



Data Science With Python

Mosky

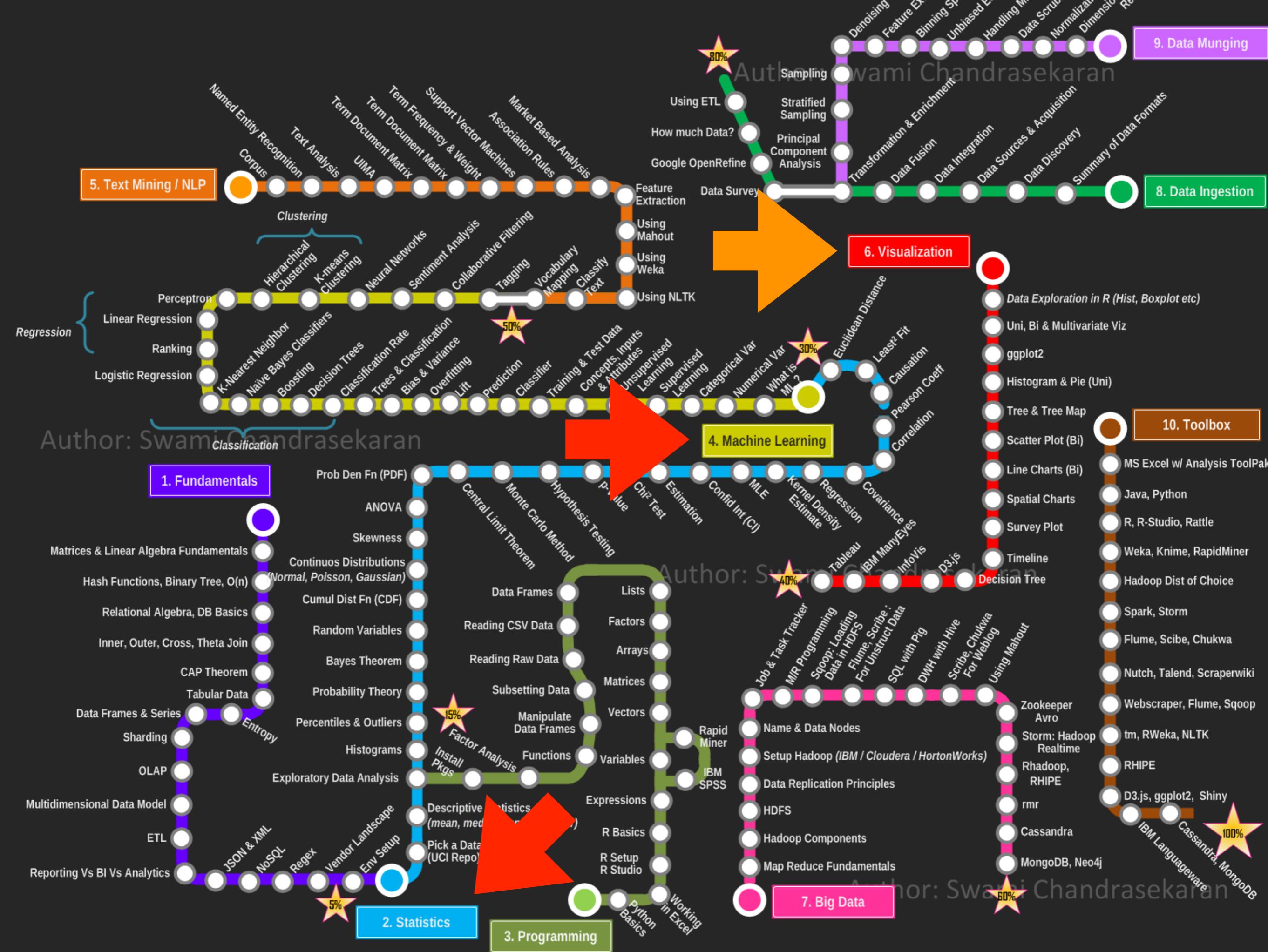
Data Science

- = Extract knowledge or insights from data.
- Data science includes:
 - Visualization
 - Statistics
 - Machine learning
 - Deep learning
 - Big data
 - And related methods
- ≈ Data mining

Data Science

- = Extract knowledge or insights from data.
- Data science includes:
 - Visualization
 - Statistics
 - Machine learning
 - Deep learning
 - Big data
 - And related methods
 - ≈ Data mining

Will be introduced.

