Programming Principles in Python (CSCI 503)

Machine Learning

Dr. David Koop





Grammar of Graphics & Altair

- "Grammar of Graphics", L. Wilkinson
- "A Layered Grammar of Graphics" + ggplot, H. Wickham
- Vega: "Declarative language for creating, saving, and sharing interactive visualization designs"
- Vega-Lite: higher-level language than Vega, carefully crafted rules for defaults
- Altair: Python interface to Vega-Lite (J. VanderPlas)
 - "spend more time understanding your data and its meaning"
 - Specify the what, minimize the amount of code directing the how
 - Python can write JSON specification just as well as any other language
 - Bindings make it more Python-friendly, integrate with pandas, add support for Jupyter, etc.









Basic Example

- import altair as alt import pandas as pd data = pd.DataFrame({'x': [1,3,4,6,10], 'y': [1,5,2,7,3]) alt.Chart(data).mark line().encode(x='x', y='y')
- Easiest to use data from a pandas data frame
 - Another option is a csv or json file
 - Can support geo_interface, too
- Chart is the basic unit
- Mark: .mark * () indicates the geometry created for each data item • Encode: .encode() allows visual properties to be set to data attributes









Visual Marks

 \rightarrow Points

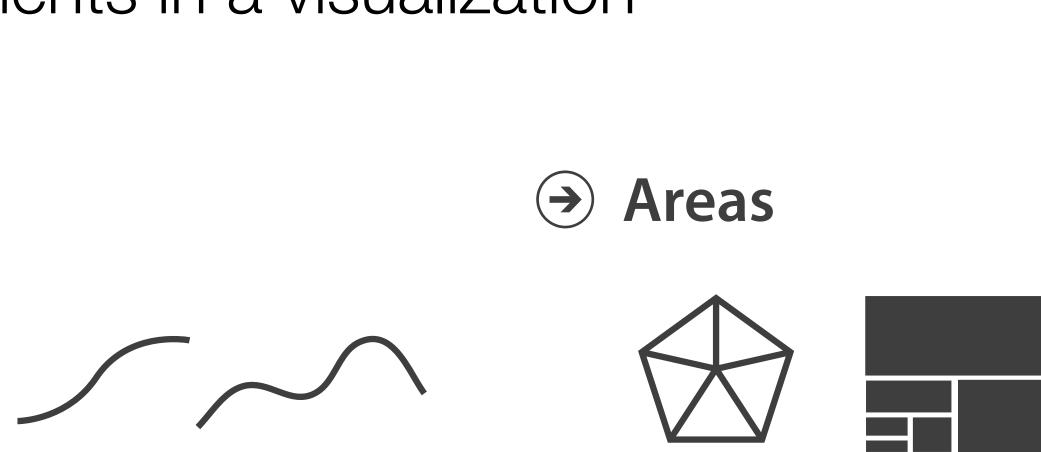
- Marks are the basic graphical elements in a visualization
- Marks classified by dimensionality:

• Also can have surfaces, volumes

 Think of marks as a mathematical definition, or if familiar with tools like Adobe Illustrator or Inkscape, the path & point definitions

 \rightarrow Lines

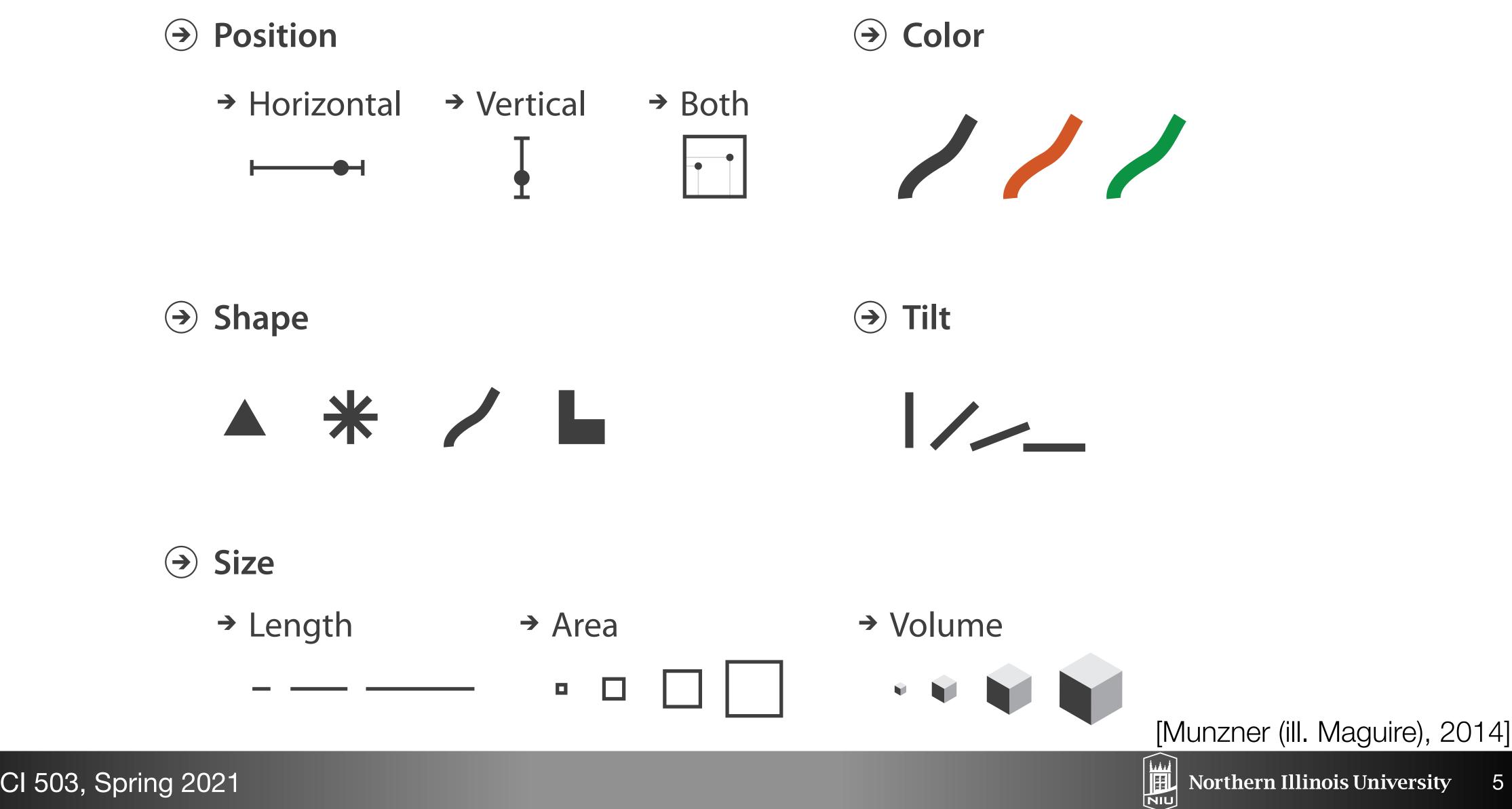
- Altair: area, bar, circle, geoshape, image, line, point, rect, rule, square, text, tick
- Also compound marks: boxplot, errorband, errorbar





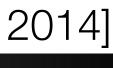


Encode via Visual Channels



D. Koop, CSCI 503, Spring 2021





5

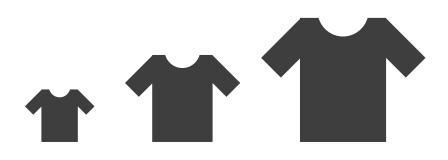
Data Attributes and Altair Types

Categorical

D. Koop, CSCI 503, Spring 2021

→ Ordered

→ Ordinal



→ Quantitative















Data Attributes and Altair Types

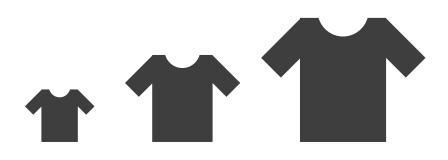
Categorical

- Categorical data = Nominal (N)
- Ordinal data = Ordinal (O)
- Quantitative data = Quantitative (Q)
- Temporal data = Temporal (T)

