

Intro to Data Visualization

Dr. Michele C. Weigle
REU Site 2022
Old Dominion University

Many slides courtesy Tamara Munzner, VAD minicourse, June 2014, <http://www.cs.ubc.ca/~tmm/talks.html#minicourse14>
Based on Tamara Munzner, [Visualization Analysis and Design](#), AK Peters / CRC Press, Oct 2014

Slides adapted from [Module-03 InfoVis](#) from ODU [CS 432/532 Web Science](#)



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](#)

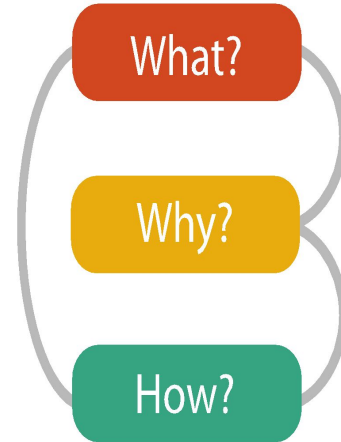
Roadmap

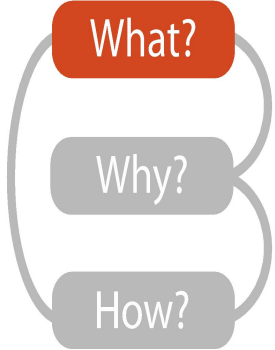
- Data and Basic Principles
- Intro to Creating Charts in Python
- Chart Types (Idioms) w/example code and exercises

VAD book available via ODU login, <https://learning.oreilly.com/library/view/visualization-analysis-and/9781466508910/>

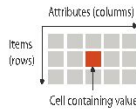
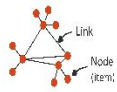
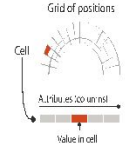
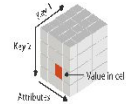










Analysis: What, why, and how

- **what** is shown?
 - **data** abstraction
 - **why** is the user looking at it?
 - **task** abstraction
 - **how** is it shown?
 - **idiom**: visual encoding and interaction
-
- what-why-how analysis framework as scaffold to think systematically about design space



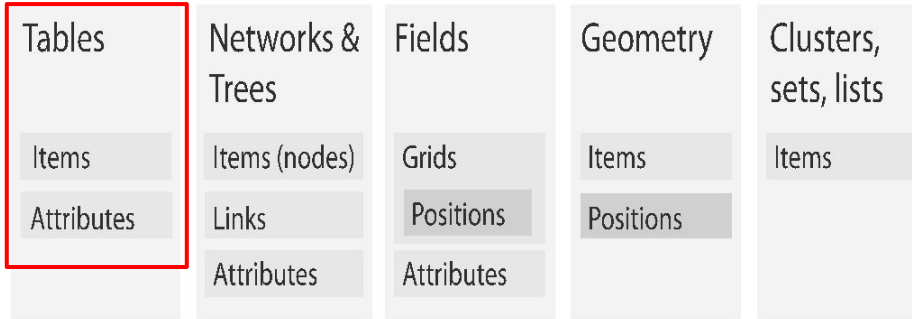


What?

Datasets	Attributes																				
<p>➔ Data Types</p> <p>➔ Items ➔ Attributes ➔ Links ➔ Positions ➔ Grids</p> <p>➔ Data and Dataset Types</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>Tables</td> <td>Networks & Trees</td> <td>Fields</td> <td>Geometry</td> <td>Clusters, sets, lists</td> </tr> <tr> <td>Items</td> <td>Items (nodes)</td> <td>Grids</td> <td>Items</td> <td>Items</td> </tr> <tr> <td>Attributes</td> <td>Links</td> <td>Positions</td> <td>Positions</td> <td></td> </tr> <tr> <td></td> <td>Attributes</td> <td>Attributes</td> <td></td> <td></td> </tr> </table> <p>➔ Dataset Types</p> <p>➔ Tables</p>  <p>➔ Networks</p>  <p>➔ Fields (Continuous)</p>  <p>➔ Multidimensional Table</p>  <p>➔ Trees</p>  <p>➔ Geometry (Spatial)</p>  <p>➔ Dataset Availability</p> <p>➔ Static</p>  <p>➔ Dynamic</p> 	Tables	Networks & Trees	Fields	Geometry	Clusters, sets, lists	Items	Items (nodes)	Grids	Items	Items	Attributes	Links	Positions	Positions			Attributes	Attributes			<p>➔ Attribute Types</p> <p>➔ Categorical</p>  <p>➔ Ordered</p> <p>➔ Ordinal</p>  <p>➔ Quantitative</p>  <p>➔ Ordering Direction</p> <p>➔ Sequential</p>  <p>➔ Diverging</p>  <p>➔ Cyclic</p> 
Tables	Networks & Trees	Fields	Geometry	Clusters, sets, lists																	
Items	Items (nodes)	Grids	Items	Items																	
Attributes	Links	Positions	Positions																		
	Attributes	Attributes																			

Dataset and data types

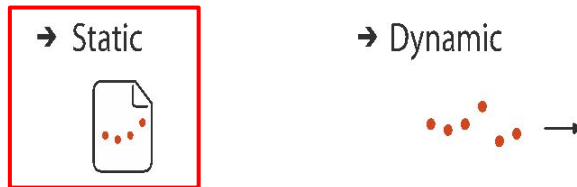
→ Data and Dataset Types



→ Data Types



→ Dataset Availability



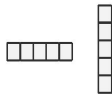
Keys and values

- key
 - independent attribute
 - used as unique index to look up items
 - simple tables: 1 key
 - multidimensional tables: multiple keys
- value
 - dependent attribute, value of cell
- classify arrangements by key count
 - 0, 1, 2, many...

Express Values



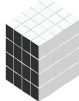
→ 1 Key
List



→ 2 Keys
Matrix



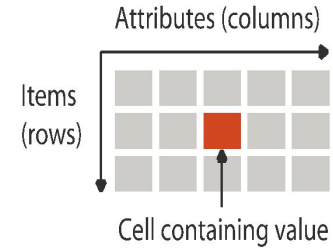
→ 3 Keys
Volume



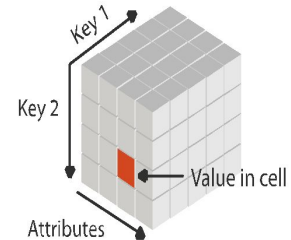
→ Many Keys
Recursive Subdivision



→ Tables



→ Multidimensional Table



Attribute types

➔ Attribute Types

➔ Categorical



➔ Ordered

➔ *Ordinal*



➔ *Quantitative*



Identify items, attributes, keys, types

date	precipitation	temp_max	temp_min	wind	weather
2012-01-01	0.0	12.8	5.0	4.7	drizzle
2012-01-02	10.9	10.6	2.8	4.5	rain
2012-01-03	0.8	11.7	7.2	2.3	rain
2012-01-04	20.3	12.2	5.6	4.7	rain
2012-01-05	1.3	8.9	2.8	6.1	rain
2012-01-06	2.5	4.4	2.2	2.2	rain
2012-01-07	0.0	7.2	2.8	2.3	rain

<https://github.com/vega/vega-datasets/blob/next/data/seattle-weather.csv>

"Graphics reveal data" -Edward Tufte

- Vis tools can allow people to explore data to find patterns or to determine if a statistical model actually fits the data
- ***But***, look out for questionable data
 - "just because it's numbers doesn't mean it's true"
 - is it a typo or something interesting?
"make sure you know which one it is"

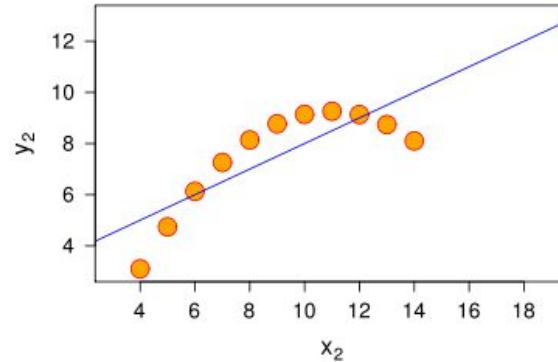
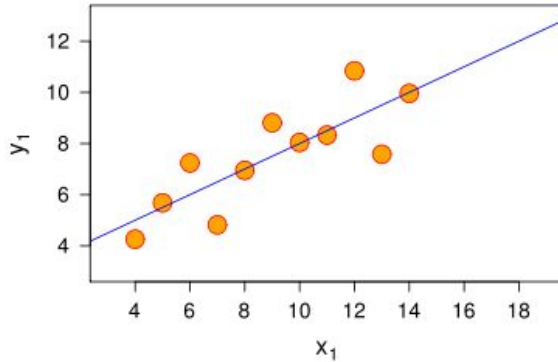
Anscombe's Quartet

	I		II		III		IV	
	x	y	x	y	x	y	x	y
	10	8,04	10	9,14	10	7,46	8	6,58
	8	6,95	8	8,14	8	6,77	8	5,76
	13	7,58	13	8,74	13	12,74	8	7,71
	9	8,81	9	8,77	9	7,11	8	8,84
	11	8,33	11	9,26	11	7,81	8	8,47
	14	9,96	14	8,1	14	8,84	8	7,04
	6	7,24	6	6,13	6	6,08	8	5,25
	4	4,26	4	3,1	4	5,39	19	12,5
	12	10,84	12	9,13	12	8,15	8	5,56
	7	4,82	7	7,26	7	6,42	8	7,91
	5	5,68	5	4,74	5	5,73	8	6,89
SUM	99,00	82,51	99,00	82,51	99,00	82,50	99,00	82,51
AVG	9,00	7,50	9,00	7,50	9,00	7,50	9,00	7,50
STDEV	3,32	2,03	3,32	2,03	3,32	2,03	3,32	2,03

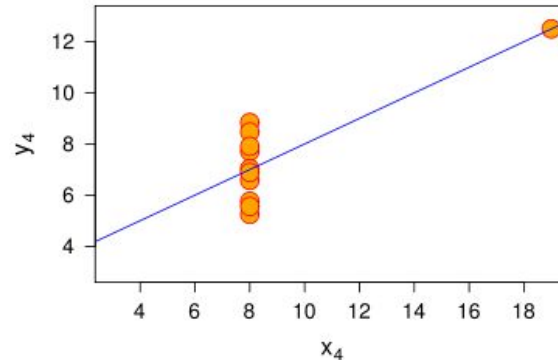
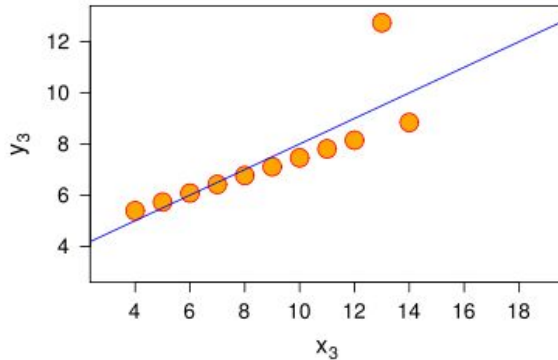
F.J. Anscombe, "Graphs in Statistical Analysis", *American Statistician*, Feb 1973

Image from <https://medium.com/datadriveninvestor/anscombes-quartet-12649db7eac0>, Munzner, Figure 1.3

The four data sets are *not* the same



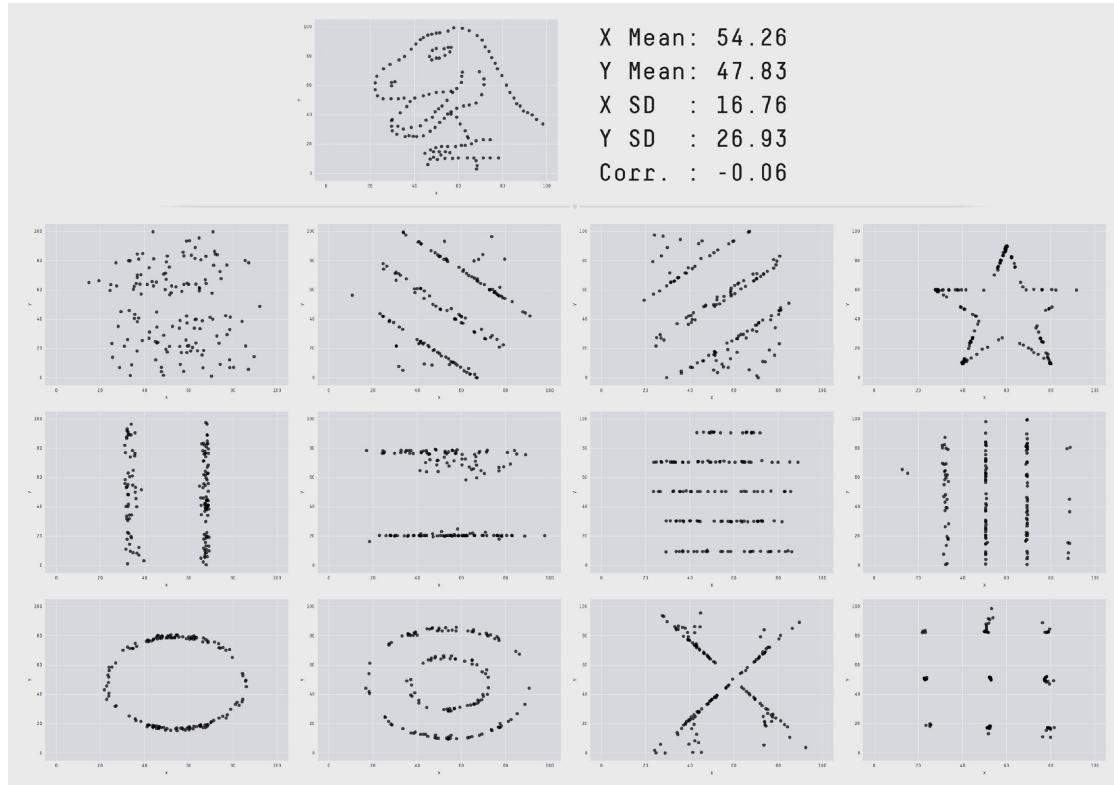
"Graphics *reveal* data"



