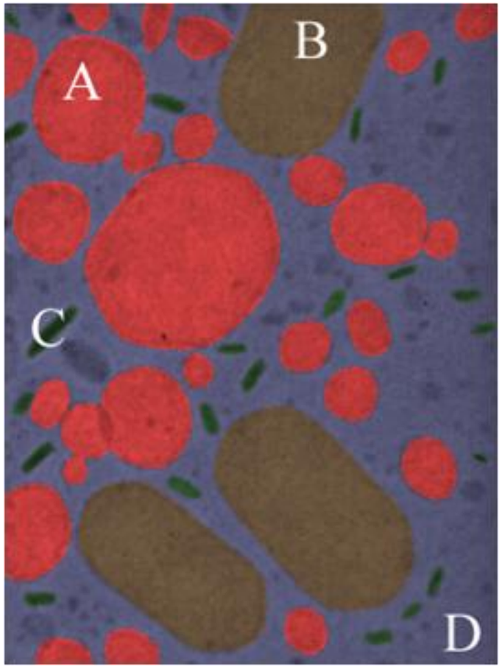
# How About Aesthetics?

Which one do people like better?

- perceived importance level of red object is the same




Vis 1

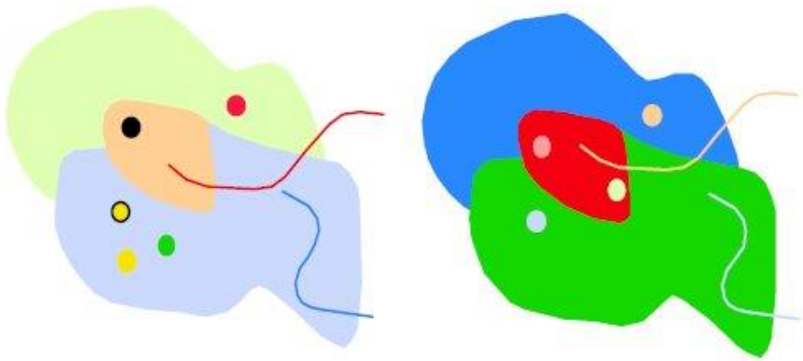



Vis 2

aesthetics

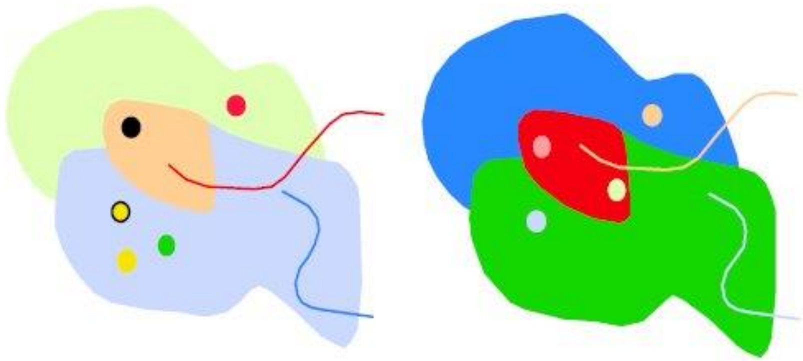
# Color Coding and Colormaps

- Color coding
  - large areas: low saturation
  - small areas: high saturation
  - maintain luminance contrast
  - break iso-luminances with borders

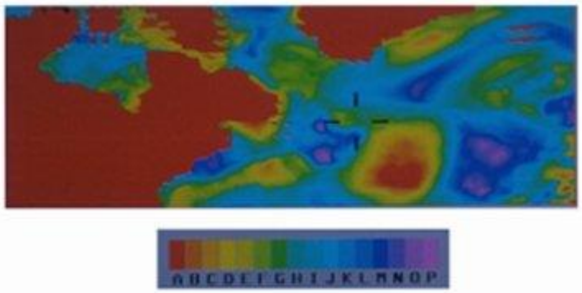


- Pseudo-coloring: assign colors to grey levels by indexing the grey levels into a color map

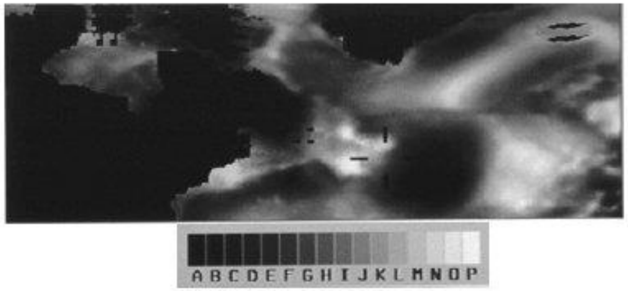
# Color Coding and Colormaps



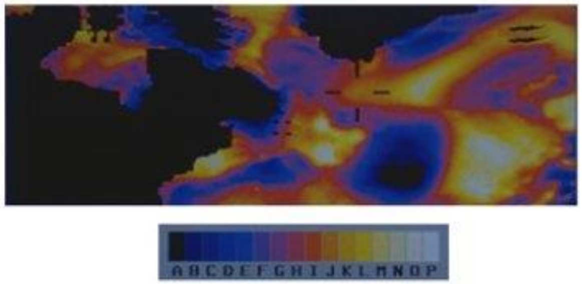
- Color coding
  - large areas: low saturation
  - small areas: high saturation
  - maintain luminance contrast
  - break iso-luminances with borders
- Pseudo-coloring: assign colors to grey levels by indexing the grey levels into a color map