D. Koop, CSCI 503, Spring 2021

→ Ordered

→ Ordinal



→ Quantitative







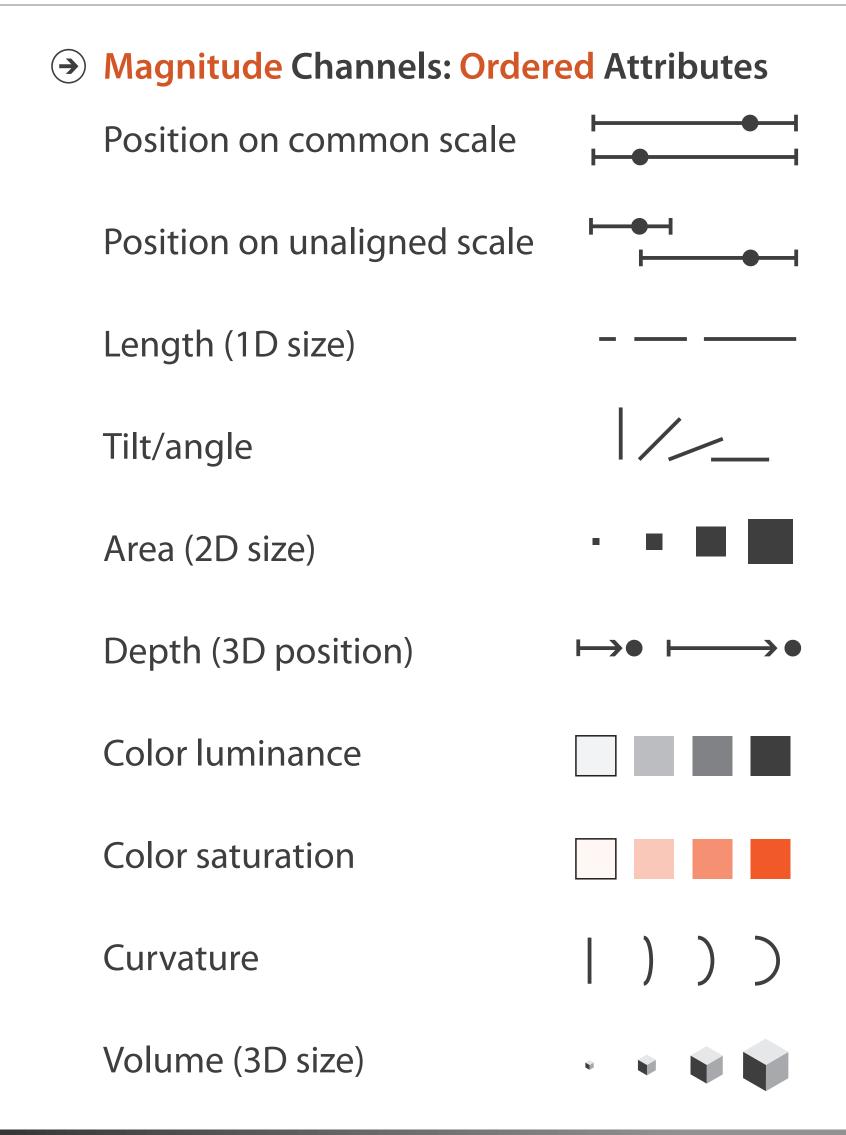






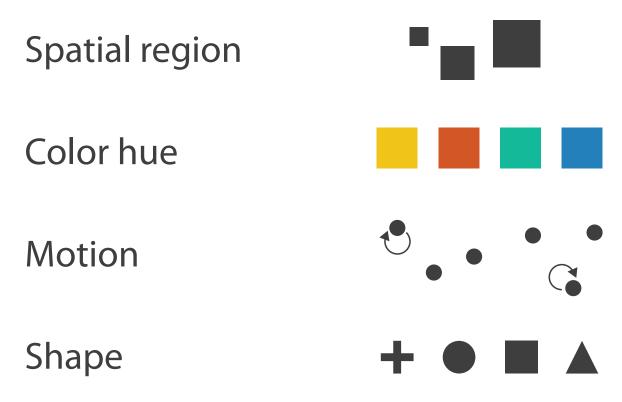


Different Channels for Different Attribute Types



D. Koop, CSCI 503, Spring 2021

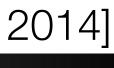
Identity Channels: Categorical Attributes



Altair will use its rules to pick whether to use color hue or saturation based on the type

[Munzner (ill. Maguire), 2014]





Altair Supports Concatenation, Layering, & Repetition

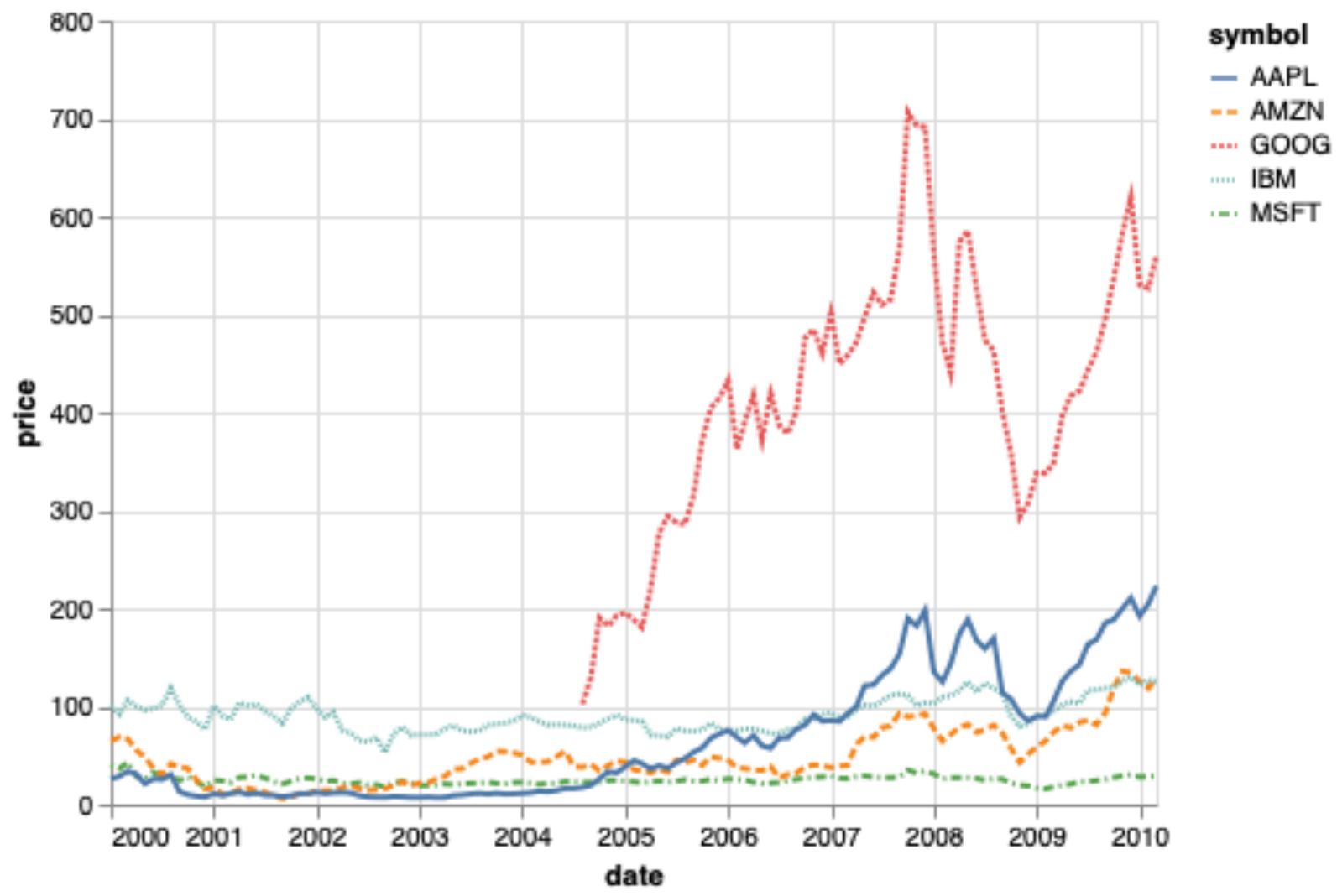
- Layering:
 - + Operator
- Concatenation:
 - Horizontal: | operator
 - Vertical: & operator
- Repetition
 - Use of .repeat for layout
 - Reference repeated variables in the encoding