<http://en.wikipedia.org/wiki/File:Anscombe.svg>

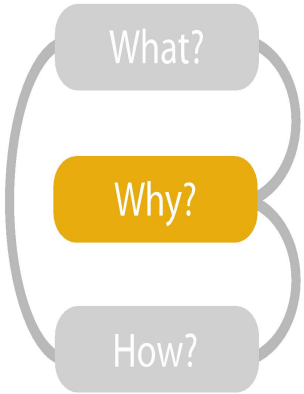
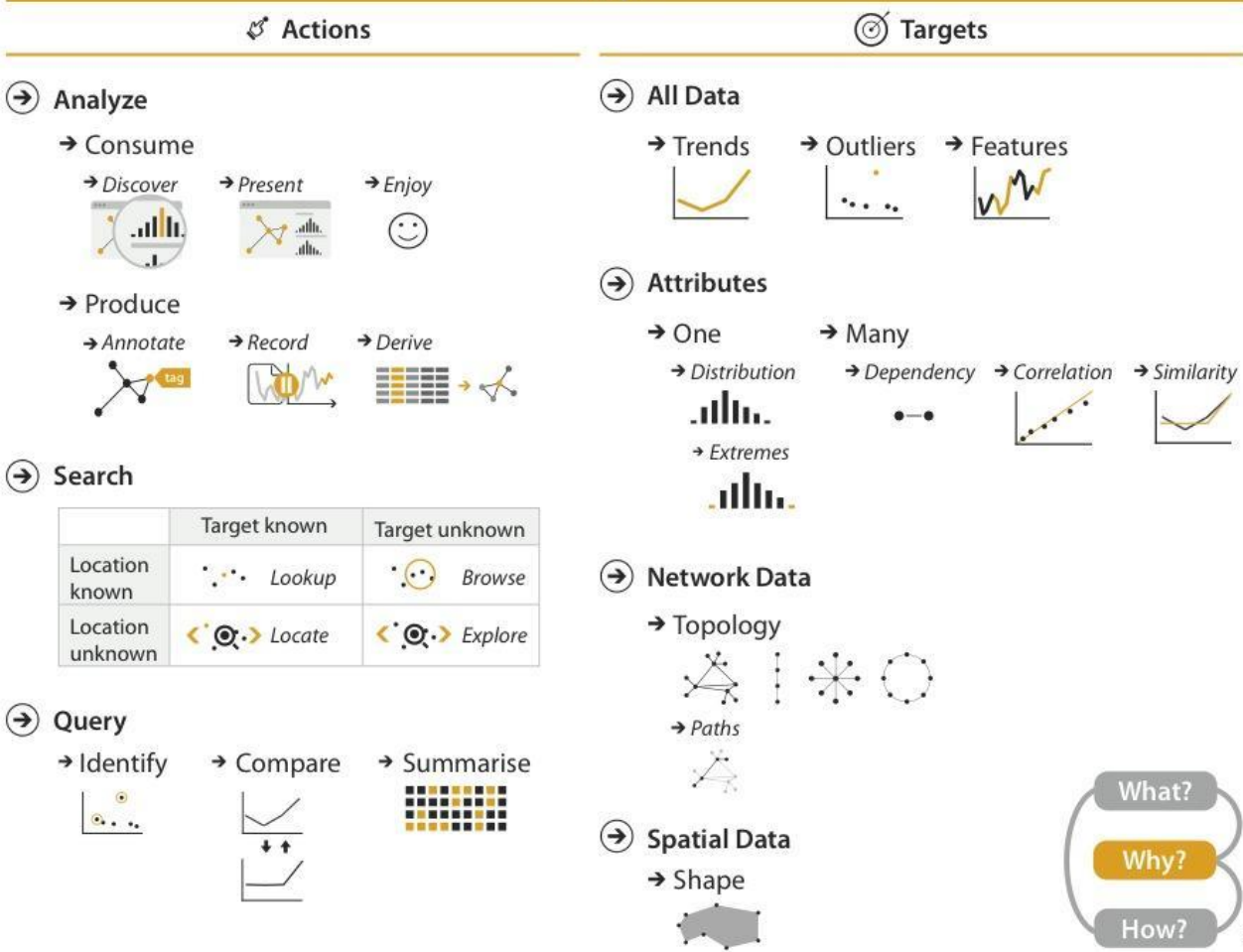
Datasaurus Dozen

<http://robertgrantstats.co.uk/drawmydata.html>



Justin Matejka, George Fitzmaurice, "Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing", ACM SIGCHI 2017, <https://www.autodeskresearch.com/publications/samestats>

Why?



- {action, target} pairs
 - discover distribution
 - compare trends
 - locate outliers
 - browse topology



High-level actions: Analyze

- consume

- discover vs present

- aka explore vs explain

- enjoy

- aka casual, social

- produce

- annotate, record

- derive

- create new *derived attributes* based on existing attributes

➔ Analyze

➔ Consume

➔ Discover



➔ Present



➔ Enjoy



➔ Produce

➔ Annotate



➔ Record




➔ Derive



Actions: Mid-level search, low-level query

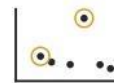
- what does user know?
 - target, location
- how much of the data matters?
 - one, some, all

→ Search

	Target known	Target unknown
Location known	 <i>Lookup</i>	 <i>Browse</i>
Location unknown	 <i>Locate</i>	 <i>Explore</i>

→ Query

→ Identify



→ Compare



→ Summarise



Why: Targets

→ ALL DATA

→ Trends



→ Outliers



→ Features



→ ATTRIBUTES

→ One

→ *Distribution*



↓ *Extremes*



→ Many

→ *Dependency*



→ *Correlation*

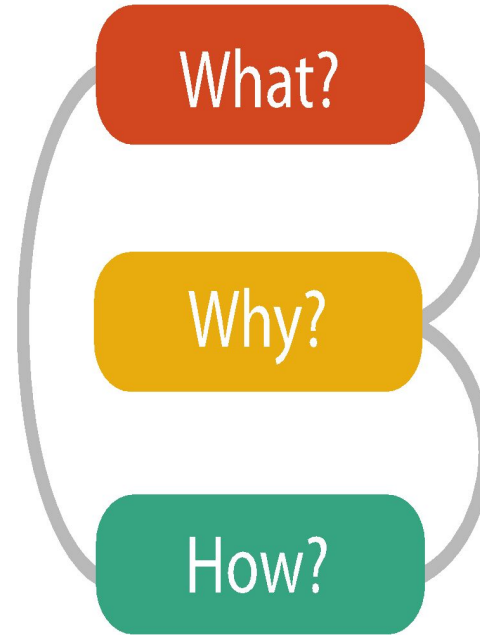


→ *Similarity*



Workflow

- What
 - data gathering
 - data wrangling
- Why
 - developing questions
 - initial analysis
- How
 - charts for analysis
 - charts for presentation



How?

Encode

→ Arrange

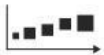
→ Express



→ Separate



→ Order



→ Align



→ Use



What?

Why?

How?

→ Map

from **categorical** and **ordered** attributes

→ Color



→ Size, Angle, Curvature, ...



→ Shape



→ Motion

Direction, Rate, Frequency, ...



Manipulate

→ Change



→ Select

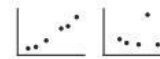


→ Navigate

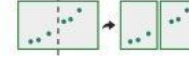


Facet

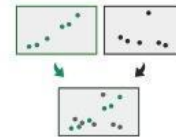
→ Juxtapose



→ Partition



→ Superimpose

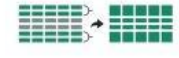


Reduce

→ Filter



→ Aggregate

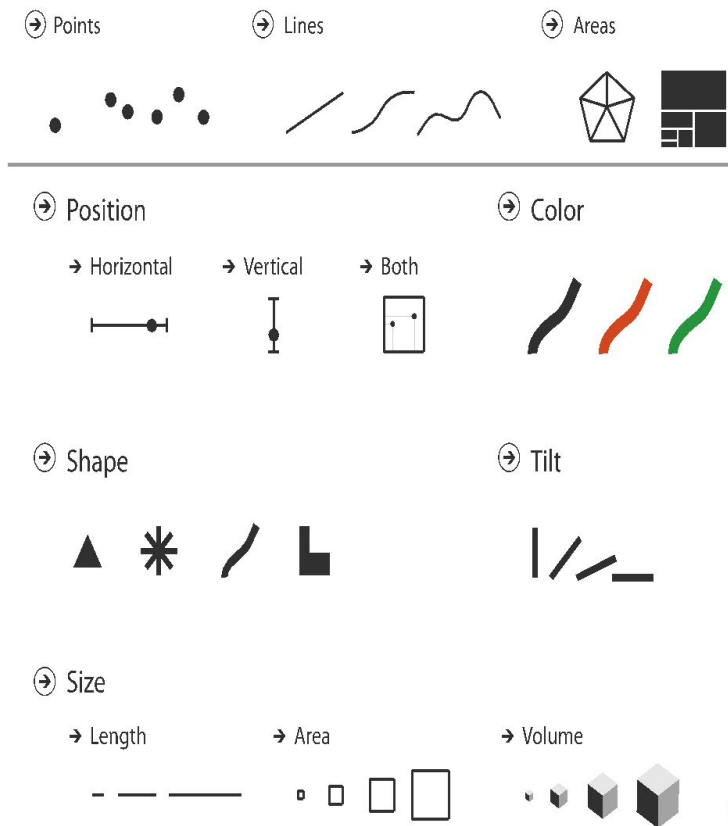


→ Embed



Definitions: Marks and channels

- marks
 - geometric primitives
- channels
 - control appearance of marks
 - can redundantly code with multiple channels
- interactions
 - point marks only convey position; no area constraints
 - can be size and shape coded
 - line marks convey position and length
 - can only be size coded in 1D (width)
 - area marks fully constrained
 - cannot be size or shape coded



Color: Luminance, saturation, hue

- 3 channels

- identity channel for categorical

- hue

- magnitude channels for ordered

- luminance
- saturation

Hue



Luminance



Saturation



Color Spaces

- Six different hues ordered by luminance
 - computed lightness L values same (HSL color space)
 - true luminance is closer to what we perceive
 - L^* is computed perceptually linear luminance - best match to what we see ($L^*a^*b^*$ color space, aka CIELAB)

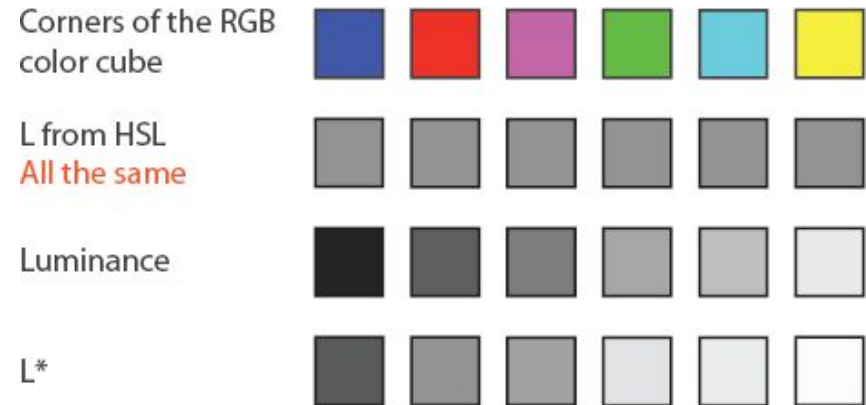


Fig 10.3, VAD

Spectral Sensitivity to Luminance

We are much more sensitive to middle wavelengths of green and yellow than to outer wavelengths of red and blue