simple spectrum sequence with iso-luminance




original greylevel map




more effective:

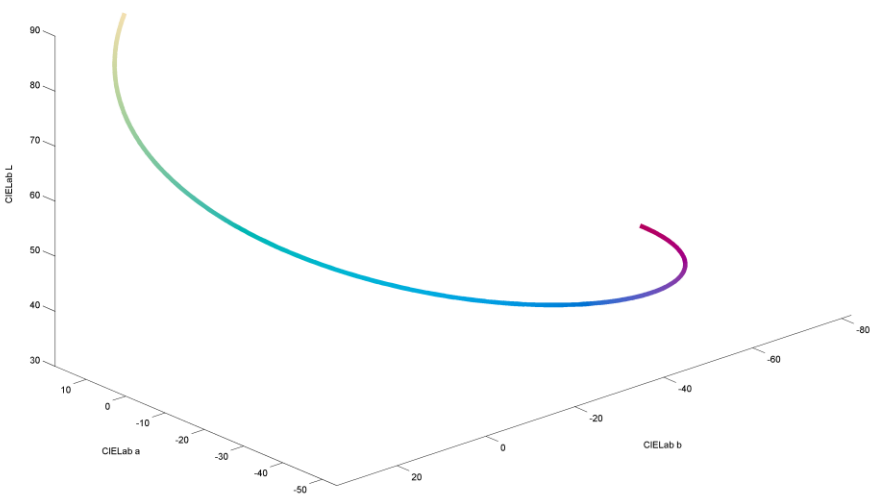
spiral sequence through color space

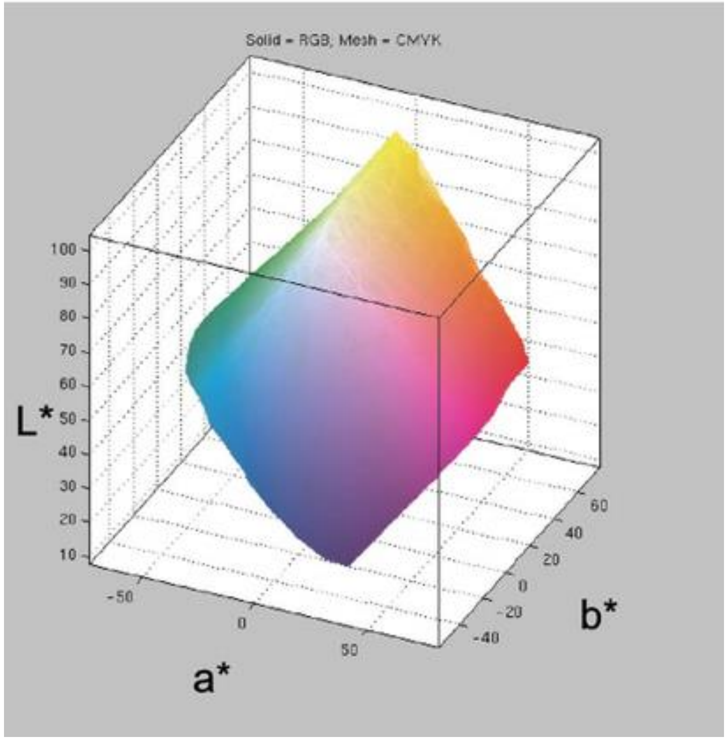
luminance increases with hue

# Spiral Through Color Space

Varies hue and intensity at the same time

- shown here: CIE Lab color space