Layering



- ·-· MSFT

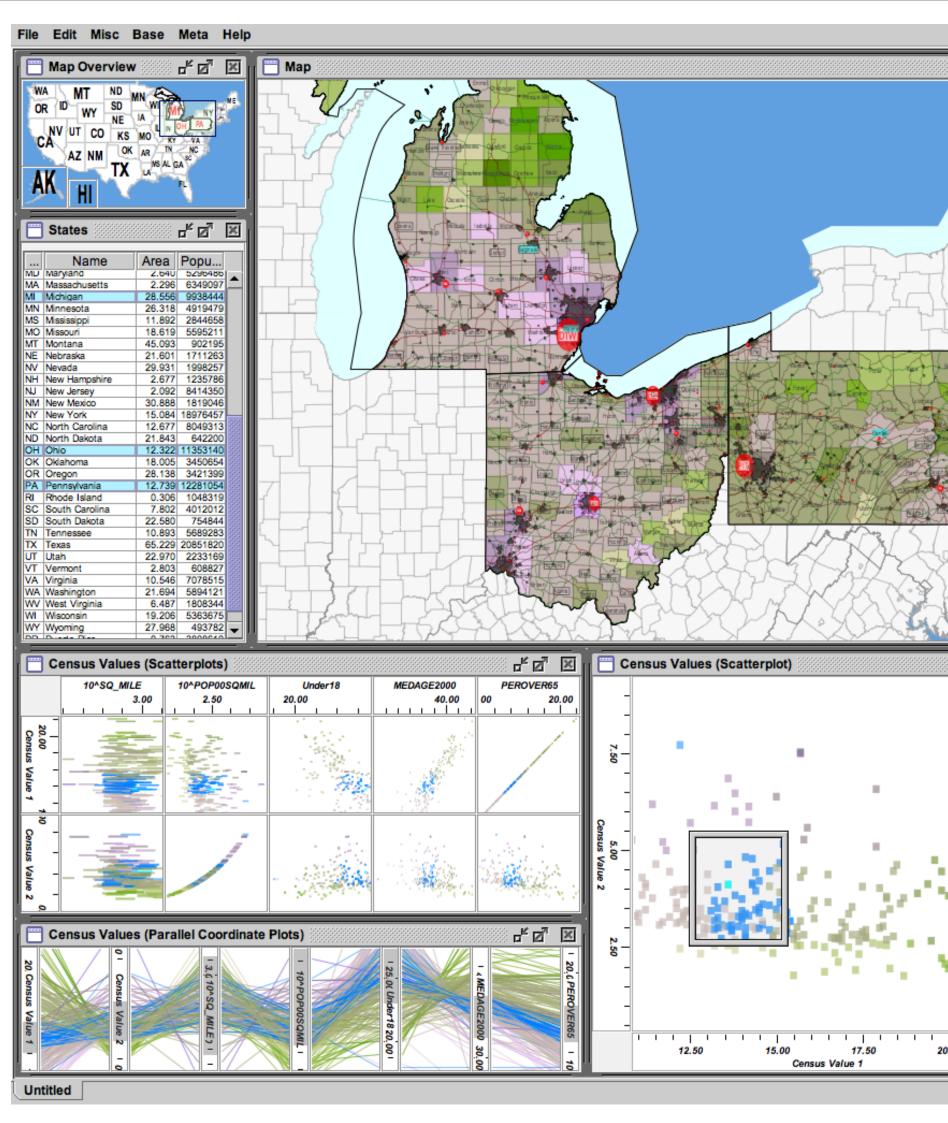








Concatenation



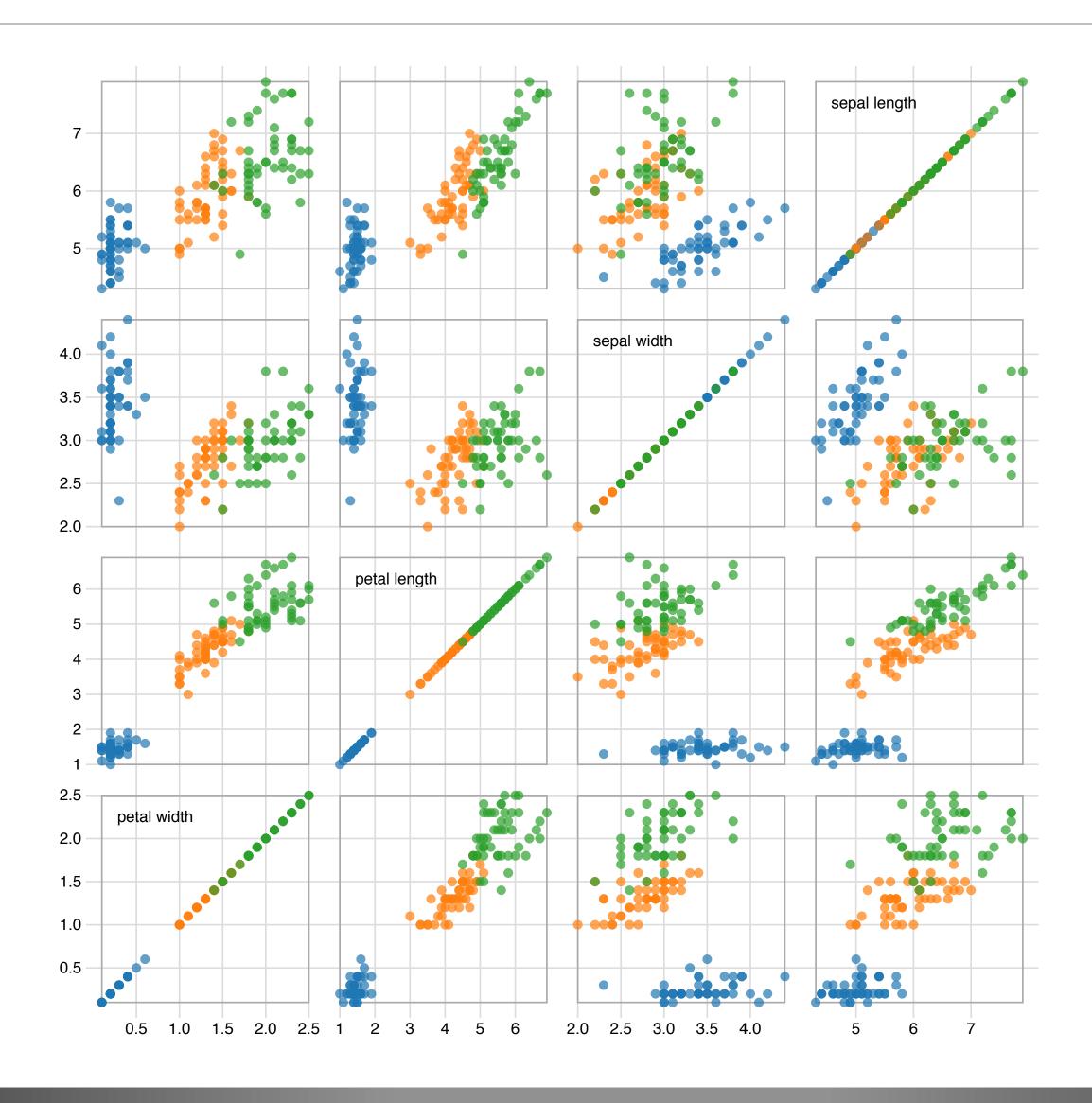
								_								_
	· 집 🛛 🛛	Counties				80 6	Ø	×	Citie	s (333)				8888 G	٢ø	×
		Name		Area	Popul	Ce	Ce		Na	me			Cou		Pop.	
1000	5 1	Montmorency Muskegon	MI MI	0.167	10315 170200	23.90 12.90	02.08		Allen Park Bellefonte		P	A Ce	ayne Co ntre Co	unty	2937 639	5
		Newaygo	MI	0.248	47874	12.80	02.75		Belleville Birch Run		N		ayne Co Saginaw	unty County	399 165	Constant and a second sec
le la		Oakland Oceana	MI	0.259	1194156 26873	14.00	06.08 02.66		Centre Hall Chesaning		P	A Ce	ntre Co		107	9
P. S.		Ogernaw Ontonagon	MI MI	0.168	21645 7818		02.49		Dearborn		N	11 Wa	yne Co	unty	9777	5
13 2 TT	2 1	Osceola Oscoda	MI MI	0.167	23197 9418	14.20	02.53		Dearborn He Detroit	lights		1I Wa	syne Co syne Co	unty	5826 95127	
HI	171	Otsego	MI	0.155	23301	13.70	02.59		Ecorse Flat Rock				ayne Co ayne Co		1122	
E. MIL	hi	Ottawa Presque Isle	MI MI	0.164	238314	10.10 22.30	04.53	7773	Frankenmuti Garden City	-	N	1I S		County	483 3004	- Callelle
17 m 7 ?		Roscommon Saginaw	MI	0.170	25469 210039		02.64		Gibraltar		N	11 Wa	yne Co	unty	426	4 88
L yhr		Saint Clair Saint Joseph	MI	0.207	164235 62422	12.20	03.88		Grosse Poin Grosse Poin				ayne Co ayne Co		567 976	
$H \square$	276	Sanilac	MI	0.278	44547	15.40	02.61		Grosse Poin Grosse Poin			11 Wa 11 Wa	ayne Co ayne Co	unty	1244 282	
LTL	751	Schoolcraft Shiawassee	MI	0.370	8903 71687	18.60	01.66		Grosse Poin		ds N	11 Wa	iyne Co	unty	1708	0
		Tuscola Van Buren	MI MI	0.234	58266 76263	12.80	02.91 03.34		Hamtramck Harper Woo			1I Wa	syne Co syne Co	unty	2297	4
and sugar by	2 1 1	Washtenaw	MI	0.204	322895	08.10	04.62		Highland Pa Howard	rk			ayne Co ntre Co		1674	-
and the Dawn	_~~	Wayne Wexford		0.174	2061162 30484	14.00	07.61 02.71		Inkster Lincoln Park		N	1I Wa	ayne Co ayne Co	unty	3011 4000	_
S-C-SQUAT		Adams	OH	0.990	30484 27330	00.00	00.00		Livonia		N	11 Wa	iyne Co	unty	10054	5
and the second second		Allen	OH	0.112	108473 52523	14.20	04.05		Melvindale Merrill		N	1I S	ayne Co Saginaw	unty County	1073	
Contraction of the second		Ashtabula	OH	0.199	102728	14.70	03.48		Milesburg Millheim				ntre Co ntre Co		118	_
A BOARD	-3-	Athens Auglaize	OH	0.138	62223 46611	09.30	03.33 03.28		Northville				avne Co		645	
	~	Belmont Brown	OH OH	0.148	70226		03.38		Airp	orts &	Sea	plan	e Ba		ťø	×
STR.		Butler Carroll	OH OH	0.127	332807 28836	10.70	05.17			11.5 0		pium				
	V II	Champaign	OH	0.118	38890	12.60	03.09			Nam			En		ounty	
	1	Clark	OH OH	0.110	144742	14.70 09.40	04.36		M MBS In	Metropo ternatio		wa	1698	MI Wa	yne Co ginaw .	
		Clinton Columbiana	OH OH	0.112	40543		03.15		DET Detroit U Univers	City ity Park		_		MI Wa PA Cer		
		Coshoctor	OH	0.156	36655 46966		02.84		YIP Willow				3046	MI Wa	yne Co	
		Cuyahoga	OH	0.129	1393978	15.60	07.43									
Mar 3		Darke		0.165	53309	15.30	03.07	_								-
් ත් 🗵 🛛 🗂	Color Scheme	e ())))))))))))))))))))))))))))))))))))			- · • 🖽	City-0	City Di	stand	ces						ťø	×
S	equential Sequ	ential Non-Gra	y	•	П				Mi Saginaw County Mi	MI	MI	MI	PA ounty MI	MI	PA	MI Wayne County
			_	-1					N N N	Jayne Cour	Mayne Count	Mayne County	PA Centre County MI	Wayne County M	PA Centre County	ğ
					ų III.				gina	Way	Way	Way	Cent	Way	Cent	Way
				1						8 6 8	8	5		2 8		e 8
									Zilwaukee Pop. 1799 1 Wyandotte	Woodhaven Pop. 12530	Westland Pop. 86602	Wayne Pop. 19051	Unionville Pop. 313 Trenton	Pop. 19584 Taylor	38.0	301
									Pop VyB	Pop 4	Pop Pop	Wayne Pop. 19	H B F	Pop. 1 Taylor	Stat Pop	Pop
					MI	ne Count		en Park 29376		0.12	0.20	0.18	5.50 0	12 0.0	5.55	0.05
					PA	ne coun	Be	lefonte	0 00 5 5	5.60	5.80	5 77	0 10 5	54 5 6	5 0 15	5 57
					Cent	re Count		p. 6395 elleville			_	-		_		_
		_			Way	ne Count	y Pop	p. 3997	1.34 0.34	0.25	0.15	0.13	5.76 0	.31 0.2	2 5.80	0.29
	Show	🛞 🛛 🔚 Lat	oel 🕴		MI Sagi	naw Co		rdh Run p. 1653		2 1.24	1.01	1.05	6.37 1	.27 1.14	4 6.42	1.20
	✓ Counties	Co	untie	es	PA	re Count		tre Hall p. 1079		5.70	5.90	5.88	0.20 5	64 5.7	5 0.18	5.67
	Cities				MI		Ch	esaning	0 35 1 33	1.36	1.12	1.16	6.64 1	40 1.2	6.70	1.34
▝▝	Roads		Cities			Saginaw County Pop. 254 MI Dearbor						-	_	_		
			Roads			Wayne County Pop. 97775			1.37 0.11	0.19	0.22	0.21	5.49 0	18 0.12	2 5.53	0.11
	Railroads	🗌 🗌 Ra	ilroa	ds	MI Way	ne Count		Heights 58264		0.20	0.13	0.13	5.58 0	.22 0.10	5.63	0.15
	✓ Airports	🖌 🖌 Air	port	s	MI	ne Count	Dee	Detroit		0.27	0.35	0.34	5.36 0	23 0.24	5.41	0.19
							M PORT	801211								
0.00	Urban Area		tes		MI	ne Count		Ecorse 11229	1 46 0 03	0.14	0.27	0.24	5.44 0	.11 0.12	2 5.48	0.06