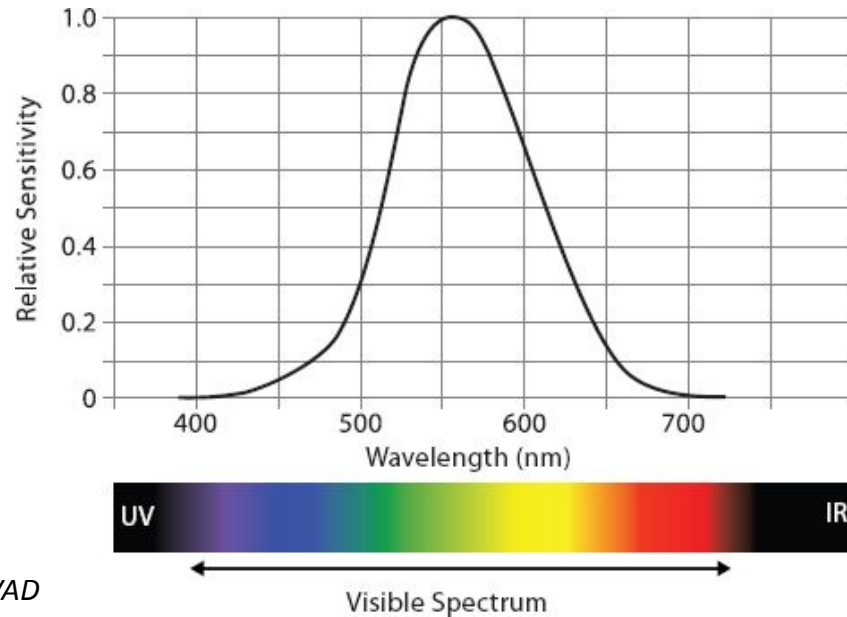


Fig 10.4, VAD

Rainbow Colormaps

- Standard rainbow colormap is perceptually non-linear
- Perceptually linear rainbow colormap
 - seems dingy, rarely used
- Segmented rainbow, good for categorical data



(a)



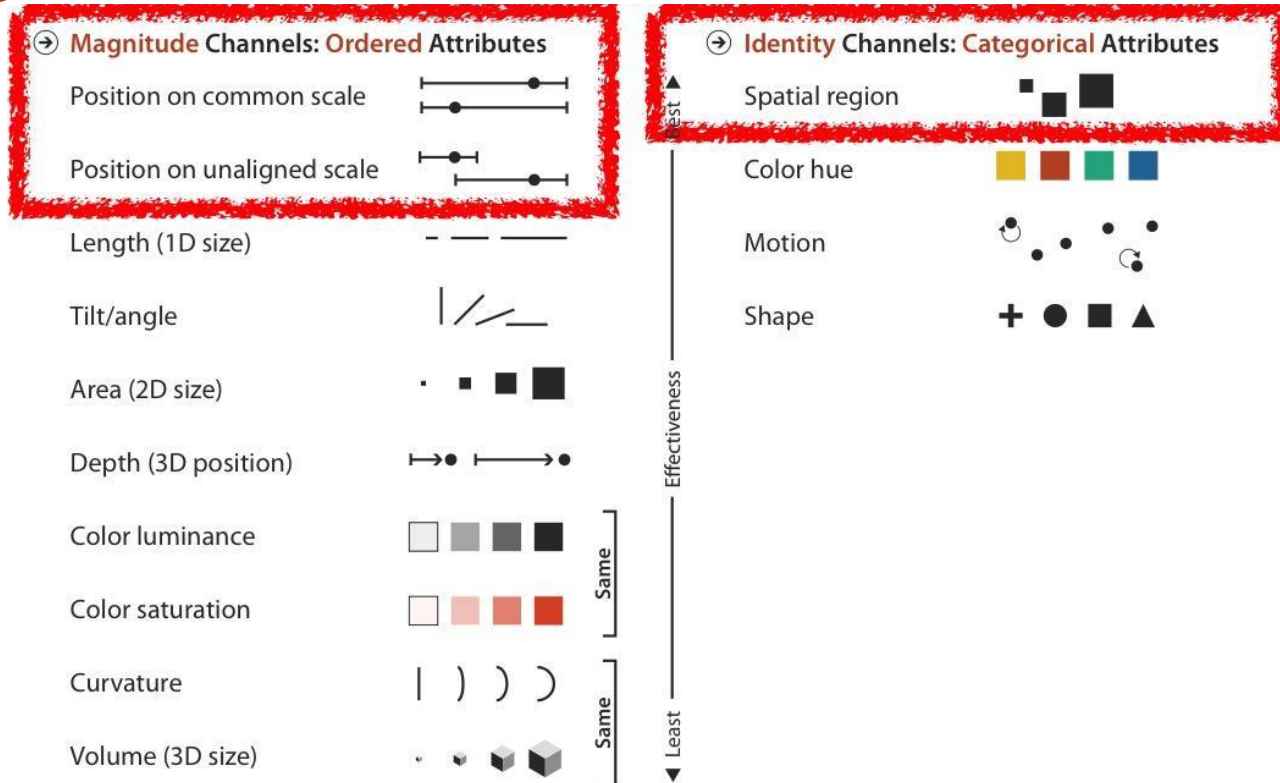
(b)



Fig 10.13, VAD

(c)

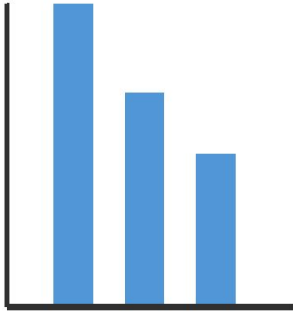
Channels: Expressiveness types and effectiveness rankings



Effectiveness and expressiveness principles

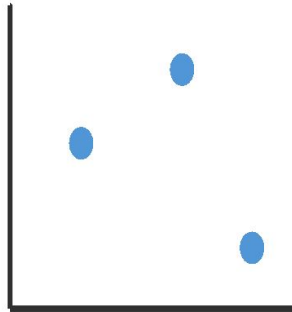
- effectiveness principle
 - encode most important attributes with highest ranked channels
- expressiveness principle
 - match channel and data characteristics
 - identity channel for categorical
 - magnitude channels for ordered

Visual encoding



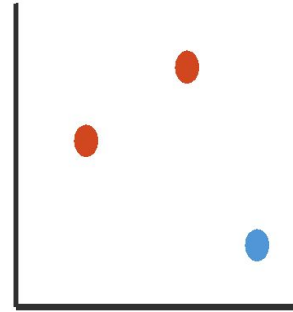
1:
vertical position

mark: line



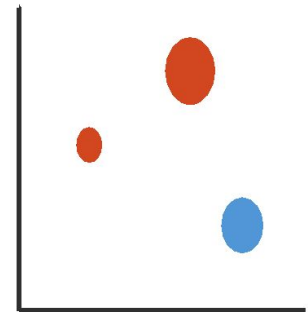
2:
vertical position
horizontal position

mark: point



3:
vertical position
horizontal position
color hue

mark: point



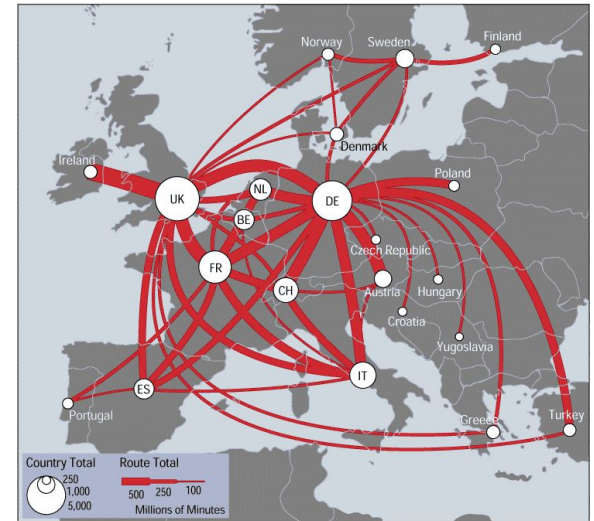
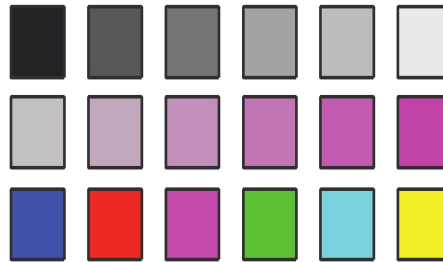
4:
vertical position
horizontal position
color hue
size (area)

mark: point

Discriminability

In a single channel, how many bins (distinguishable steps/levels) are there?

- luminance: < 5
- saturation: about 3
- hue: 6-7
- linewidth: only a few

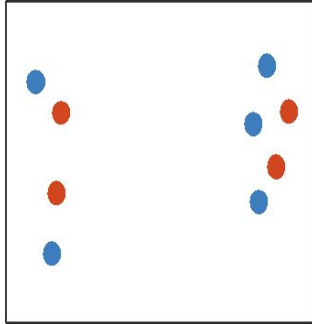


source: [Telecommunications Traffic Flow Map](#)

Separability vs. Integrality

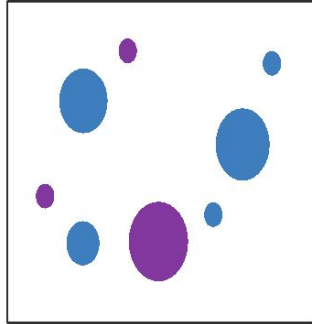
When combining multiple channels, can the viewer separate the values mapped to each channel?

Position
+ Hue (Color)



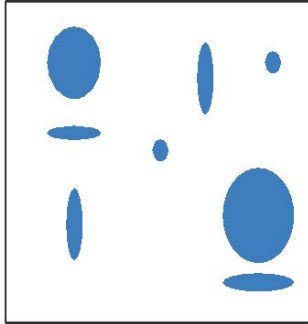
Fully separable

Size
+ Hue (Color)



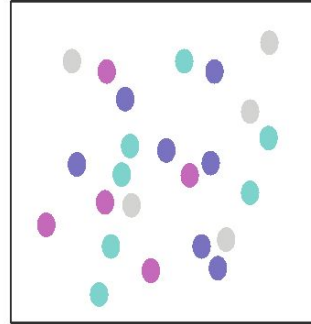
Some interference

Width
+ Height



Some/significant
interference

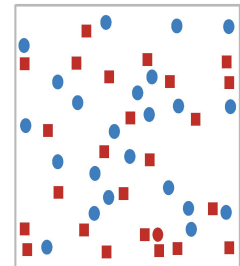
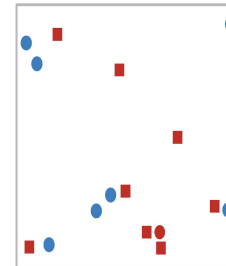
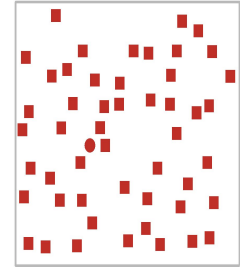
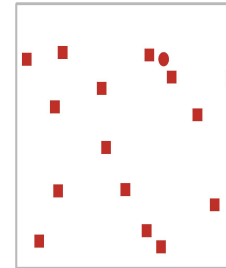
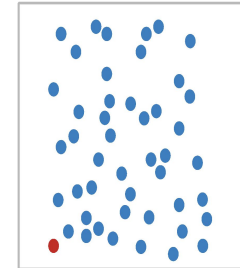
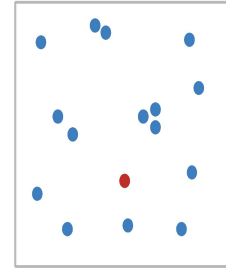
Red
+ Green



Major interference

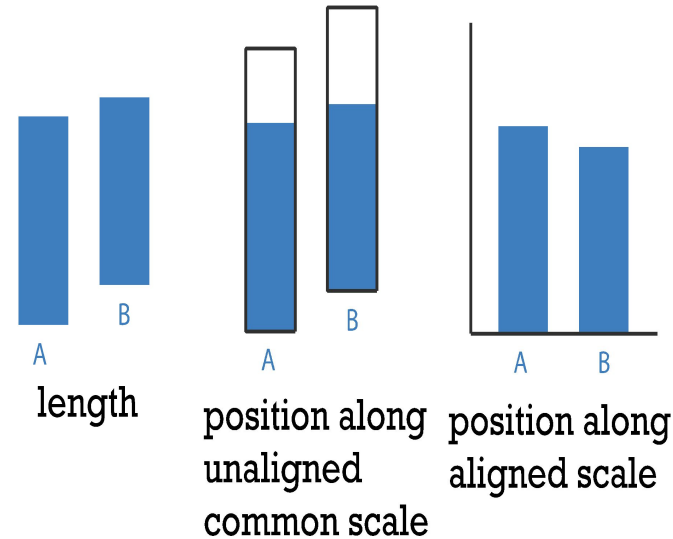
Popout

- find the red dot
 - how long does it take?



Relative vs. absolute judgments

- perceptual system mostly operates with relative judgments, not absolute
 - accuracy increases with common frame/scale and alignment
 - adding frame allows us to judge the length of the unfilled bar
 - sizes have a larger difference than the filled bars
 - aligning achieves the same effect without using a frame



after [Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Cleveland and McGill. Journ. American Statistical Association 79:387 (1984), 531–554.]