# The Rainbow Colormap

As we saw, colors can add detail information to a visualization

- instead of 256 levels get $256^3 = 16{,}777{,}216$

Oftentimes you have a visualization with just one variable

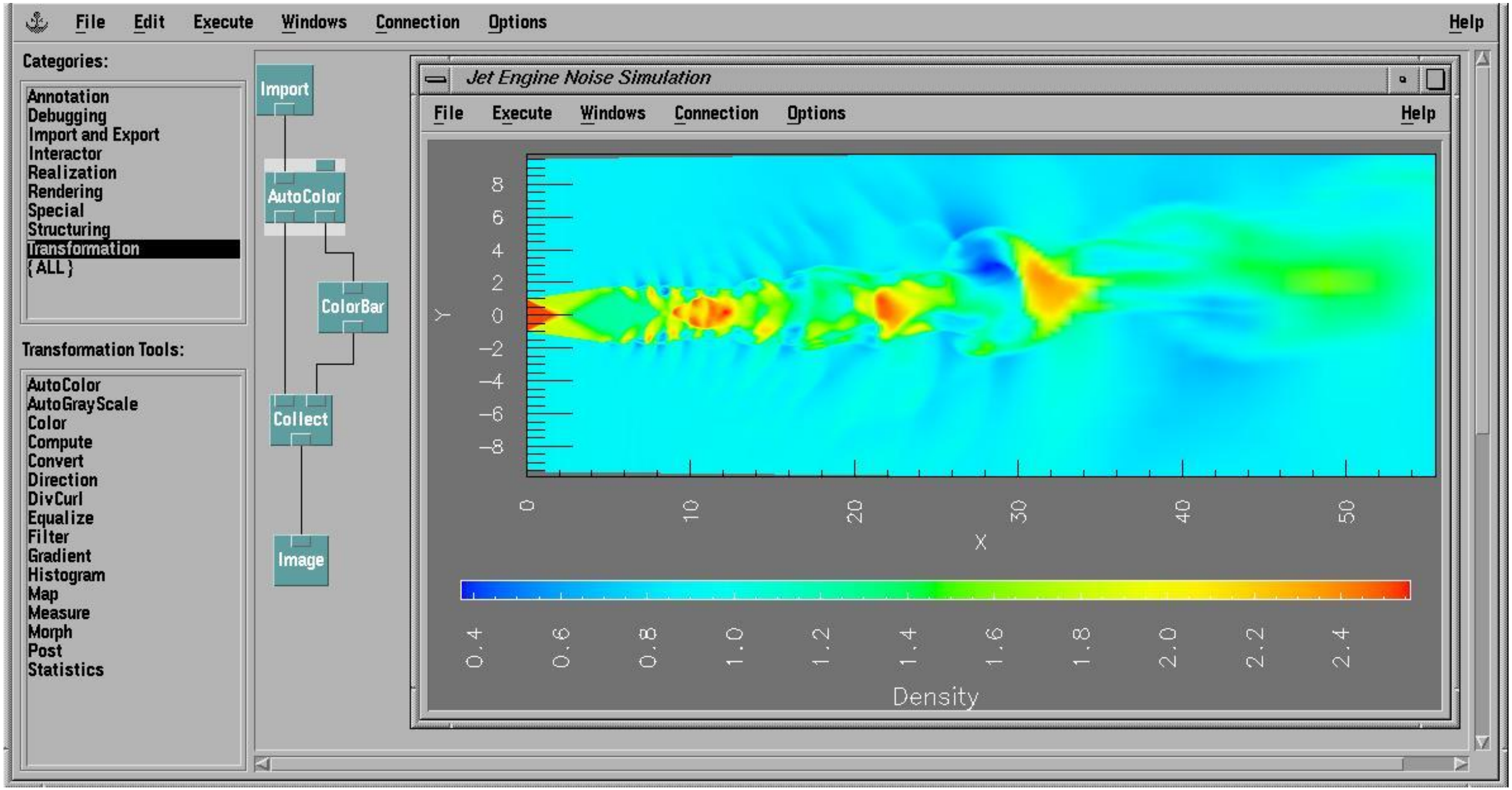
- this would give you a grey level image
- how to turn this into a color image for better detail

Solution 1:

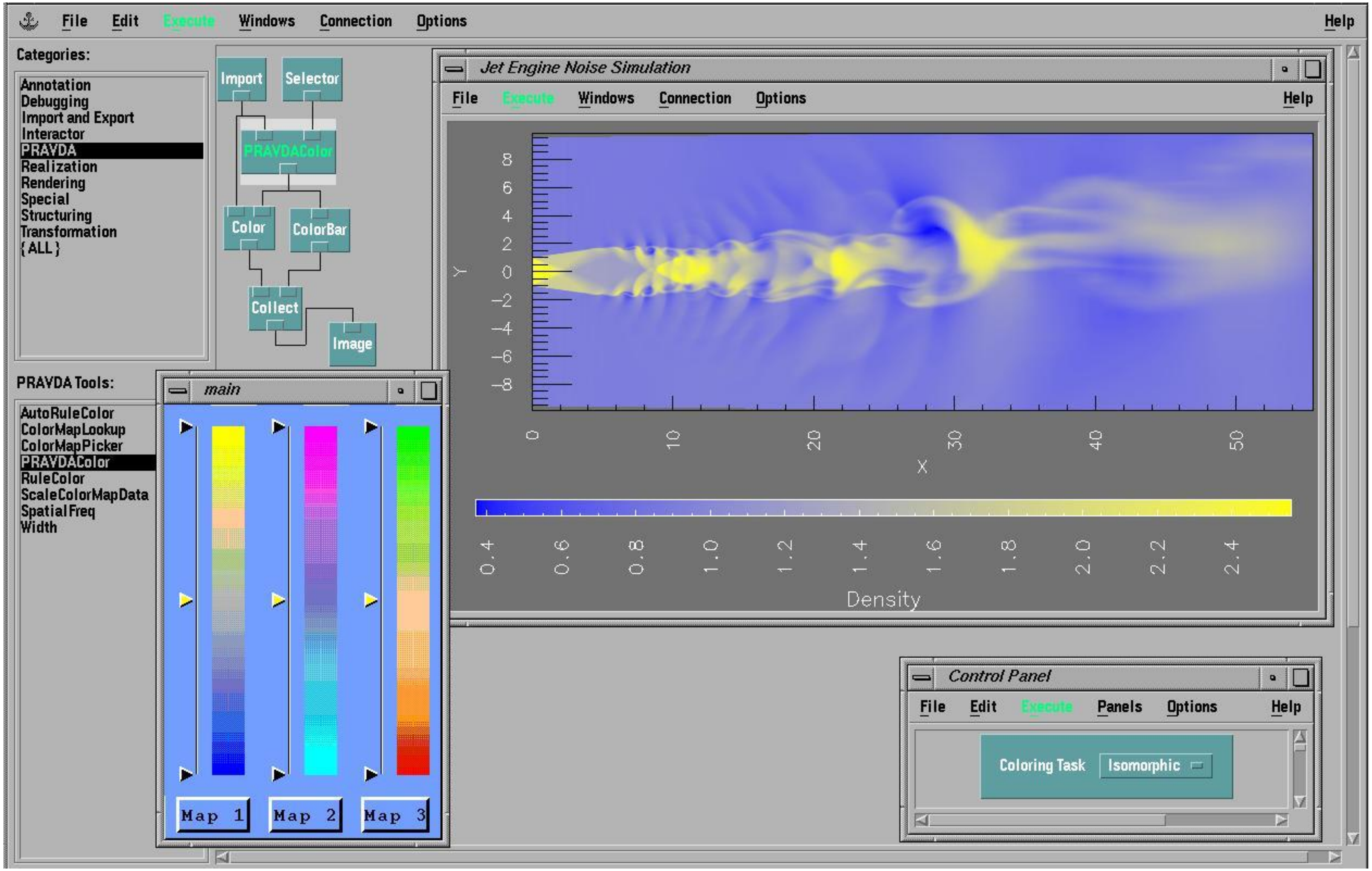
- map to hue → the rainbow colormap

- can you see all adjacent colors at the same contrast?

# Avoid rainbow Colormaps



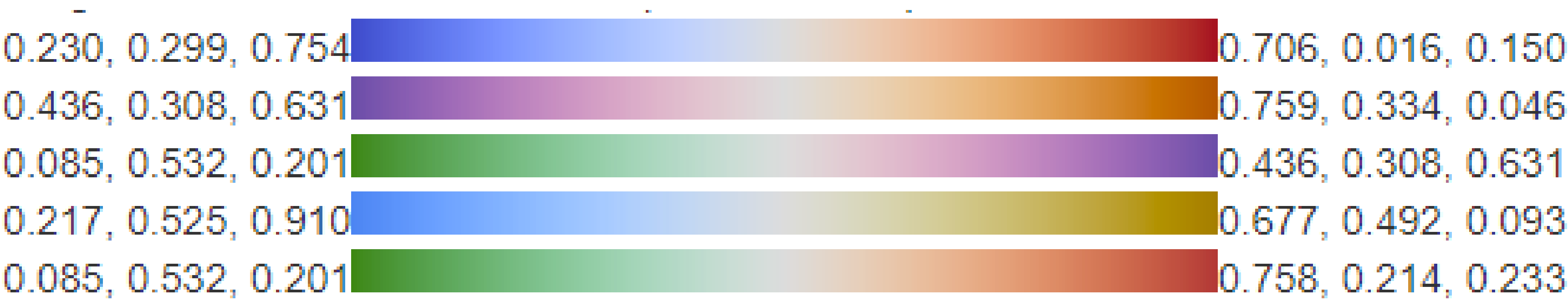
# Better: Linear Hue



# Moreland's Diverging Colormaps

Algorithmically generated

- all have the same midpoint value (0.865, 0.865, 0.865)
- begin and end point listed here