Repetition

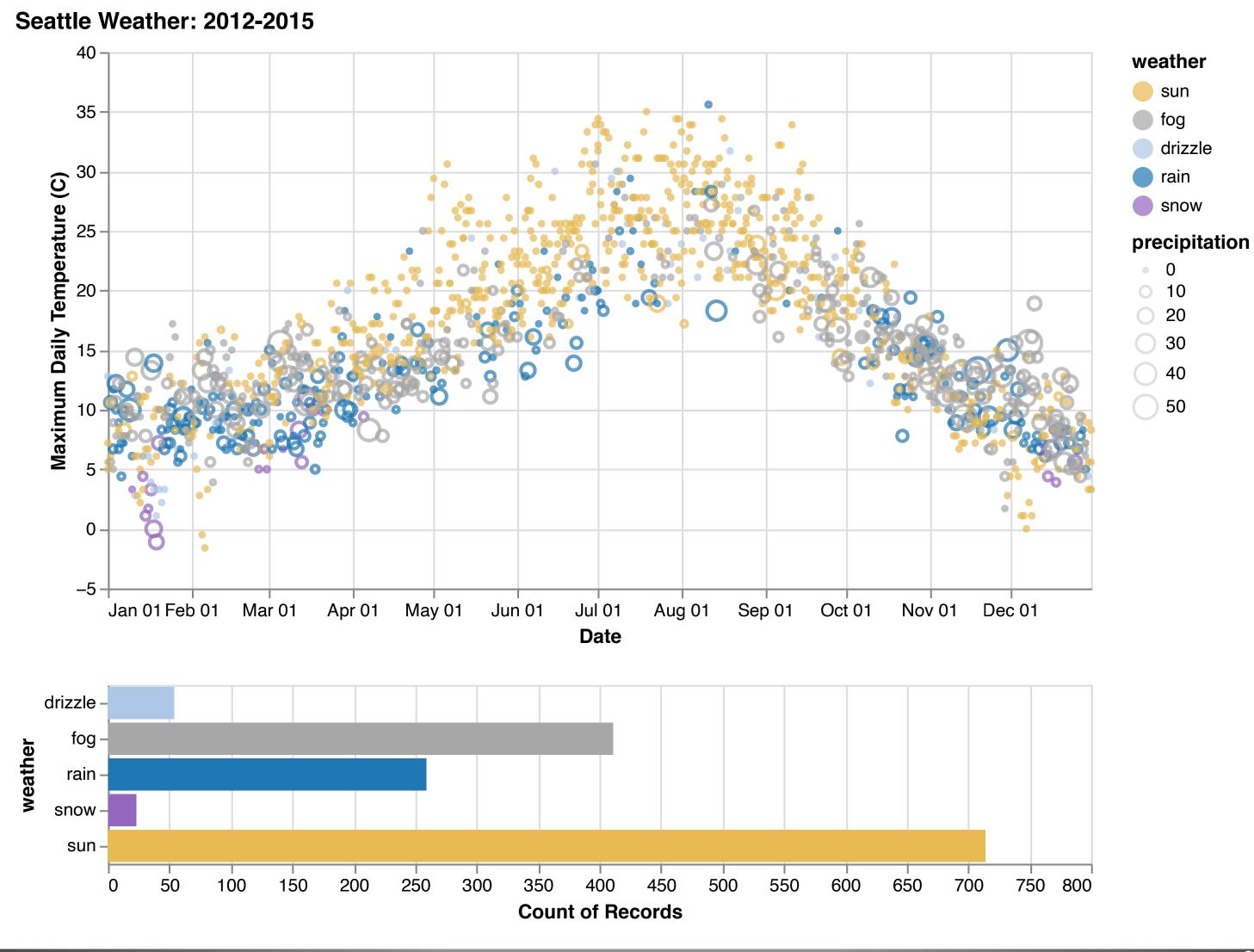








Interaction

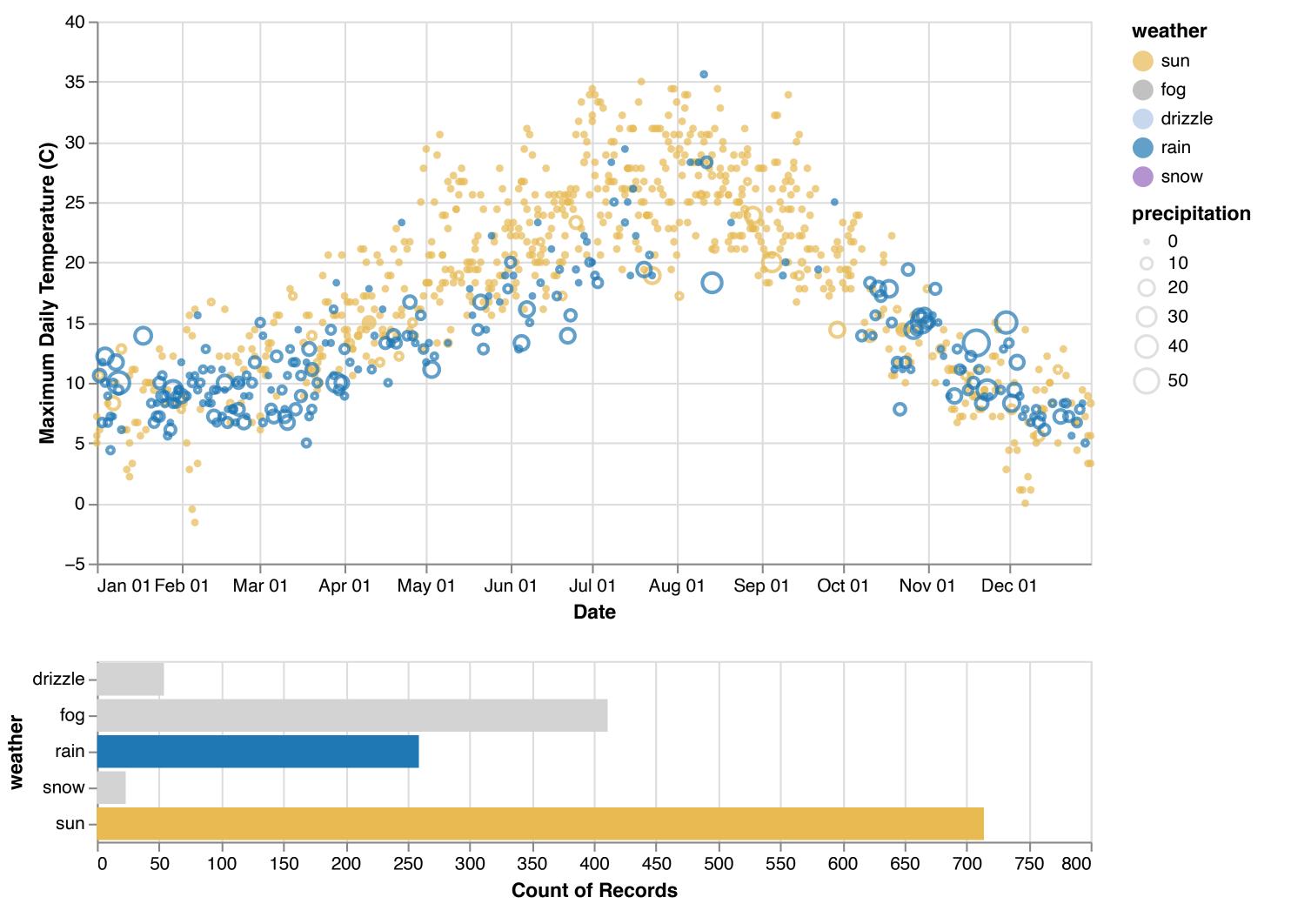






Weather Selection: Rain vs. Sun

Seattle Weather: 2012-2015

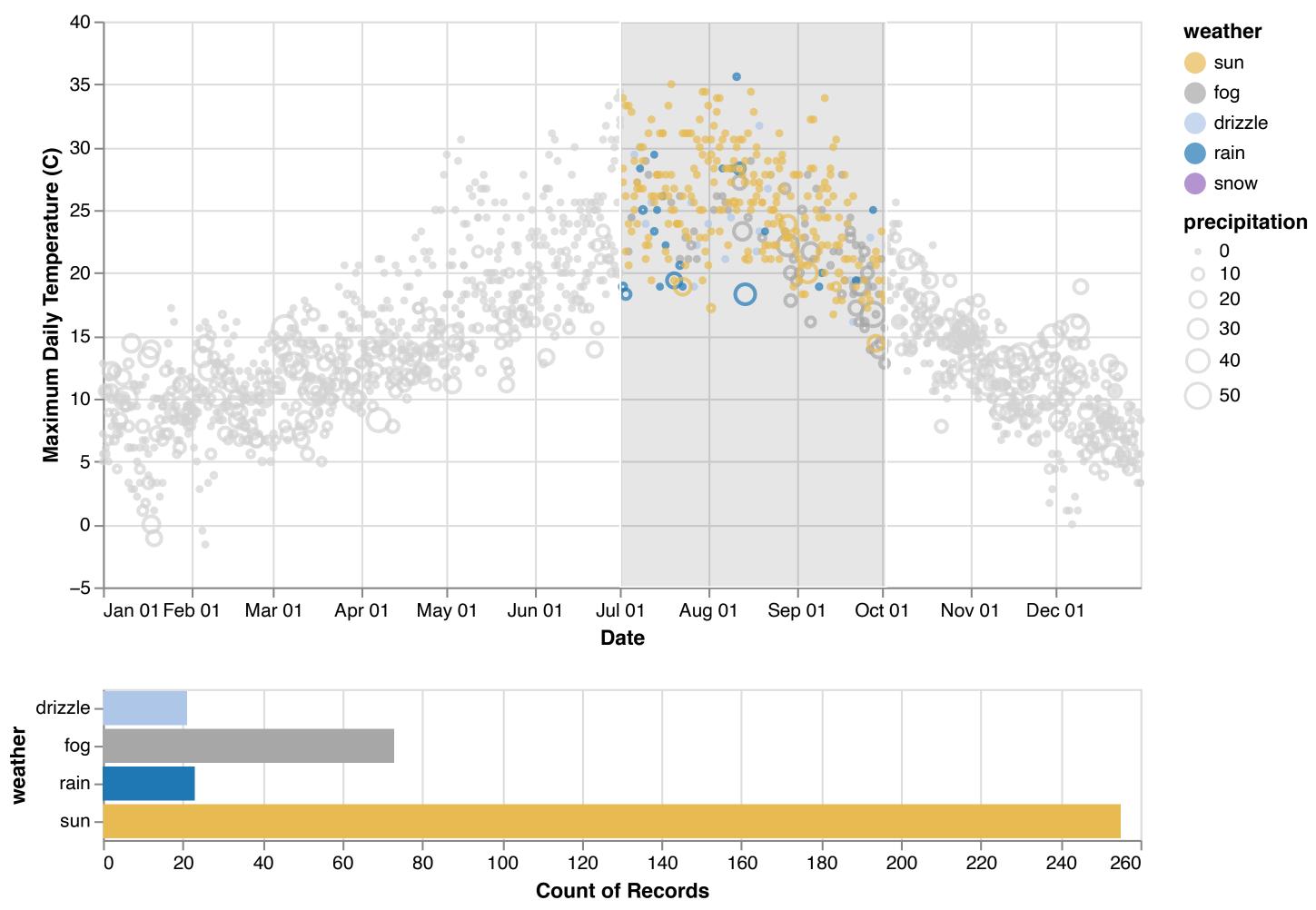






Date Selection: July-September Sun

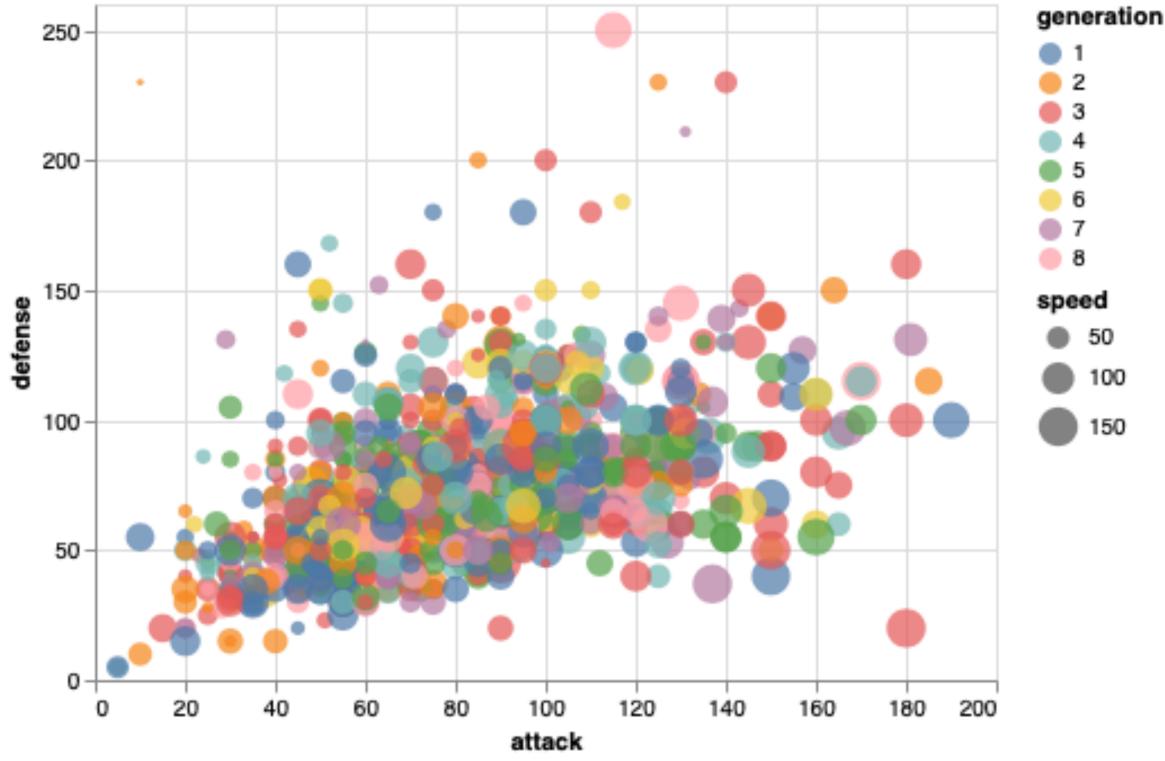
Seattle Weather: 2012-2015







<u>Assignment 8</u>



- Back to Pokémon Data
- Calculate MaxCP in pandas and find highest per generation
 - Analyze attack, defense, and speed by primary type and generation using visualizations created with matplotlib and altair











Final Exam

- Monday, April 26, 2:00-3:50pm, Online (Blackboard)
- More comprehensive than Test 2
- Expect questions from topics covered on Test 1 and 2
- Expect questions from the last three weeks of class (data, visualization, machine learning)
- Similar format





Machine Learning Intro





Tasks Machine Learning can Help With

Identifying the zip code from handwritten digits on an envelope



- Detecting fraudulent activity in credit card transactions
- Identifying topics in a set of blog posts
- Grouping customers with similar preferences

[A. Müller & S. Guido, Introduction to Machine Learning with Python, J. Steppan (MNIST image)]





Questions when building a machine learning solution

- answer that question?
- What is the best way to phrase my question(s) as a machine learning problem?
- Have I collected enough data to represent the problem I want to solve?
- What features of the data did I extract, and will these enable the right predictions?
- How will I measure success in my application?