How?

Encode

⌚ Arrange

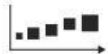
→ Express



→ Separate



→ Order



→ Align



→ Use



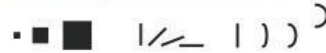
⌚ Map

from **categorical** and **ordered** attributes

→ Color



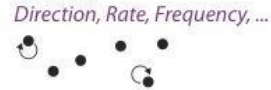
→ Size, Angle, Curvature, ...



→ Shape



→ Motion



Manipulate

⌚ Change



⌚ Select

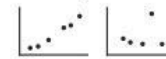


⌚ Navigate

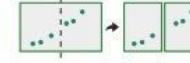


Facet

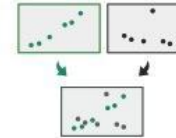
⌚ Juxtapose



⌚ Partition



⌚ Superimpose



Reduce

⌚ Filter



⌚ Aggregate



⌚ Embed



What?

Why?

How?

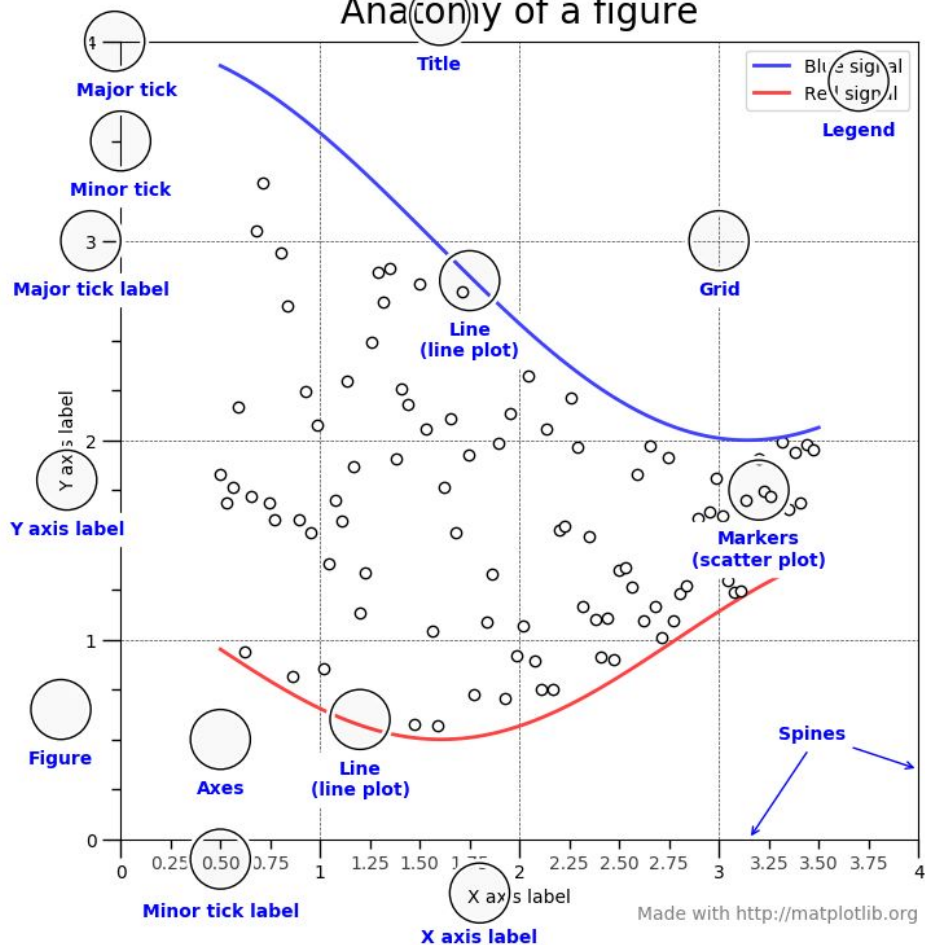
Creating Charts in Python

- Matplotlib, <https://matplotlib.org/>
 - most popular Python plotting library, similar to MATLAB
 - provides full control over the plot
 - developed way before Pandas, so harder to integrate
- Pandas, https://pandas.pydata.org/docs/user_guide/visualization.html
 - visualization in Pandas is a wrapper around Matplotlib
- Seaborn, <https://seaborn.pydata.org/>
 - high-level interface to Matplotlib
 - closely integrated with Pandas
 - allows for more complex charts than Pandas plot()
 - can still use Matplotlib for advanced chart customization

A Few Python Charting Resources

- Matplotlib API reference, <https://matplotlib.org/stable/api/index>
- Matplotlib Tutorials, <https://matplotlib.org/stable/tutorials/index.html>
- Seaborn API reference, <https://seaborn.pydata.org/api.html>
- Seaborn user guide and tutorial, <https://seaborn.pydata.org/tutorial.html>
- Pandas plot API reference, <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>
- Matplotlib + Seaborn + Pandas: An Ideal Amalgamation for Statistical Data Visualisation, <https://towardsdatascience.com/matplotlib-seaborn-pandas-an-ideal-amalgamation-for-statistical-data-visualisation-f619c8e8baa3>
- Visualization with Seaborn, <https://jakevdp.github.io/PythonDataScienceHandbook/04.14-visualization-with-seaborn.html>
- Data Visualization with Pandas and Seaborn, <https://levelup.gitconnected.com/data-visualization-with-pandas-and-seaborn-5de444b567a0>

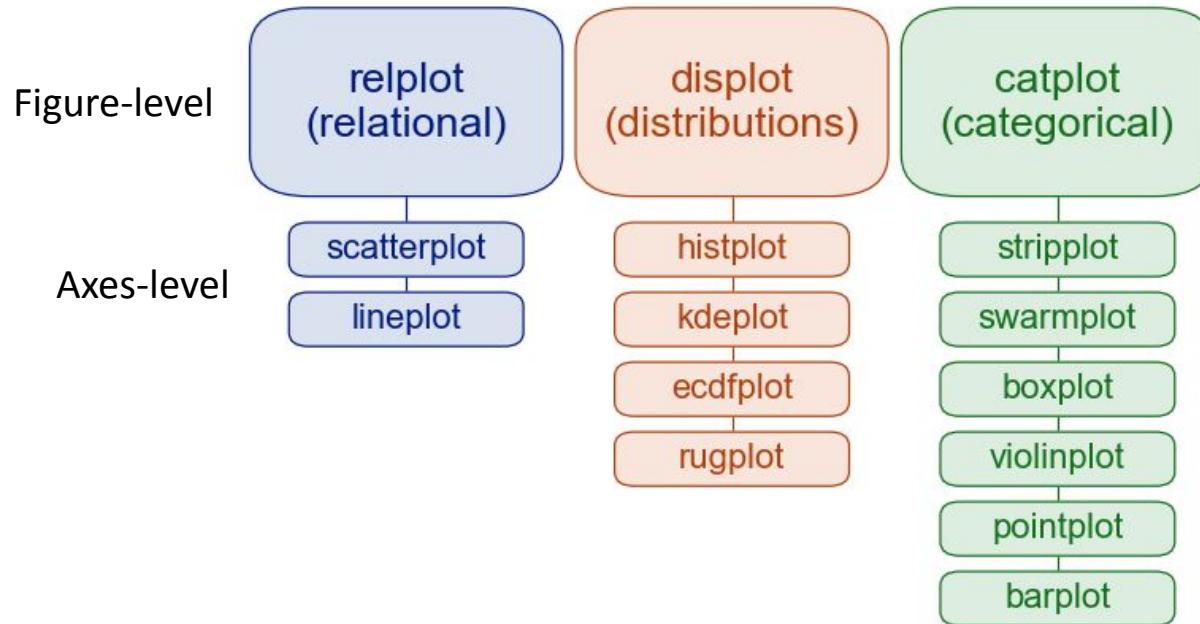
Anatomy of a figure



- **Figure**
 - the whole figure
- **Axes**
 - an actual plot in the figure
 - can be multiple of these (subplots) in a single figure
- **Axis**
 - an actual x/y axis in a specific plot

*Functions in Seaborn are either **Figure-level** or **Axis-level***

Seaborn Figure-Level vs. Axes-Level



https://seaborn.pydata.org/tutorial/function_overview.html#figure-level-vs-axes-level-functions

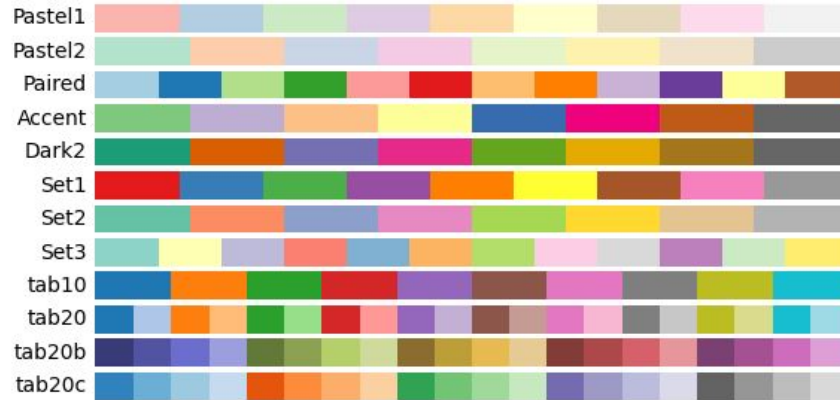
Colormaps in Matplotlib

Perceptually Uniform Sequential colormaps



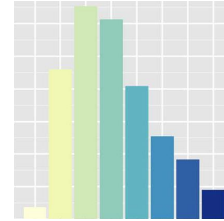
Qualitative colormaps

Qualitative = categorical

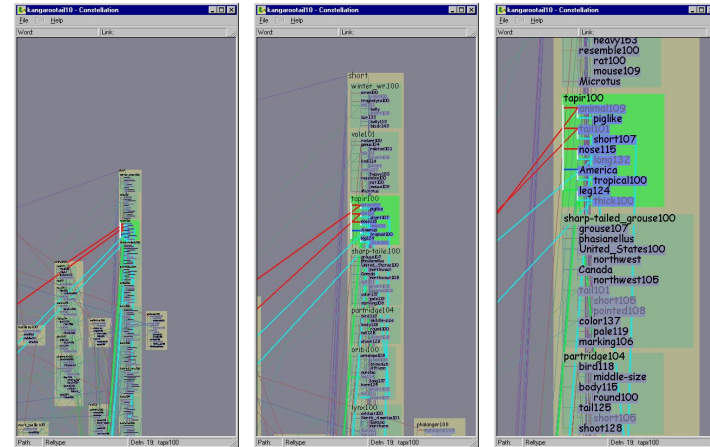


Idiom design space

- **idiom**: distinct approach to creating or manipulating visual representation



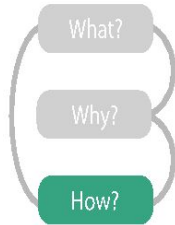
- how to draw it: **visual encoding** idiom
 - many possibilities for how to create
- how to manipulate it: **interaction** idiom
 - even more possibilities
 - make single idiom dynamic
 - link multiple idioms together through interaction



[A layered grammar of graphics. Wickham. *Journal of Computational and Graphical Statistics* 19:1 (2010), 3–28.]

[Interactive Visualization of Large Graphs and Networks. Munzner. Ph.D. thesis, Stanford University Department of Computer Science, 2000.]

Idiom design choices: Part 1



Encode

→ Arrange

→ Express



→ Order



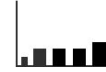
→ Use



→ Separate



→ Align



→ Map

from **categorical** and **ordered** attributes

→ Color



→ Size, Angle, Curvature, ...



→ Shape



→ Motion

Direction, Rate, Frequency, ...



Arrange tables

① Express Values



② Separate, Order, Align Regions

→ Separate



→ Order



→ Align



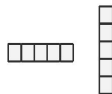
Reminder: Keys and values

- key
 - independent attribute
 - used as unique index to look up items
 - simple tables: 1 key
 - multidimensional tables: multiple keys
- value
 - dependent attribute, value of cell
- classify arrangements by key count
 - 0, 1, 2, many...