https://www.kennethmoreland.com/color-maps/

# Brewer Scales

## Nominal scales

- distinct hues, but similar emphasis

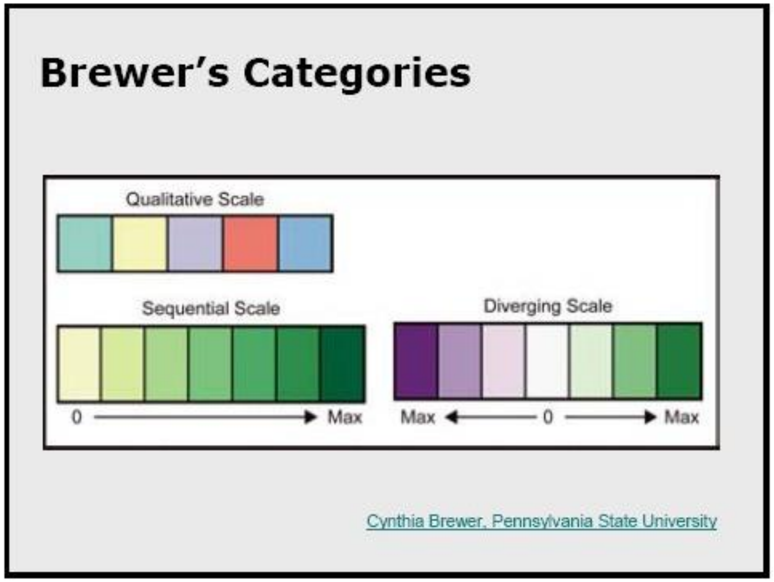
## Sequential scales

- vary in lightness and saturation
- vary slightly in hue

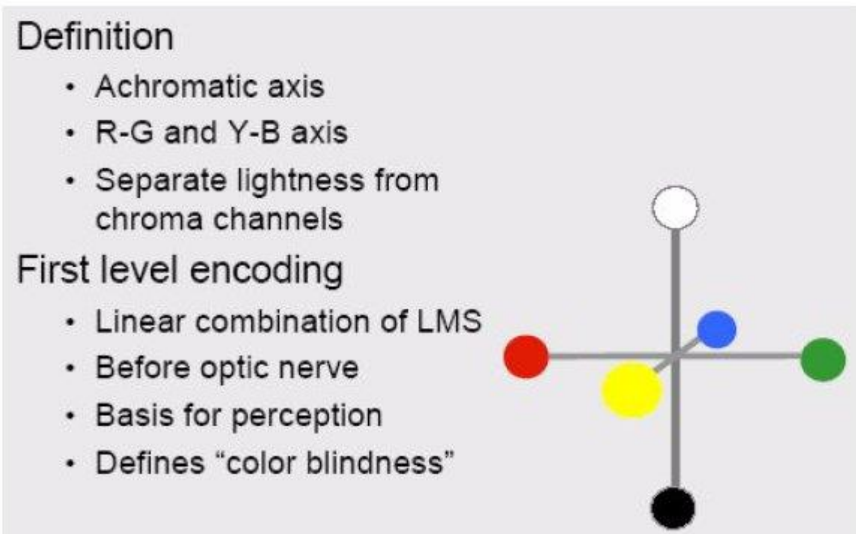
## Diverging scale

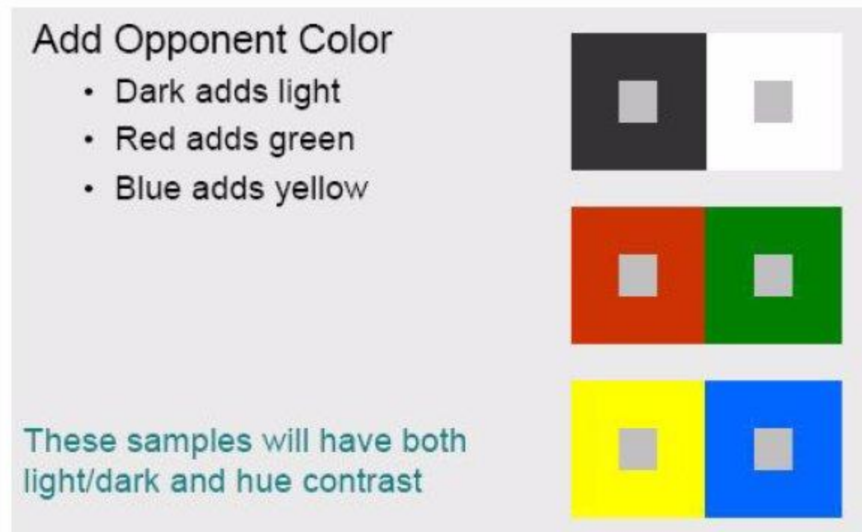
- complementary sequential scales
- neutral at "zero"