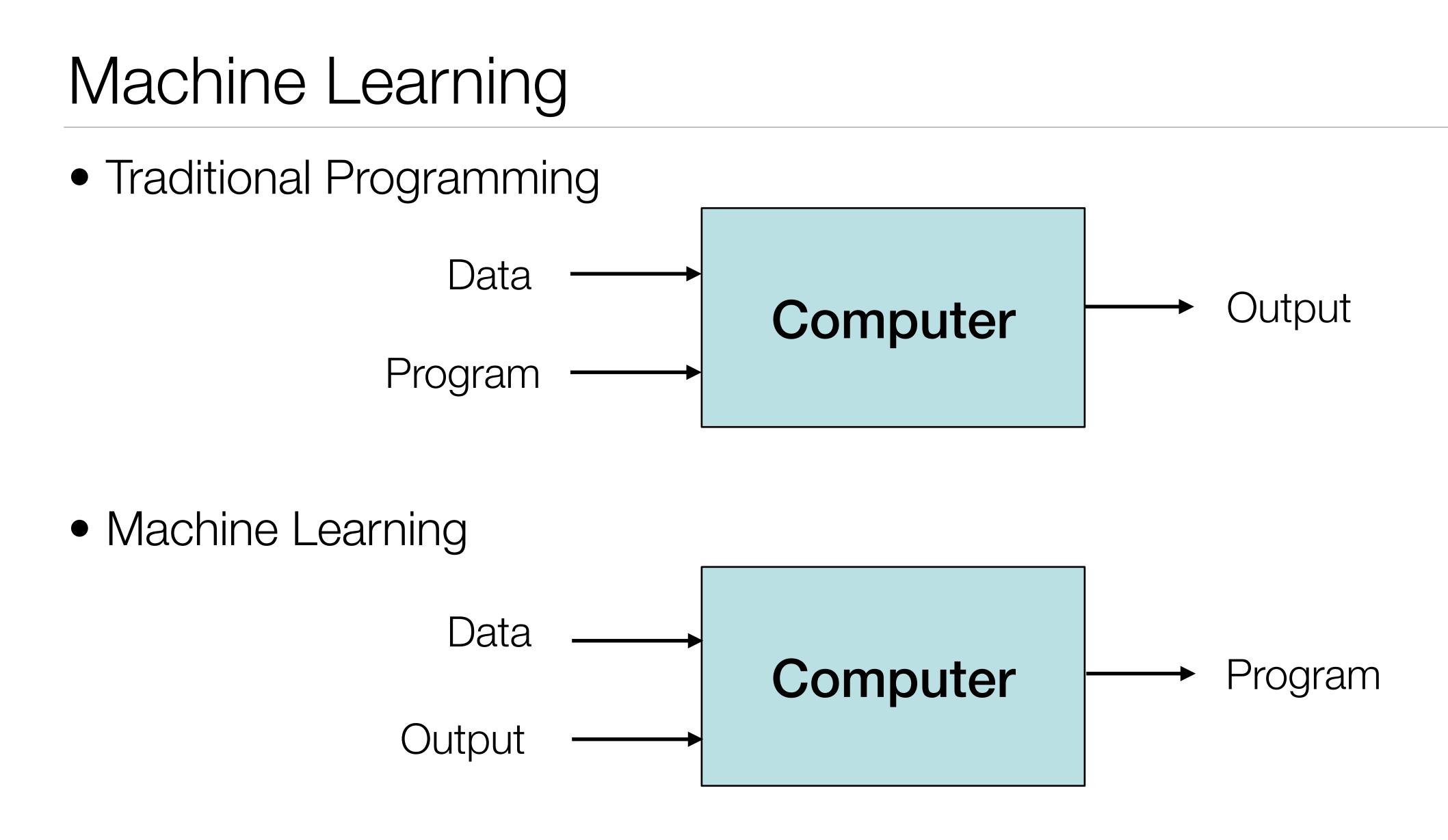
What question(s) am I trying to answer? Do I think the data collected can







19





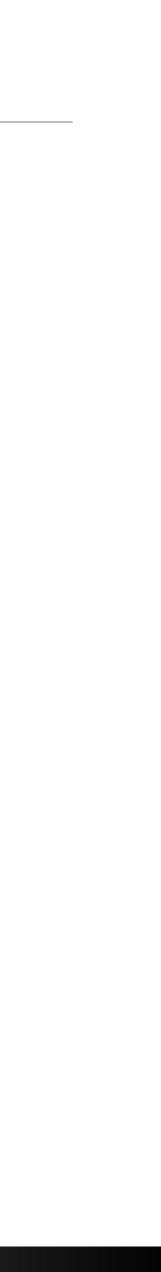




Machine Learning

- Every machine learning algorithm has three components:
 - Representation
 - Evaluation
 - Optimization









Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

D. Koop, CSCI 503, Spring 2021







22

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.











Optimization

- Combinatorial optimization
 - E.g.: Greedy search
- Convex optimization
 - E.g.: Gradient descent
- Constrained optimization
 - E.g.: Linear programming











Types of Learning

- Supervised (inductive) learning - Training data includes desired outputs
- Unsupervised learning
 - Training data does not include desired outputs
- Semi-supervised learning
 - Training data includes a few desired outputs
- Reinforcement learning
 - Rewards from sequence of actions











Areas of Machine Learning

- Supervised learning
 - Decision tree induction
 - Rule induction
 - Instance-based learning
 - Bayesian learning
 - Neural networks
 - Support vector machines
 - Model ensembles
 - Learning theory

- Unsupervised learning
 - Clustering
 - Dimensionality reduction











Supervised & Unsupervised Tasks



- Detecting fraudulent activity in credit card transactions (supervised)
- Identifying topics in a set of blog posts (**unsupervised**)
- Grouping customers with similar preferences (**unsupervised**)

[A. Müller & S. Guido, Introduction to Machine Learning with Python, J. Steppan (MNIST image)]

D. Koop, CSCI 503, Spring 2021

• Identifying the zip code from handwritten digits on an envelope (supervised)