→ Express Values



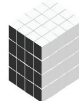
→ 1 Key
List



→ 2 Keys
Matrix



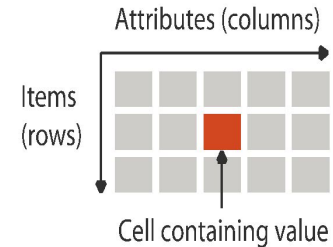
→ 3 Keys
Volume



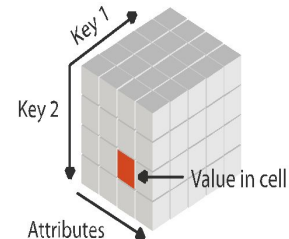
→ Many Keys
Recursive Subdivision



→ Tables



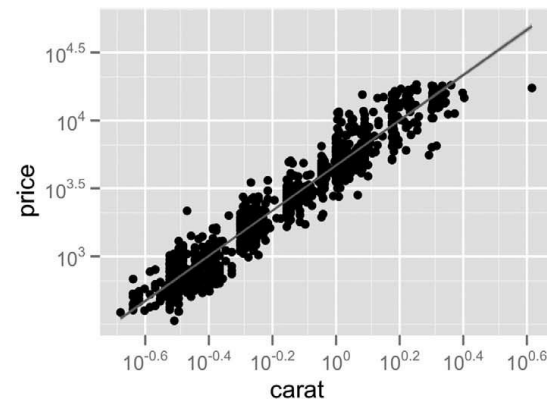
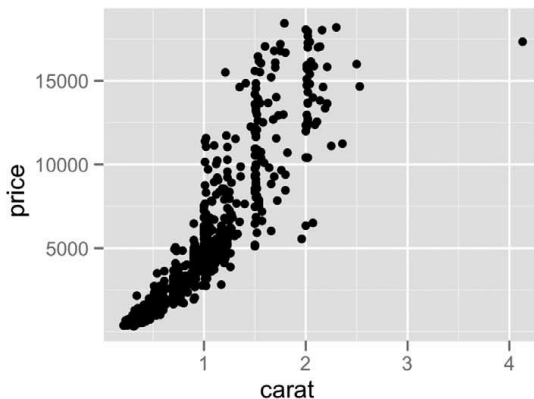
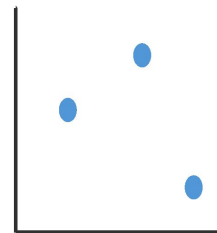
→ Multidimensional Table



Idiom: scatterplot

- ***express*** values
 - quantitative attributes
- no keys, only values
 - data:
 - 2 quant attribs
 - mark: points
 - channels
 - horiz + vert position
 - tasks
 - find trends, outliers, distribution, correlation, clusters
 - scalability: hundreds of items

➔ Express Values



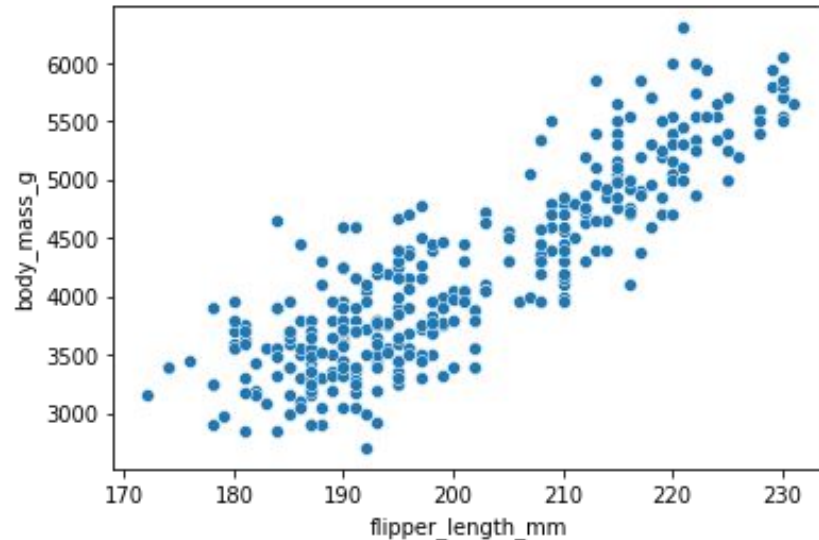
source: [A layered grammar of graphics. Wickham. *Journ. Computational and Graphical Statistics* 19:1 (2010), 3–28.]

Basic Scatterplot using Seaborn

```
sns.scatterplot(data=penguins, x="flipper_length_mm", y="body_mass_g")
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0



notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Some keys: Categorical regions

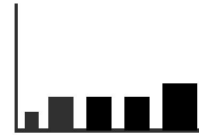
→ Separate



→ Order



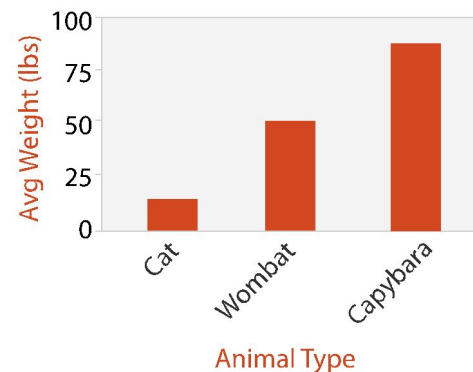
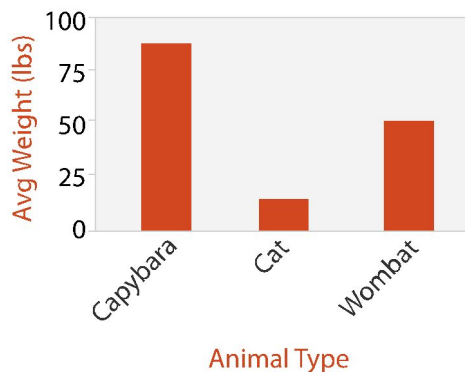
→ Align



- **regions**: contiguous bounded areas distinct from each other
 - using space to *separate* (proximity)
 - following expressiveness principle for categorical attributes
- use ordered attribute to *order* and *align* regions

Idiom: bar chart

- one key, one value
 - data
 - 1 categ attrib, 1 quant attrib
 - mark: lines
 - channels: length to express quant value
 - spatial regions: one per mark
 - **separated** horizontally, **aligned** vertically
 - **ordered** by quant attrib
 - » by label (alphabetical), by length attrib (data-driven)
 - task
 - compare, lookup values
 - scalability
 - dozens to hundreds of levels for key attrib

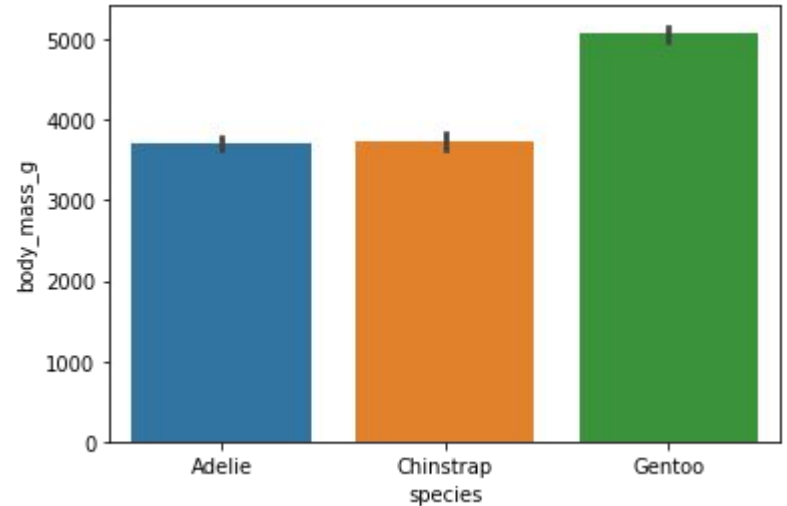


Basic Bar Chart using Seaborn

```
sns.barplot(data=penguins, x="species", y="body_mass_g");
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

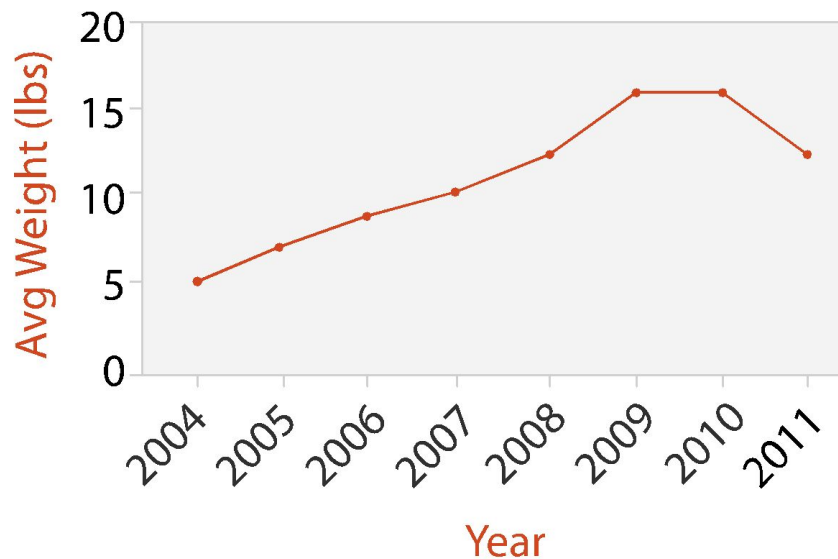
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0



notebook: <https://colab.research.google.com/drive/1Y2IJFV4NgzLc2J3pxvP34VtqUw7jKmEe>

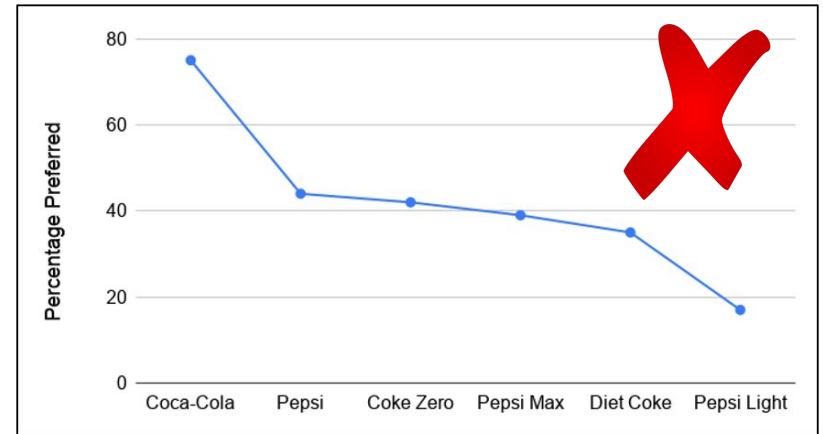
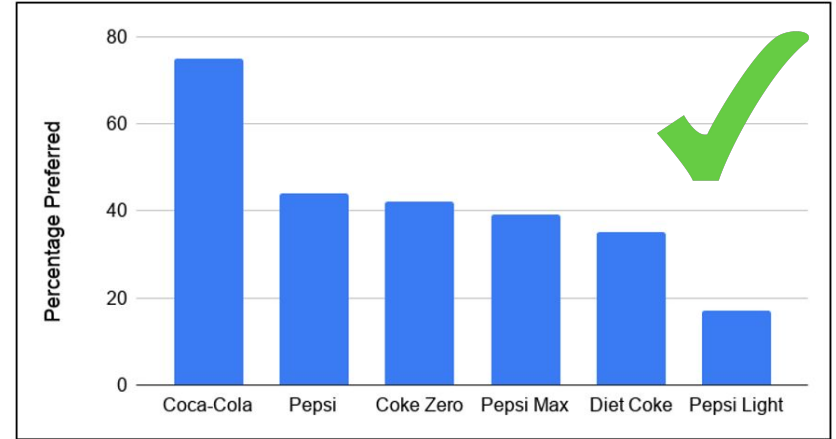
Idiom: line chart

- one key, one value
 - data
 - 1 quant attrib, 1 ordered attrib
 - mark: points
 - line connection marks between them
 - channels
 - aligned vertical position
 - separated and ordered by key attrib into horizontal regions
 - task
 - find trend
 - connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next



Choosing bar vs line charts

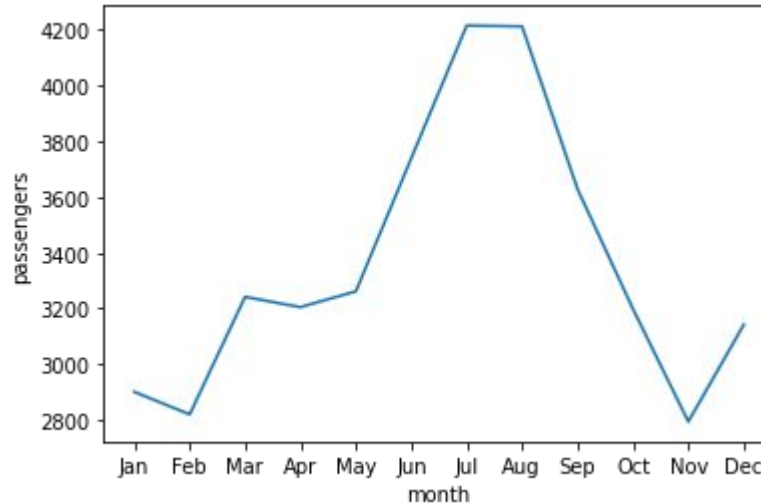
- depends on type of key attrib
 - bar charts if categorical
 - line charts if ordered
- do not use line charts for categorical key attribs
 - violates expressiveness principle
 - implication of trend so strong that it overrides semantics!