http://colorbrewer2.org/



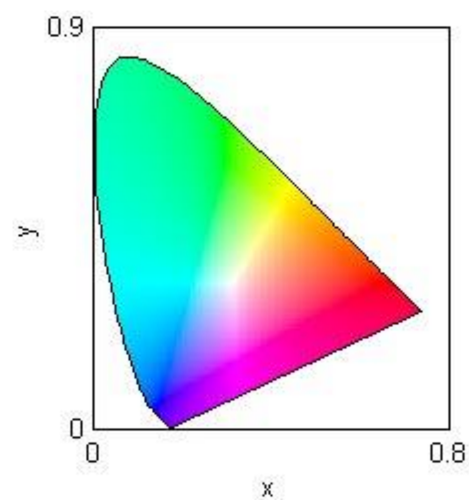
# Opponent Color



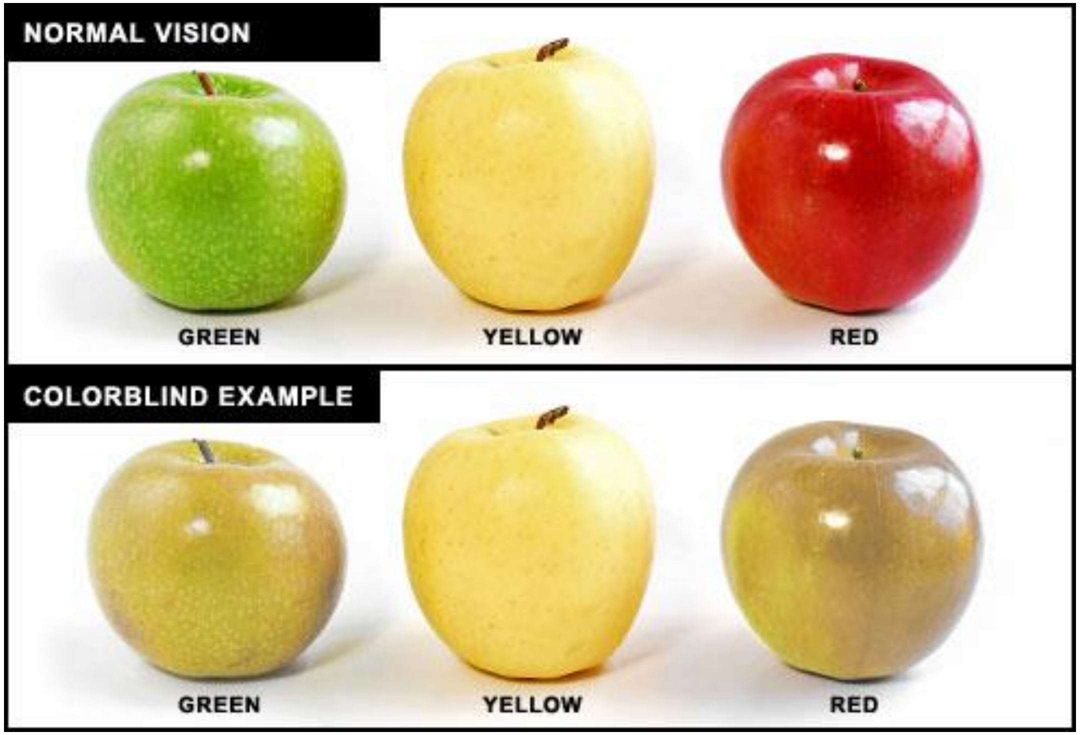




Opponent colors do not mix

- can only see one of the opponents
- there is no blueish yellow
- there is no reddish green



# Color Blindness




Most common is deficiency in distinguishing red and green

# Forms of Color Blindness

normal

The colors of the rainbow as viewed by a person with no color vision deficiencies.

green missing

The colors of the rainbow as viewed by a person with deuteranopia.

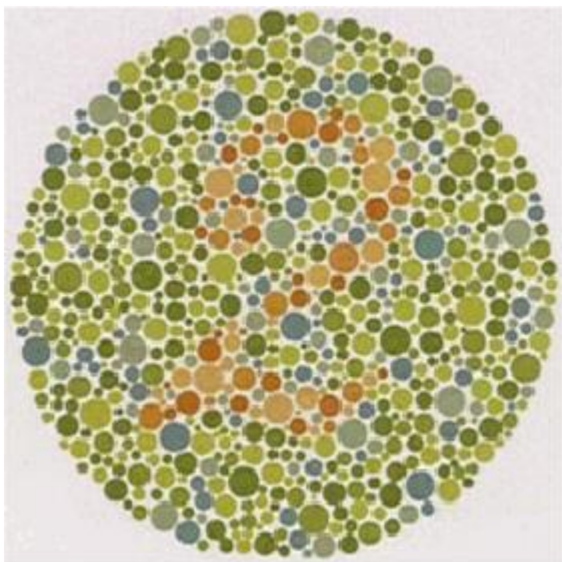
red missing

The colors of the rainbow as viewed by a person with protanopia.