Northern Illinois University

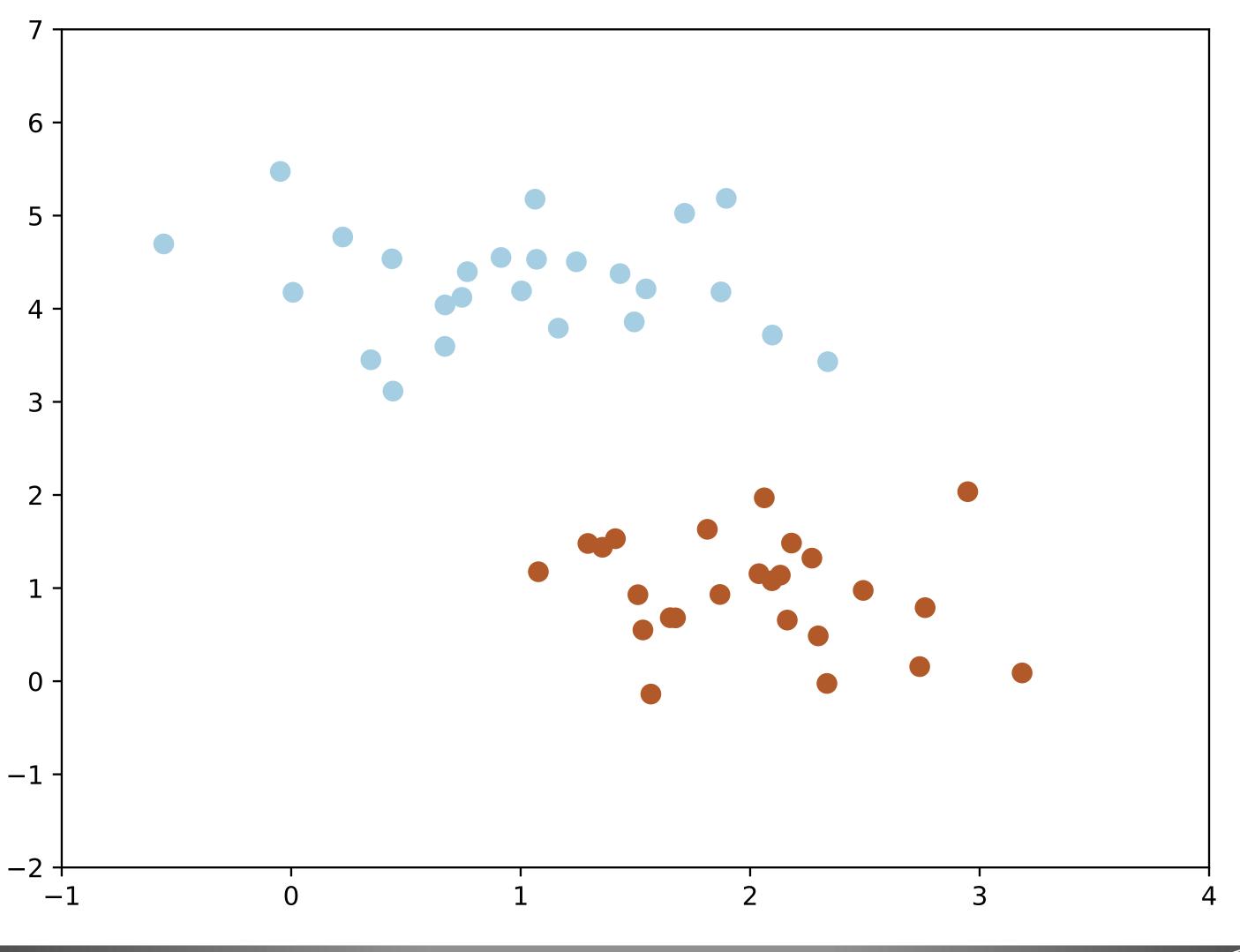








Supervised Learning





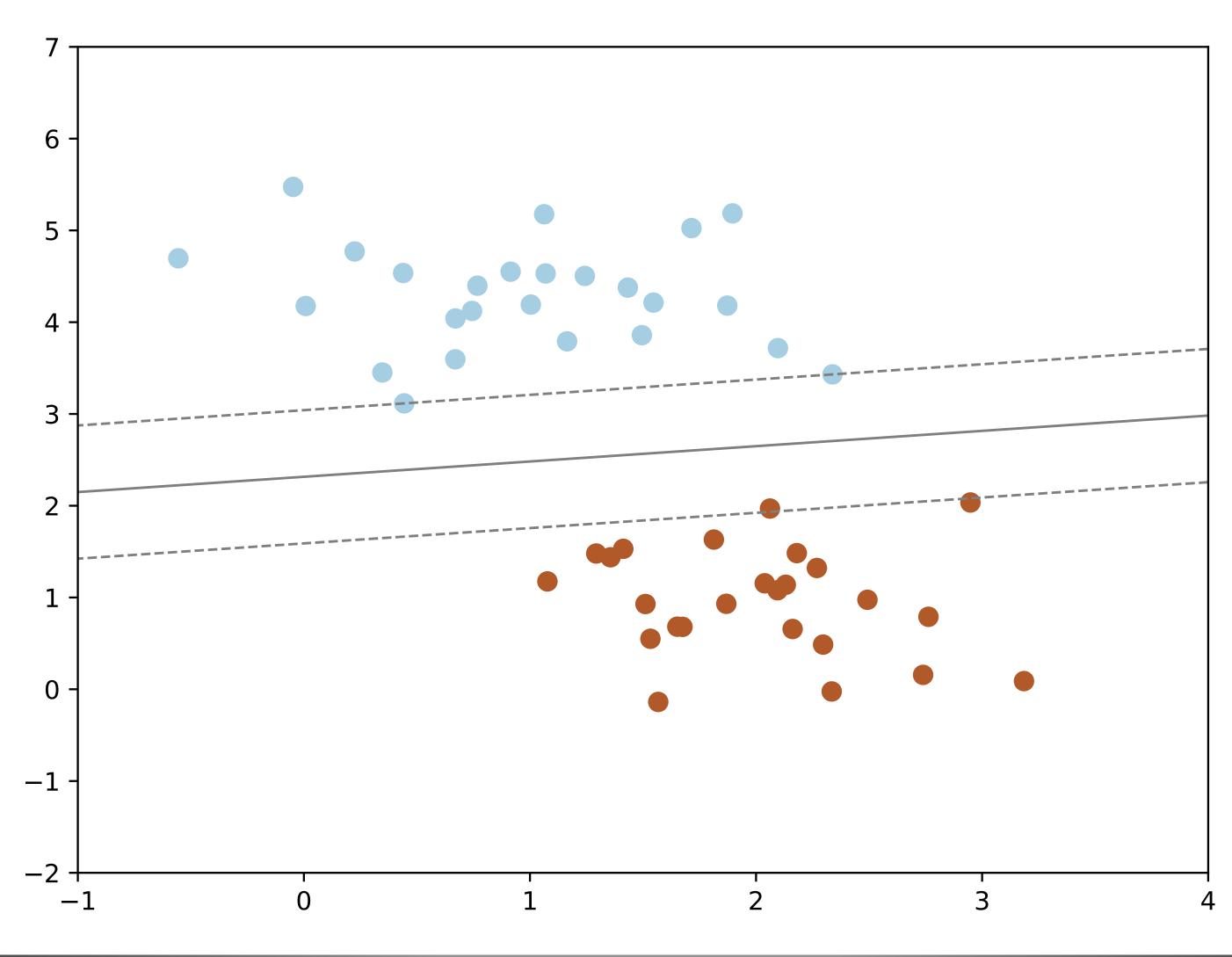








Supervised Learning: Learned Algorithm (Fit)





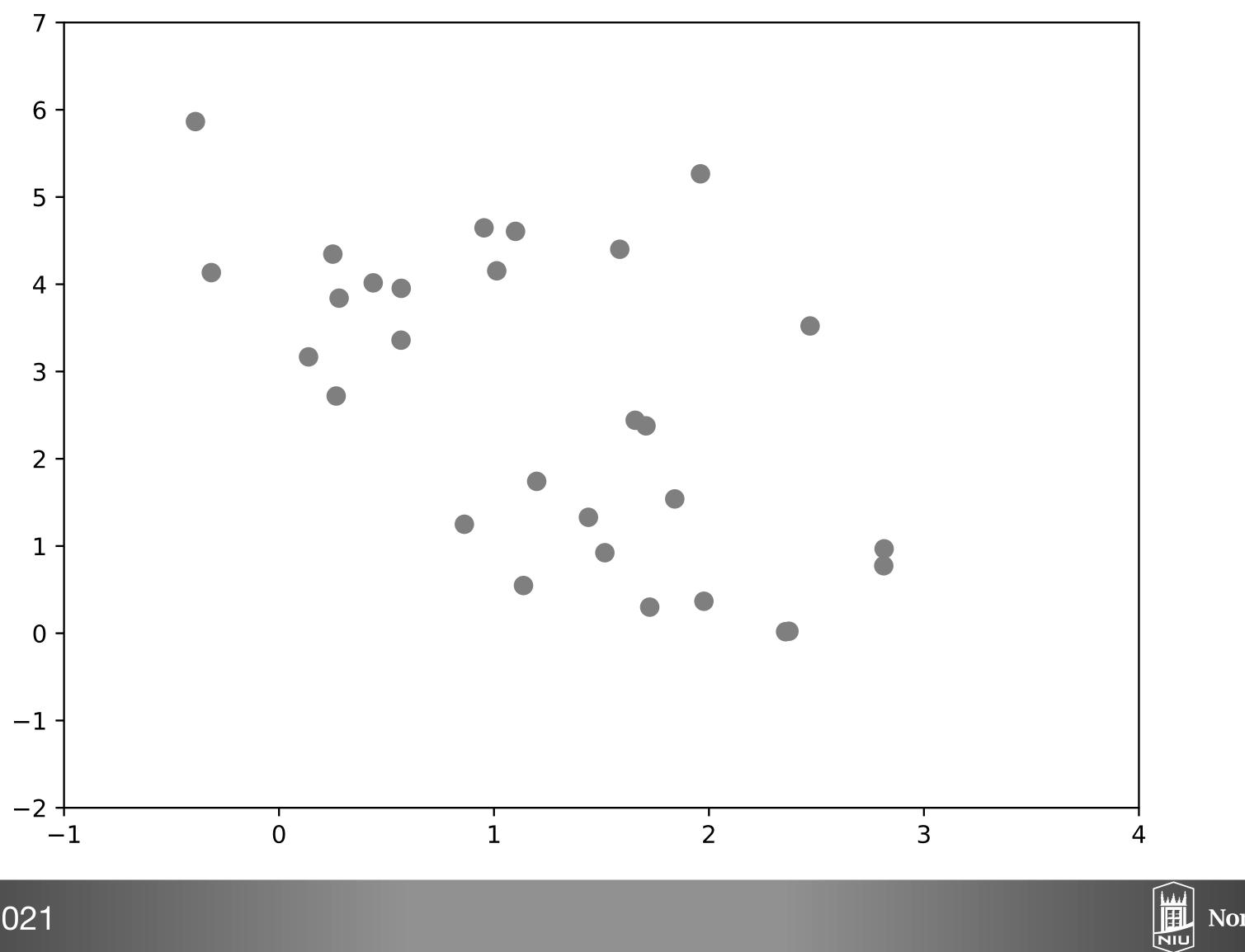








Supervised Learning: Prediction



D. Koop, CSCI 503, Spring 2021

[J. VanderPlas]