Basic Line Chart using Seaborn

```
sns.lineplot(data=per_month, x="month", y="passengers")
```

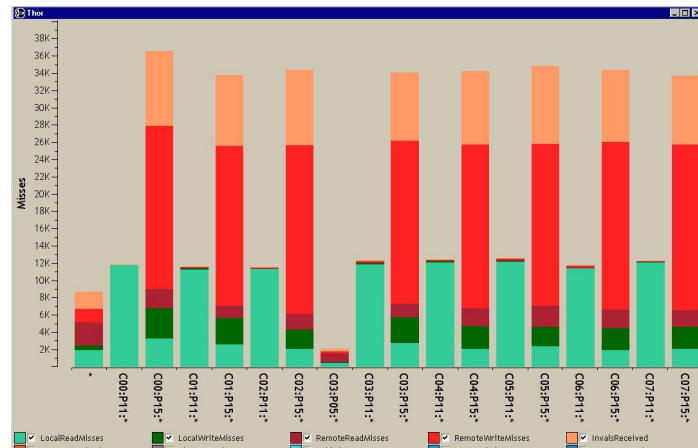
per_month		
	month	passengers
0	Jan	2901
1	Feb	2820
2	Mar	3242
3	Apr	3205
4	May	3262



notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Idiom: stacked bar chart

- data: 2 categ attrib, 1 quant attrib
- mark: vertical stack of line marks
 - **glyph**: composite object, internal structure from multiple marks
- channels
 - length and color hue
 - spatial regions: one per glyph
 - aligned: full glyph, lowest bar component
 - unaligned: other bar components
- task: part-to-whole relationship
- scalability: several to one dozen levels for stacked attrib

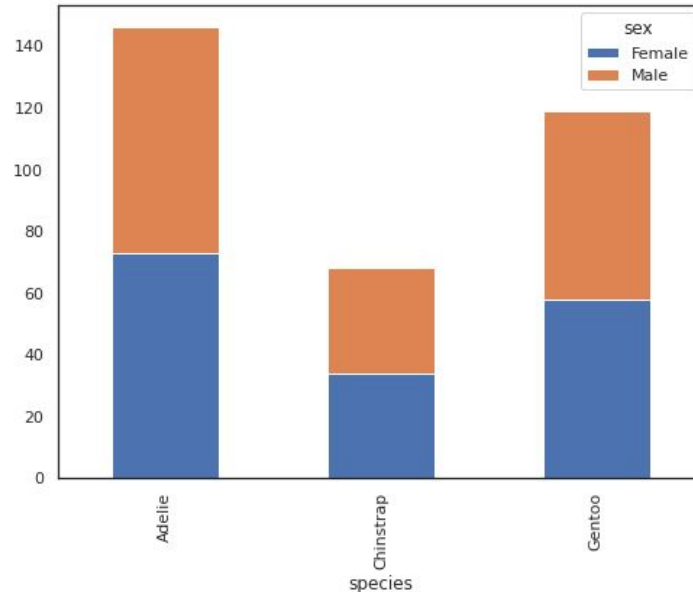


source: [Using Visualization to Understand the Behavior of Computer Systems. Bosch. Ph.D. thesis, Stanford Computer Science, 2001.]

Stacked Bar Chart using Pandas Plot

```
penguin_pivot.plot.bar(stacked=True)
```

penguin_pivot		
sex	Female	Male
species		
Adelle	73	73
Chinstrap	34	34
Gentoo	58	61



Idiom: Grouped Bar Chart

- data: 2 categ attrib, 1 quant attrib
- mark: multibar glyph in each region
- channels
 - length and color hue
 - spatial regions: one per glyph
- task: compare, lookup values (same as bar chart)

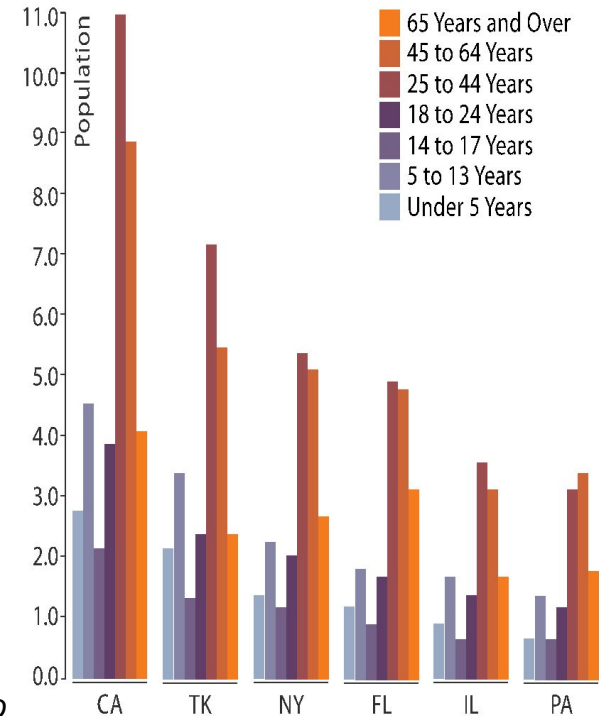


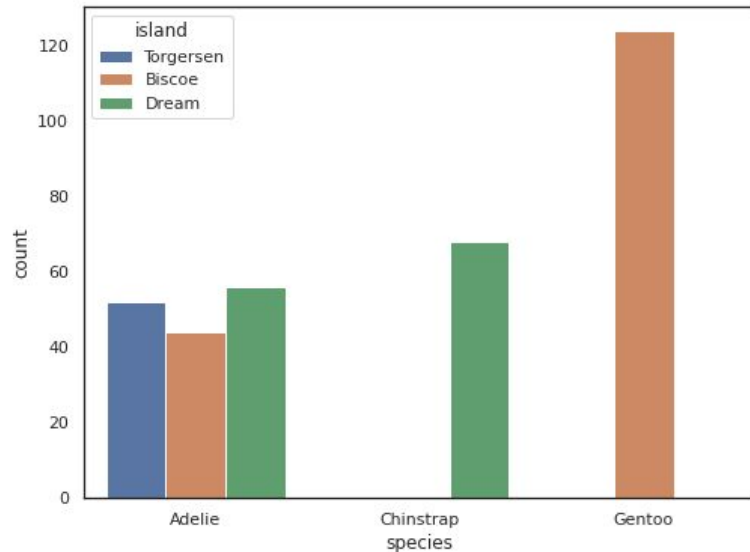
Fig 12.8a, VAD

Grouped Bar Chart using Seaborn

```
sns.countplot(data=penguins, x="species", hue="island")
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	39.1	18.7
1	Adelie	Torgersen	39.5	17.4
2	Adelie	Torgersen	40.3	18.0

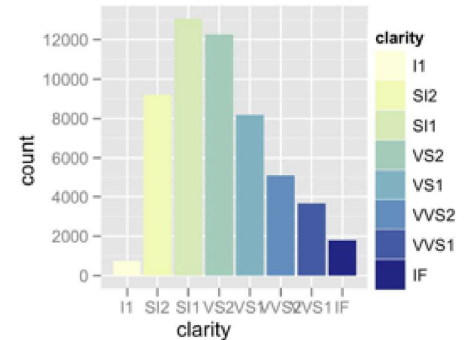
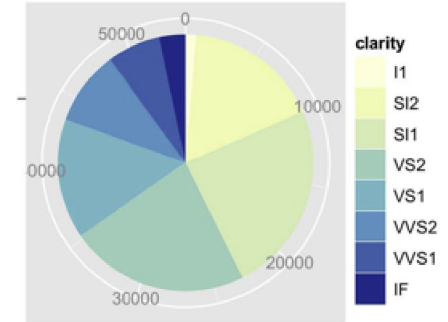


notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Idioms: pie chart

- pie chart
 - area marks with angle channel
 - accuracy: angle/area much less accurate than line length
- data
 - 1 categ key attrib, 1 quant value attrib
- task
 - part-to-whole judgments

Think carefully before choosing pie charts when there are more than 3-4 categories.



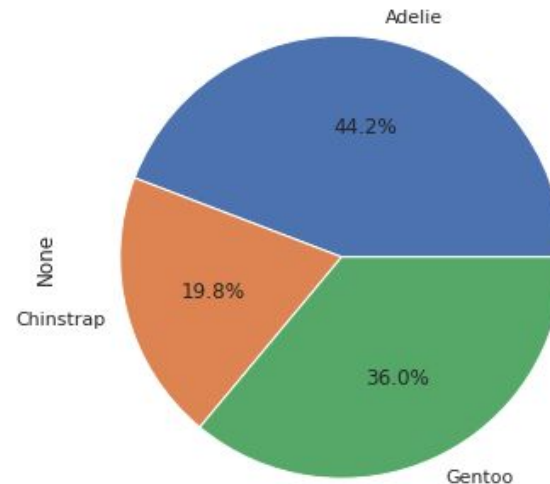
[A layered grammar of graphics. Wickham. Journ. Computational and Graphical Statistics 19:1 (2010), 3–28.]

Pie Chart using Pandas Plot

```
data.plot.pie(autopct="% .1f%%")
```

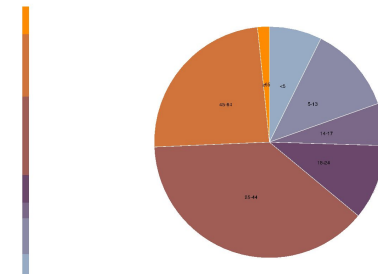
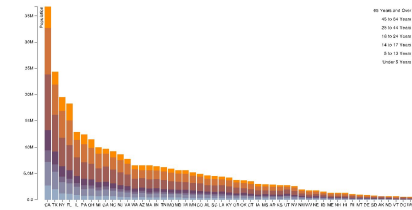
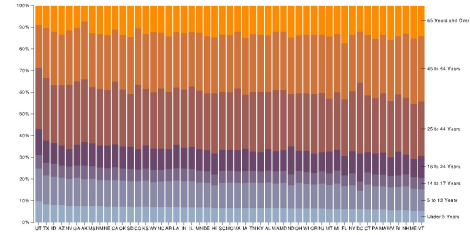
```
data = penguins.groupby("species").size()
data