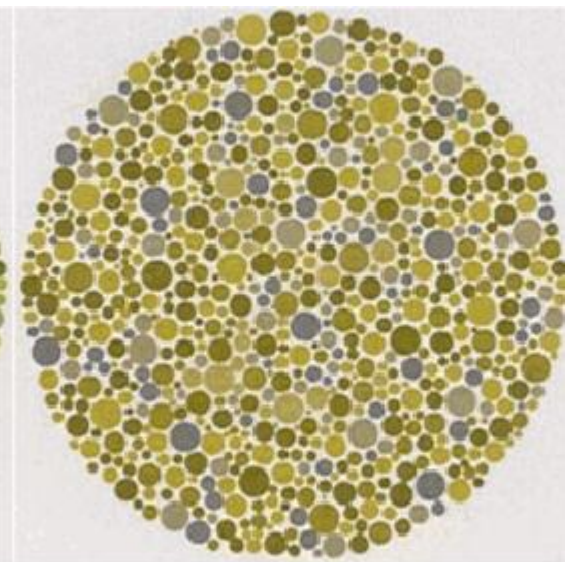
blue missing (rare)

The colors of the rainbow as viewed by a person with tritanopia.

# Ishihara Test

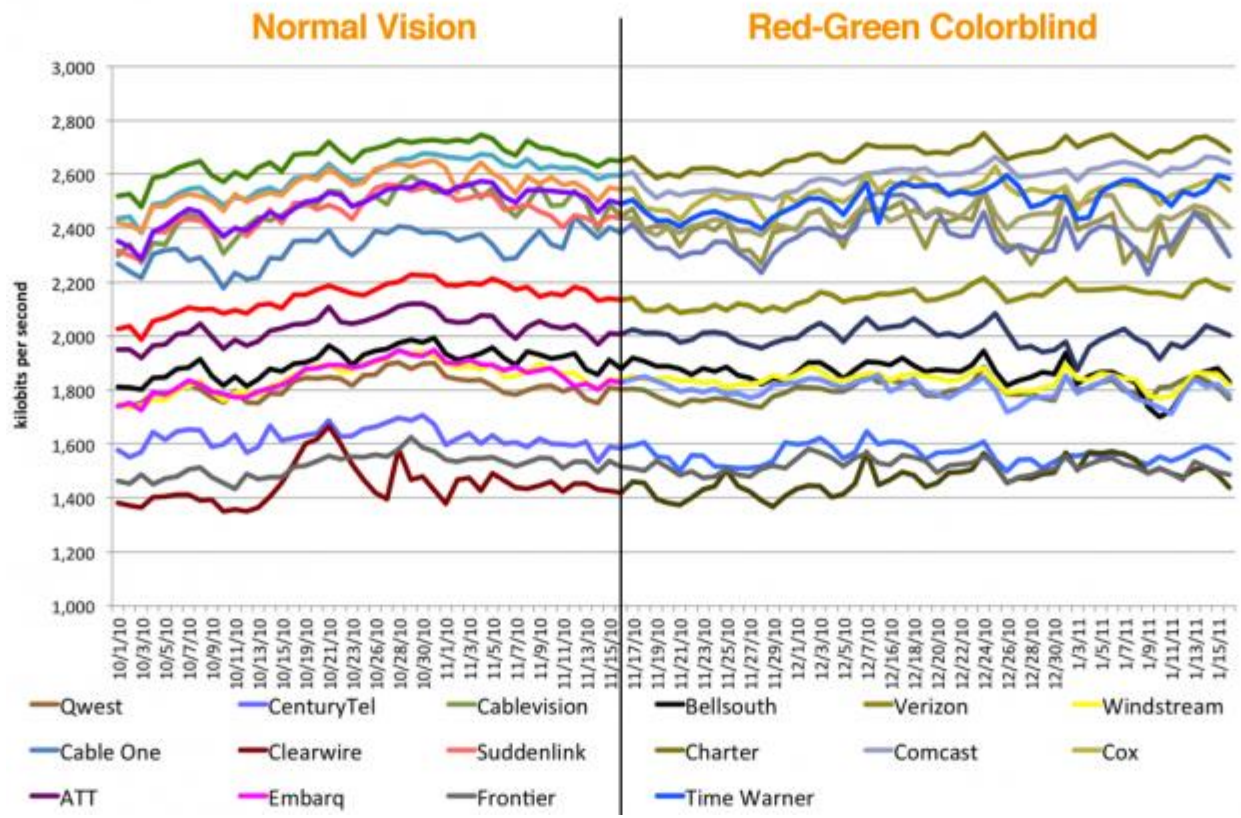


normal



protanopia

# Line Charts



Normal Vision
Red-Green Colorblind
kilobits per second
3,000
2,800
2,600
2,400
2,200
2,000
1,800
1,600
1,400
1,200
1,000
Qwest
CenturyTel
Cablevision
Bellsouth
Verizon
Windstream
Cable One
Clearwire
Suddenlink
Charter
Comcast
Cox
ATT
Embarq
Frontier
Time Warner

# Designing For Color Deficient Users

8% (0.5%) of US males (females) are color deficient

- so be careful when designing visualizations

What to do?