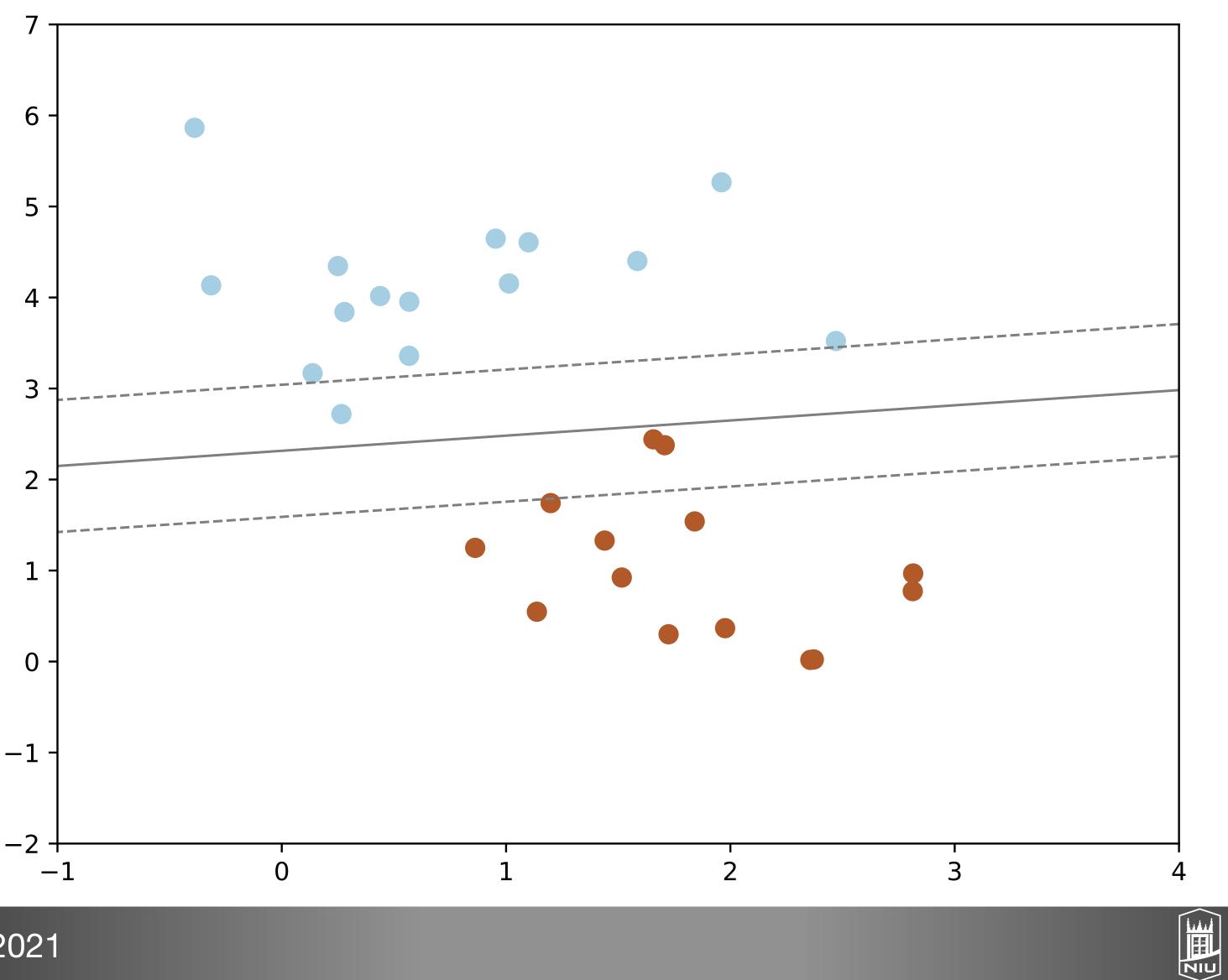
Northern Illinois University







Supervised Learning: Prediction





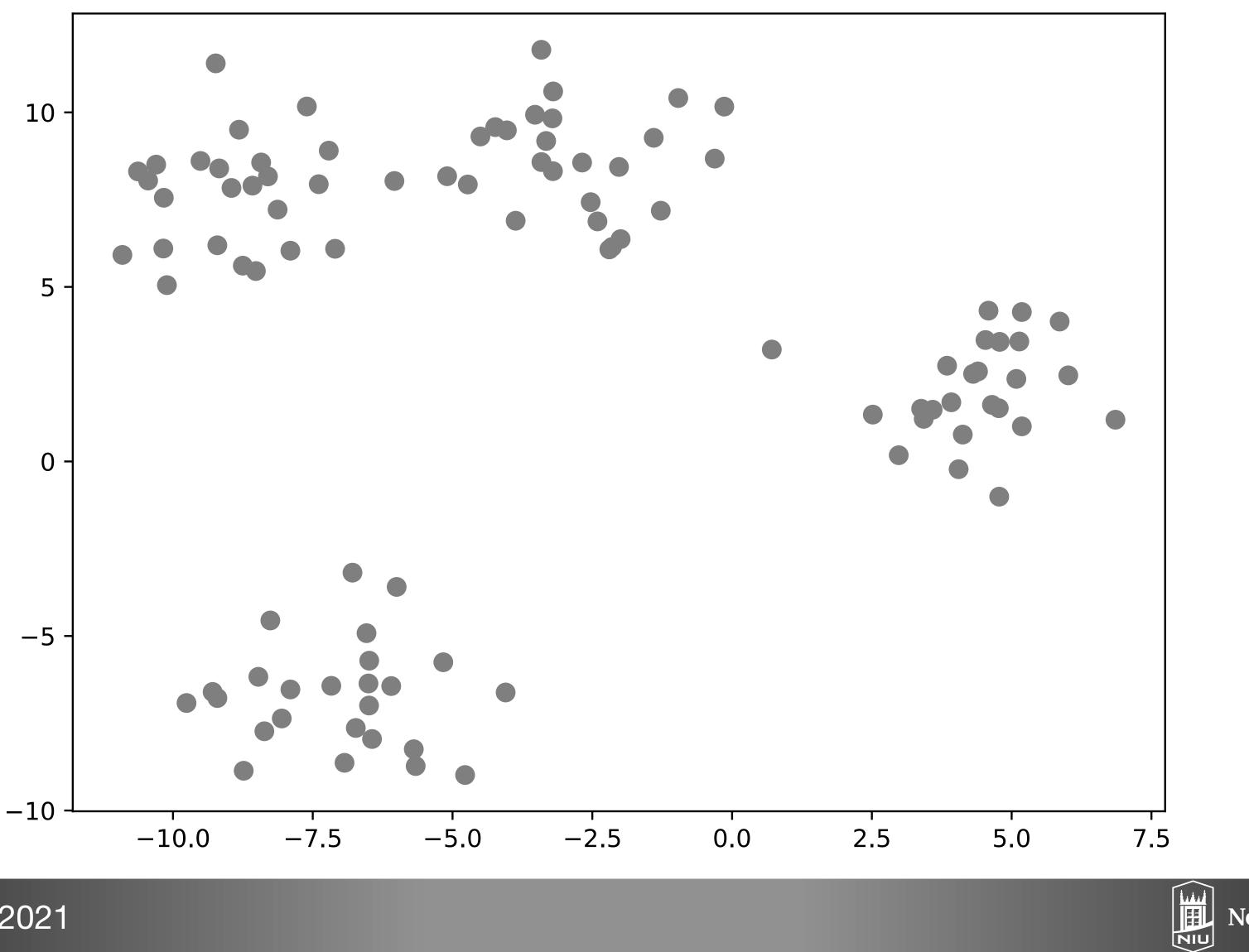








Unsupervised Learning: Input



D. Koop, CSCI 503, Spring 2021



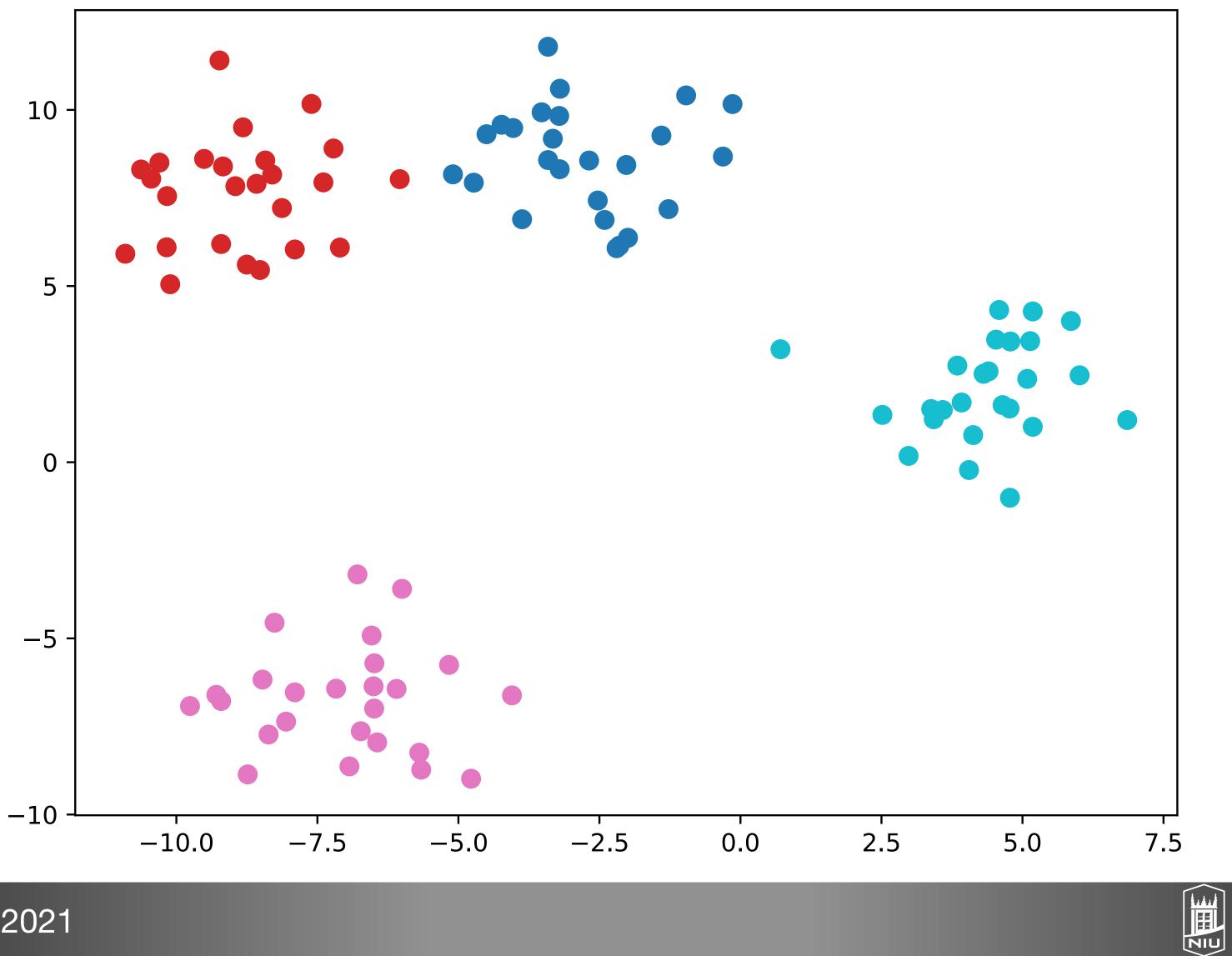
Northern Illinois University







Unsupervised Learning: Output



D. Koop, CSCI 503, Spring 2021

[J. VanderPlas]









Scikit-Learn

- Started as a Google Summer of Code project! (D. Cournapeau, 2007)
- Rewritten by scientists at INRIA (France) in 2010
- Written in Python using numpy, some optimizations using C (cython)
- The "gold standard" for machine learning in python









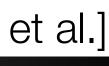


scikit-learn Principles

- Consistency: all objects share consistent, documented interface
- Inspection: parameters and parameter values determined by learning algorithms are stored and exposed as public attributes
- Non-proliferation of classes: only learning algs are classes, not datasets or parameters; easier to combine with other libraries
- Composition: create and reuse building blocks
- Sensible defaults: user-defined parameters should have meaningful defaults





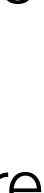




scikit-learn entities

- Data: numpy matrices (also pandas series, data frames), process batches • Estimators: all supervised & unsupervised algs implement common interface
- - estimator initialization does not do learning, only attaches parameters
 - fit does the learning, learned parameters exposed with trailing underscore
- Predictor: extends estimator with predict method
 - also provides score method to return value indicating prediction quality
- Transformer: help modify or filter data before learning
 - Preprocessing, feature selection, feature extraction, and dimensionality reduction vis transform method
 - Can combine fit and transform via fit transform









Penguin Example







Deep Learning

- neurons work together
- Hierarchical with multiple layers
- Usually takes advantage of GPUs
- Frameworks:
 - pytorch
 - TensorFlow
 - keras
 - theano

D. Koop, CSCI 503, Spring 2021

Deep learning is tied to neural networks, attempting to mimic how human