species
Adelie      152
Chinstrap   68
Gentoo     124
dtype: int64
```



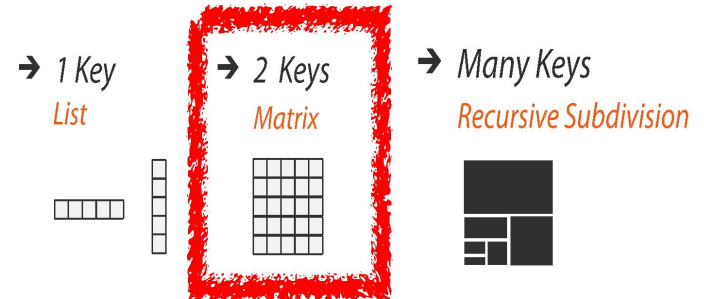
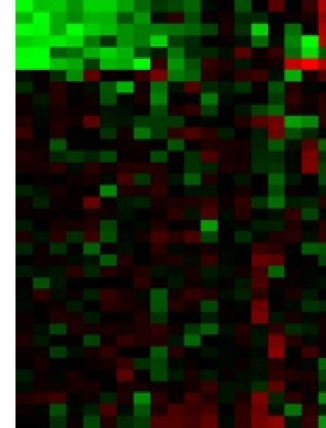
Idioms: normalized stacked bar chart

- task
 - part-to-whole judgments
- normalized stacked bar chart
 - stacked bar chart, normalized to full vert height
 - single stacked bar equivalent to full pie
 - high information density: requires narrow rectangle
- pie chart
 - information density: requires large circle



Idiom: heatmap

- two keys, one value
 - data
 - 2 categ attribs (gene, experimental condition)
 - 1 quant attrib (expression levels)
 - marks: area
 - separate and align in 2D matrix
 - indexed by 2 categorical attributes
 - channels
 - color by quant attrib
 - (ordered diverging colormap)
 - task
 - find clusters, outliers
 - scalability
 - 1M items, 100s of categ levels, ~10 quant attrib levels

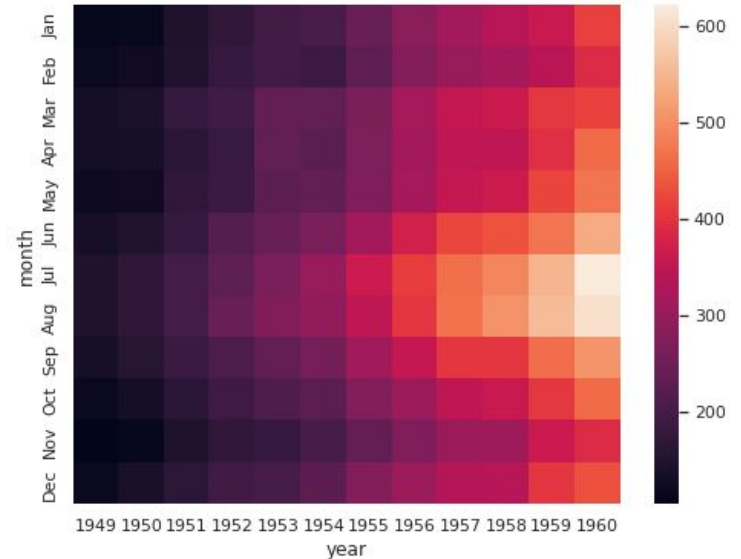


Heatmap using Seaborn

`sns.heatmap(pivot)`

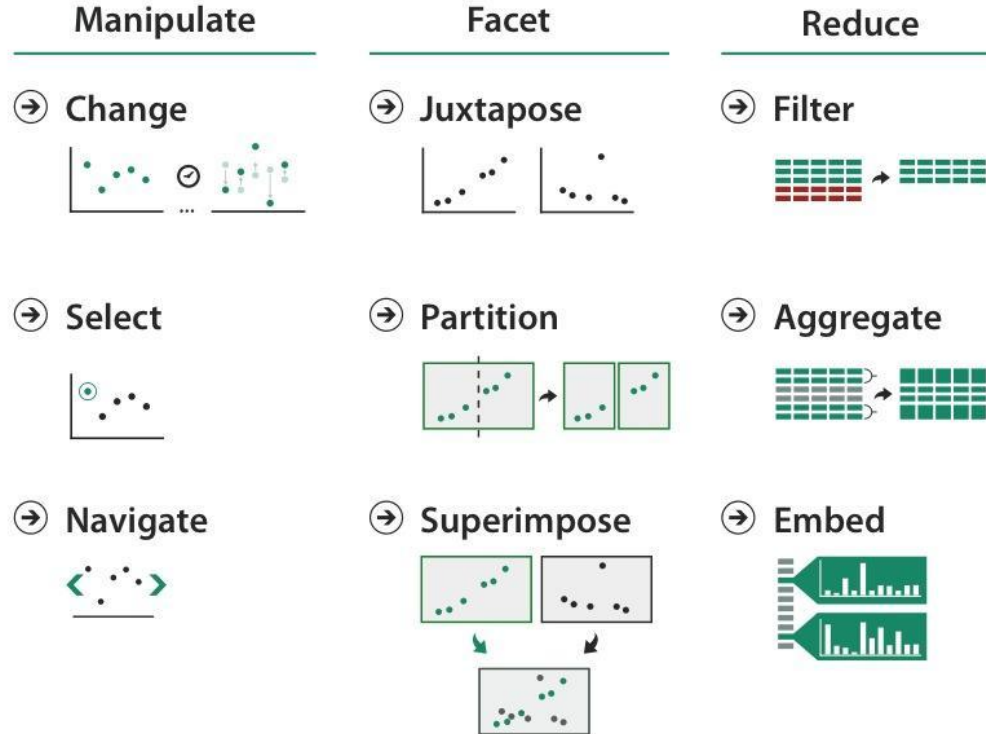
```
pivot = flights.pivot("month", "year", "passengers")
pivot.head()
```

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472



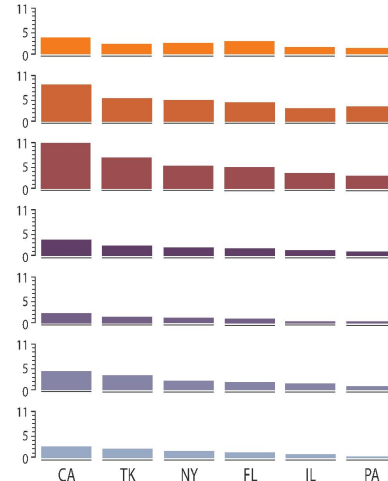
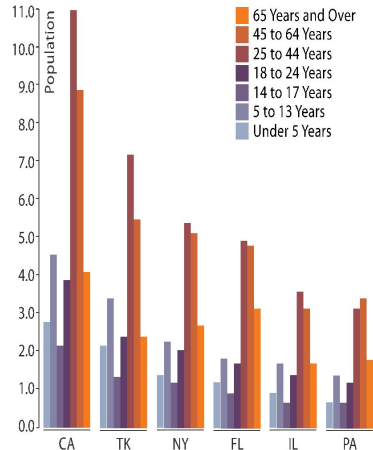
notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Idiom design choices: Part 2



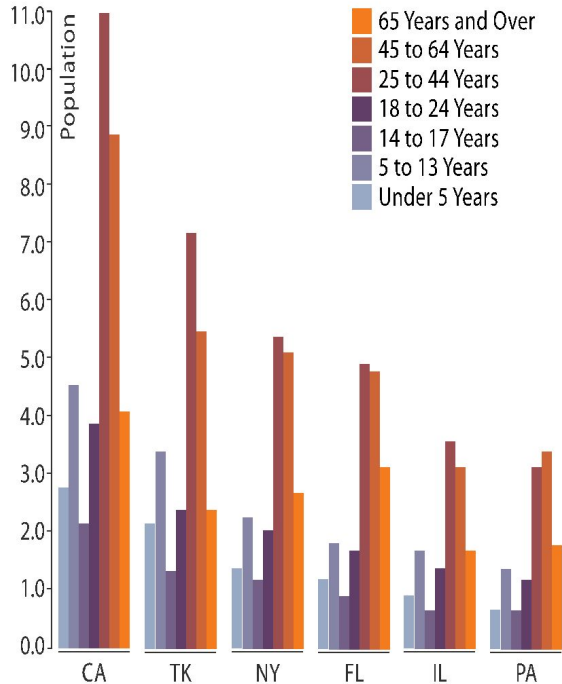
Partitioning: List alignment

- single bar chart with grouped bars
 - split by state into regions
 - complex glyph within each region showing all ages
 - compare: easy within state, hard across ages
- small-multiple bar charts
 - split by age into regions
 - one chart per region
 - compare: easy within age, harder across states

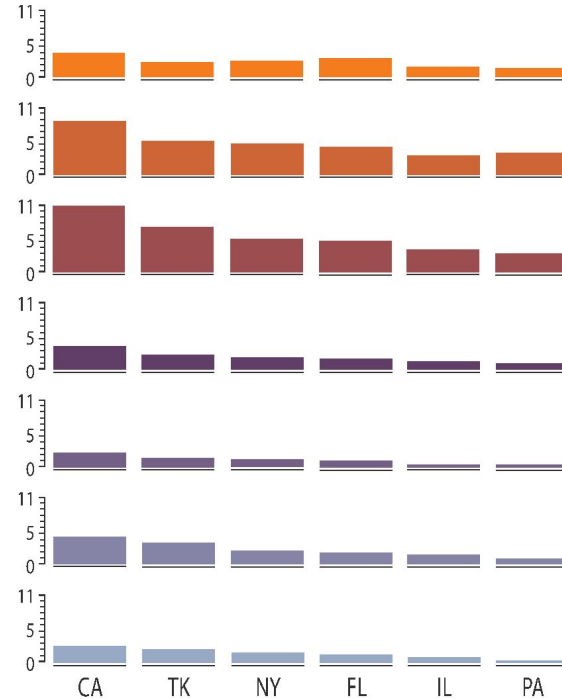


Partitioning: List alignment and comparison

easy within state, hard across ages

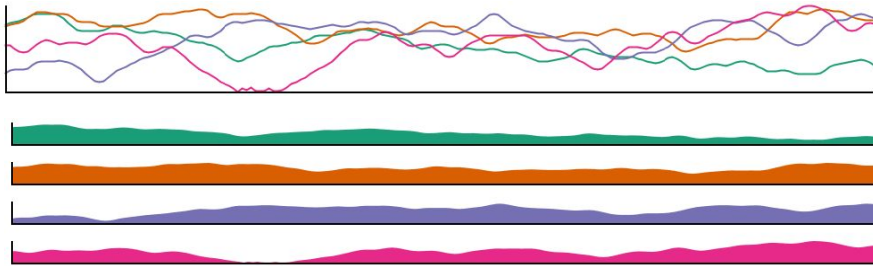


easy within age, harder across states

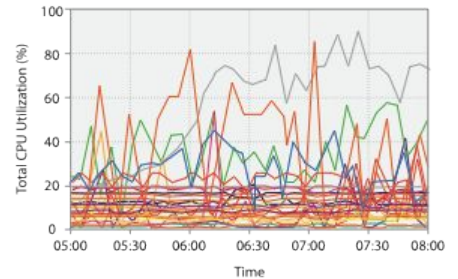
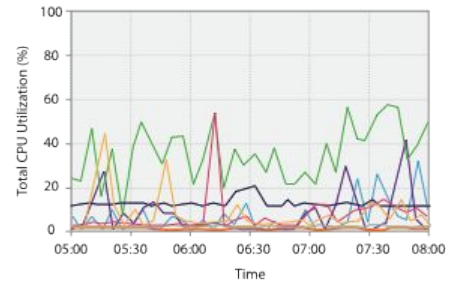
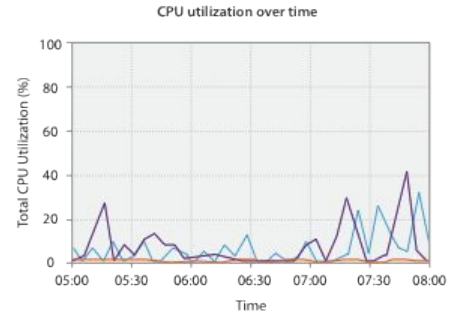


Superimposing limits

- few layers, but many lines
 - up to a few dozen, but not hundreds
- superimpose vs juxtapose: empirical study
 - superimposed for local visual, multiple for global
 - same screen space for all multiples, single superimposed
 - tasks
 - local: maximum, global: slope, discrimination



[Graphical Perception of Multiple Time Series. Javed, McDonnel, and Elmqvist. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE InfoVis 2010) 16:6 (2010), 927–934.]

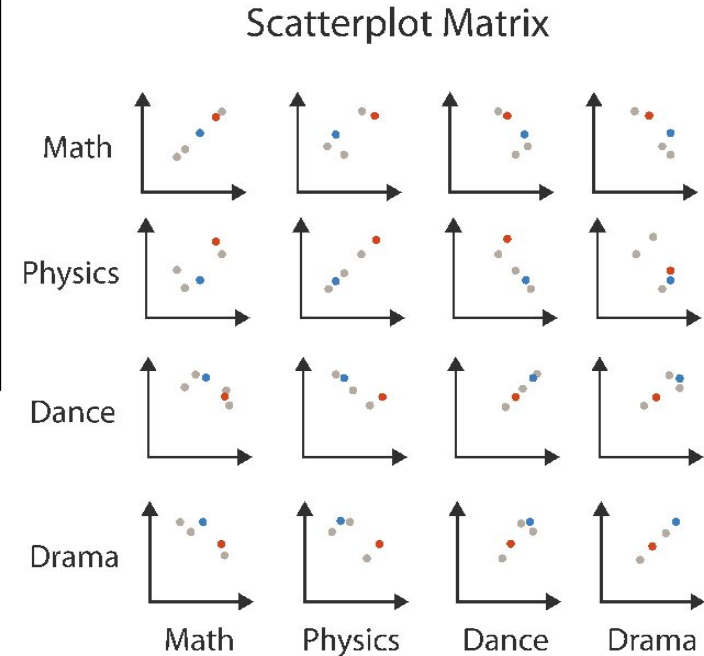


Idiom: scatterplot matrix

- scatterplot matrix (SPLOM)

- rectilinear axes, point mark
- all possible pairs of axes
- scalability
 - one dozen attribs
 - dozens to hundreds of items
- task: correlation
 - positive correlation: diagonal low-to-high
 - negative correlation: diagonal high-to-low
 - uncorrelated

Table			
Math	Physics	Dance	Drama
85	95	70	65
90	80	60	50
65	50	90	90
50	40	95	80
40	60	80	90



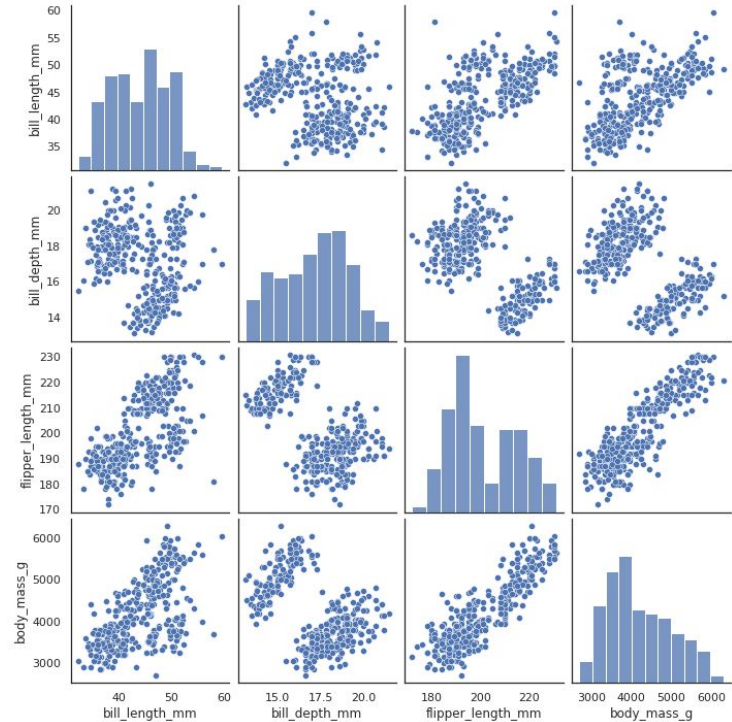
source: [[Visualization Course Figures](#). McGuffin, 2014]

Scatterplot Matrix using Seaborn