- use different intensities for red-green (e.g. light green, dark red)
- space red and green colored colors dots far apart or make large
- add symbols to line charts or vary line style
- avoid using gradient colors to indicate data value

# SUMMING UP

Use Luminance for detail, shape, and form

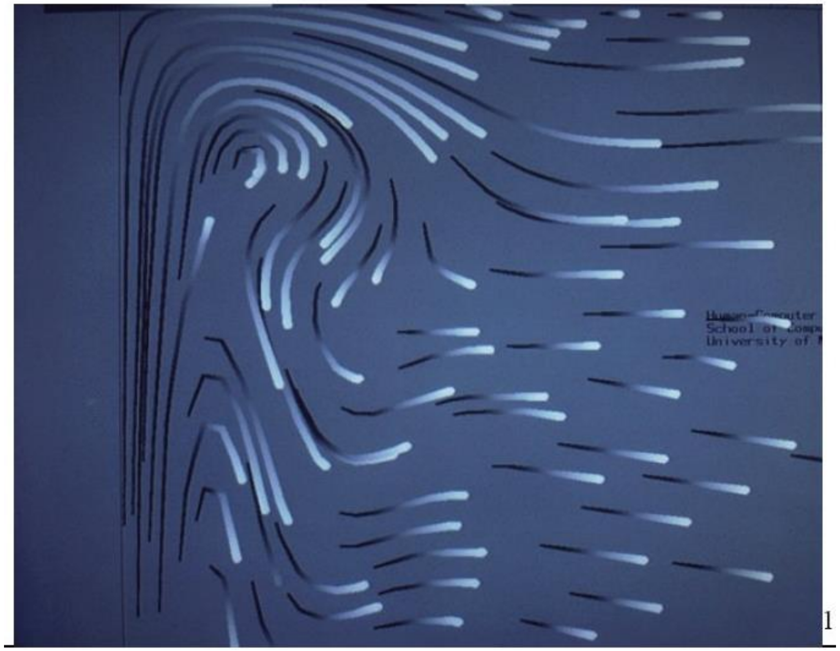
Use color for coding – few colors

Use strong colors for small areas

Use subtle colors to code large areas

Visualization artistry:

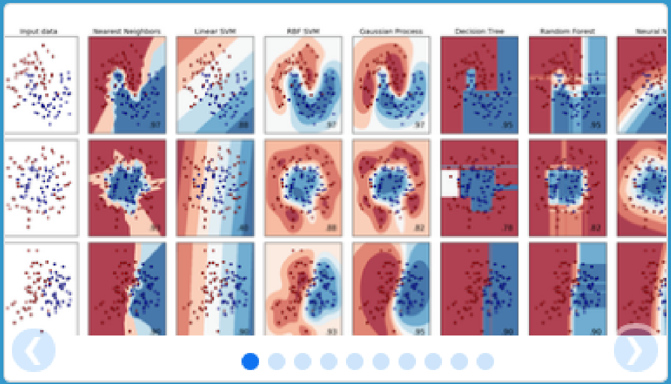
- Use of luminance to indicate direction



scikit learn

Home Installation Documentation Examples

Google™ Custom Search
Search



# scikit-learn

*Machine Learning in Python*

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

## Classification

Identifying to which category an object belongs to.

**Applications**: Spam detection, Image recognition.
**Algorithms**: SVM, nearest neighbors, random forest, ...

— Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications**: Drug response, Stock prices.
**Algorithms**: SVR, ridge regression, Lasso, ...

— Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications**: Customer segmentation, Grouping experiment outcomes
**Algorithms**: k-Means, spectral clustering, mean-shift, ...

— Examples

## Dimensionality reduction

Reducing the number of random variables to consider.

**Applications**: Visualization, Increased efficiency
**Algorithms**: PCA, feature selection, non-negative matrix factorization.

— Examples

## Model selection

Comparing, validating and choosing parameters and models.

**Goal**: Improved accuracy via parameter tuning
**Modules**: grid search, cross validation, metrics.

— Examples

## Preprocessing

Feature extraction and normalization.

**Application**: Transforming input data such as text for use with machine learning algorithms.
**Modules**: preprocessing, feature extraction.

— Examples