```
sns.pairplot(penguins)
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0



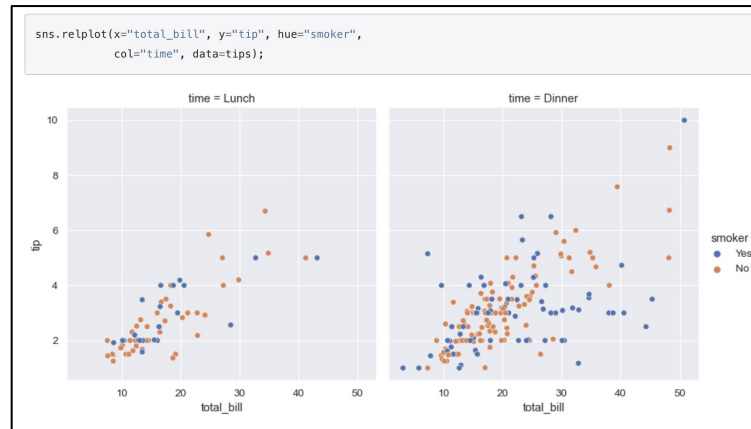
notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

More Multiple Views in Seaborn

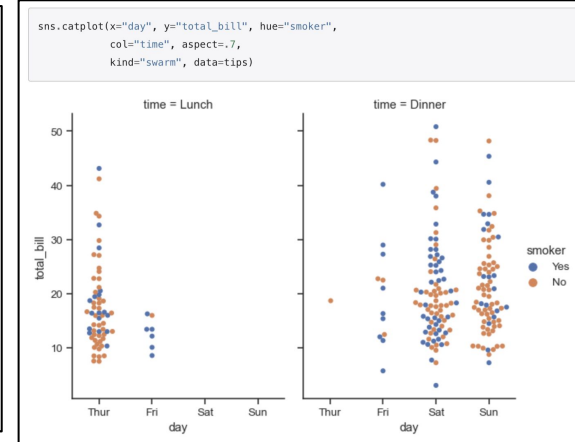
jointplot()



relplot()



catplot()



https://seaborn.pydata.org/tutorial/function_overview.html#combining-multiple-views-on-the-data
<https://seaborn.pydata.org/tutorial/relational.html#showing-multiple-relationships-with-facets>
<https://seaborn.pydata.org/tutorial/categorical.html#showing-multiple-relationships-with-facets>

Reduce items and attributes

- filter
 - pro: straightforward and intuitive
 - to understand and compute
 - con: out of sight, out of mind
- aggregation
 - pro: inform about whole set
 - con: difficult to avoid losing signal
- not mutually exclusive
 - combine filter, aggregate
 - combine reduce, change, facet

Reducing Items and Attributes

Filter

→ Items



→ Attributes

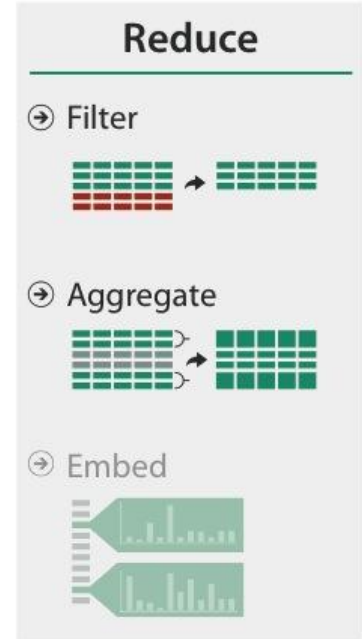
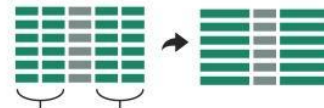


Aggregate

→ Items

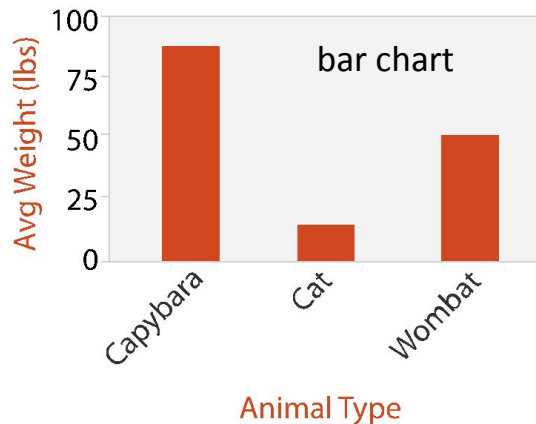
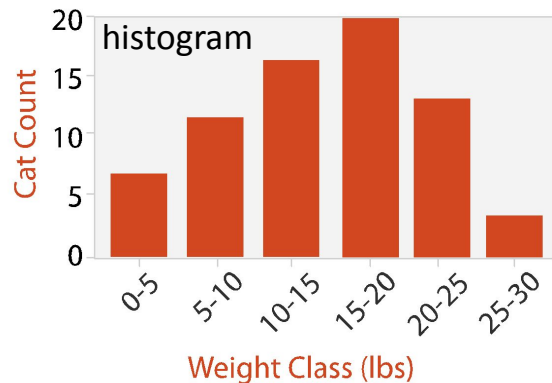


→ Attributes



Idiom: histogram

- static item aggregation
- task: find distribution
- data: table
- derived data
 - new table: keys are bins, values are counts
- bin size crucial
 - pattern can change dramatically depending on discretization (bin size)

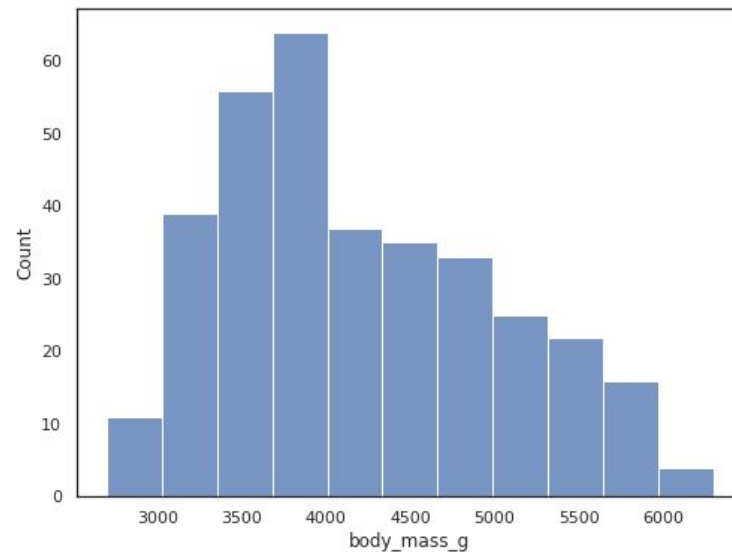


Histogram using Seaborn

```
sns.histplot(penguins['body_mass_g'])
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

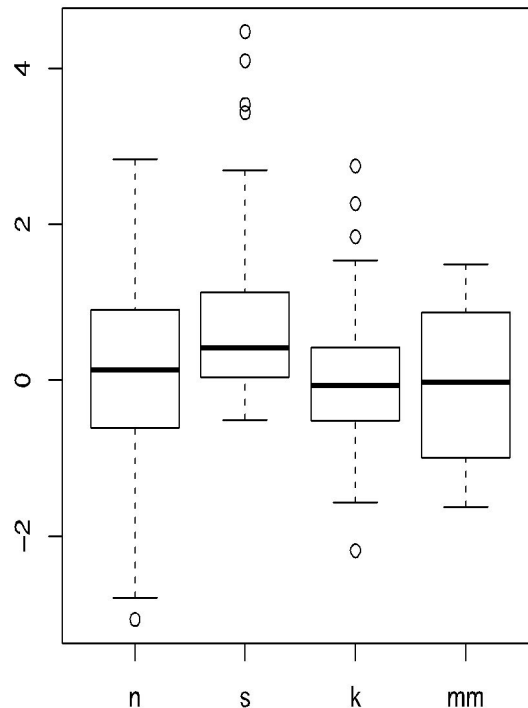
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0



notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Idiom: **boxplot**

- static item aggregation
- task: find distribution
- data: table
- derived data
 - 5 quant attribs
 - median: central line
 - lower and upper quartile: boxes
 - lower upper fences: whiskers
 - values beyond which items are outliers
 - outliers beyond fence cutoffs explicitly shown



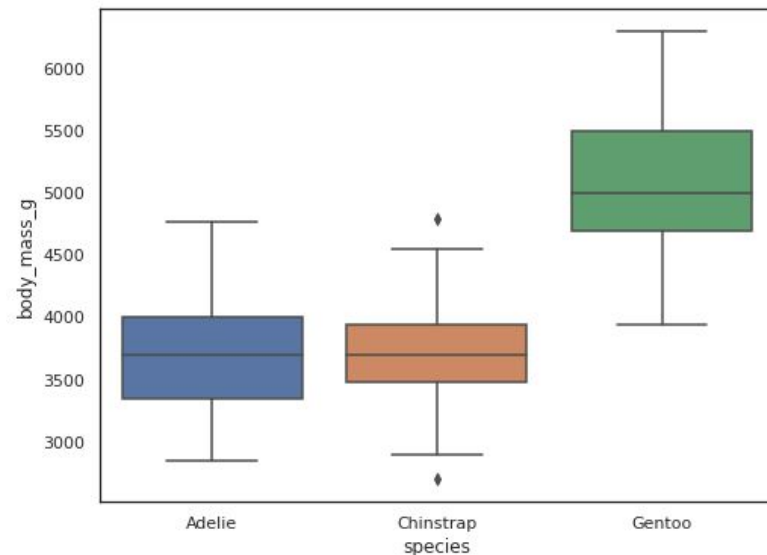
source: [[40 years of boxplots](#). Wickham and Stryjewski. 2012]

Boxplot using Seaborn

```
sns.boxplot(data=penguins, x="species", y="body_mass_g")
```

```
penguins = sns.load_dataset("penguins")  
penguins.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0



notebook: <https://colab.research.google.com/drive/1Y2IJFV4NqzLc2J3pxvP34VtqUw7jKmEe>

Exercise - Scatterplot

- Divide into groups of 2
- Examine the Seaborn dataset `car_crashes`
 - https://github.com/mwaskom/seaborn-data/blob/master/car_crashes.csv
 - original description:
<https://www.kaggle.com/datasets/fivethirtyeight/fivethirtyeight-bad-drivers-dataset>
- Use Seaborn to generate a **scatterplot** that examines two of the quantitative attributes
 - use informative x, y axis labels and an informative title
- What conclusions can you draw from the chart?
- What further questions do you have after visualizing the data?

Exercise - Bar Chart

- Divide into groups of 2
- Examine the Seaborn dataset `car_crashes`
 - https://github.com/mwaskom/seaborn-data/blob/master/car_crashes.csv
 - original description:
<https://www.kaggle.com/datasets/fivethirtyeight/fivethirtyeight-bad-drivers-dataset>
- Use Seaborn to generate a horizontal **bar chart** that shows the top 10 states in terms of one of the quantitative values
 - use informative x, y axis labels and an informative title
- What conclusions can you draw from the chart?
- What further questions do you have after visualizing the data?

Exercise - Boxplot

- Divide into groups of 2
- Examine the Seaborn dataset tips
 - <https://github.com/mwaskom/seaborn-data/blob/master/tips.csv>
- Use Seaborn to generate a **boxplot** that shows the distributions of percent tip (*derived attribute*) for Lunch and Dinner
 - use informative x, y axis labels and an informative title
- What conclusions can you draw from the chart?
- What further questions do you have after visualizing the data?

Exercise - Think about Data

- Divide into groups of 2
- Examine the Seaborn dataset tips
 - <https://github.com/mwaskom/seaborn-data/blob/master/tips.csv>
- How would you use the data to answer the question, *what factors play into getting the highest percent tips?*
 - what data manipulations would you need to perform?
 - what type of chart would be appropriate to show the results? why?

You don't actually have to generate a chart, just plan out what you'd need to do.

Interested in learning more?

Watch Tamara Munzner's D3 Unconference 2015 Keynote (55 min),
<https://youtu.be/jVC6SQS23ak> or her full-day tutorial at IEEE VIS 2021 (18 videos),
https://www.youtube.com/playlist?list=PLT4XLHmqHJBdB24LAQPk_PV7wrwpJFh5a

Still interested?

Check out ODU CS 625 - Data Visualization

<https://github.com/odu-cs625-datavis/public/tree/main/fall21#readme>

covers most of Munzner's textbook

Still interested?

Check out ODU CS 725/825 - Information Visualization and Visual Analytics

<https://github.com/odu-cs725-infovis/public/tree/main/spr22#readme>

more advanced visualizations, research-based, assumes knowledge of all material from CS 625