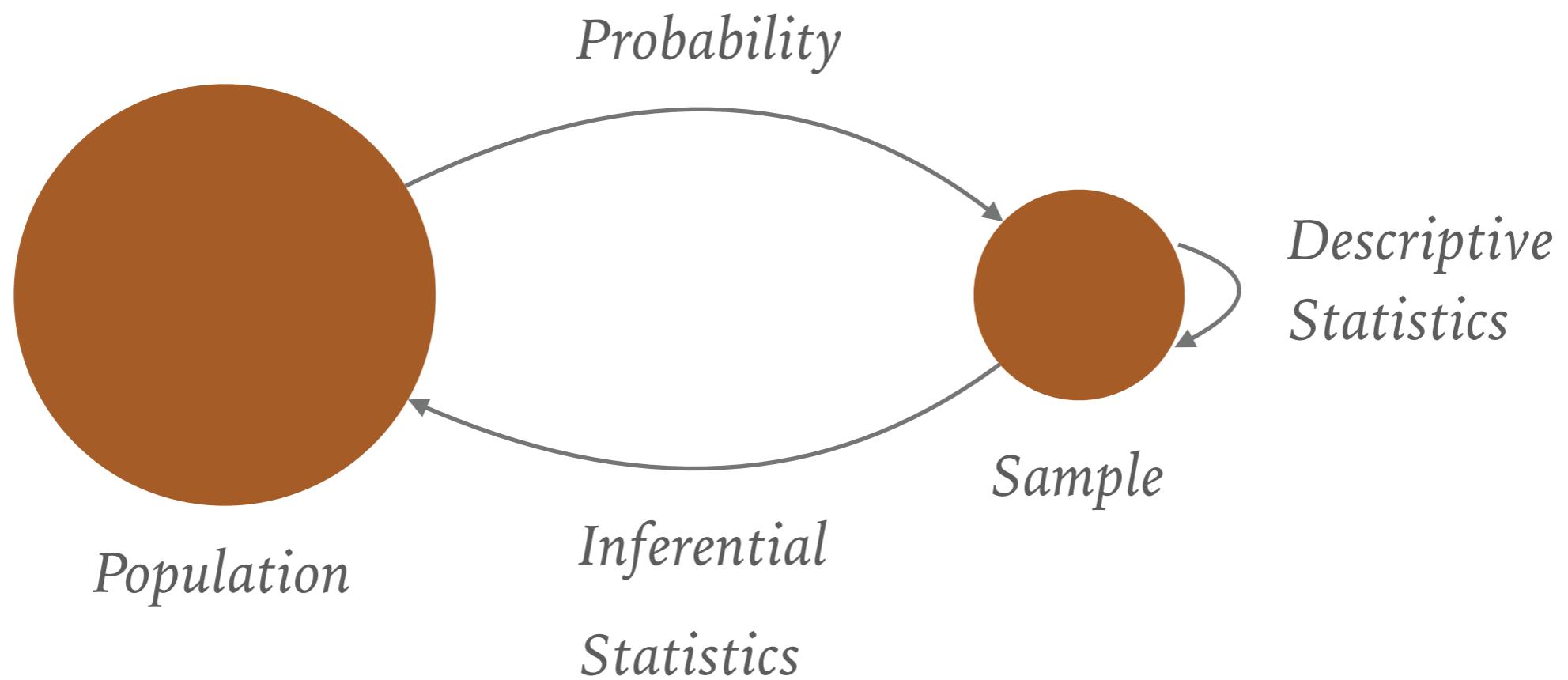


- It's kind of outdated, but still contains lot of keywords.
- [MrMimic/data-scientist-roadmap – GitHub](#)
- [Becoming a Data Scientist – Curriculum via Metromap](#)

Statistics vs. Machine Learning

- Machine learning = statistics - checking of assumptions 😊
- But does resolve more problems. 👍
- Statistics constructs more solid inferences.
- Machine learning constructs more interesting predictions.

Probability, Descriptive Statistics, and Inferential Statistics



Machine Learning vs. Deep Learning

- Deep learning is the most renowned part of machine learning.
 - A.k.a. the “AI”.
- Deep learning uses artificial neural networks (NNs).
- Which are especially good at:
 - Computer vision (**CV**) 
 - Natural language processing (**NLP**) 
 - Machine translation
 - Speech recognition
 - Too costly to simple problems.

Big Data

- The “size” is constantly moving.
 - As of 2012, ranges from 10ⁿ TB to n PB, which is 100x.
- Has high-3Vs:
 - Volume, amount of data.
 - Velocity, speed of data in and out.
 - Variety, range of data types and sources.
- A practical definition:
 - A single computer can't process in a reasonable time.
 - Distributed computing is a big deal.

Today.

- “Models” are the math models.
- “Statistical models”, emphasize inferences.
- “Machine learning models”, emphasize predictions.
- “Deep learning” and “big data” are gigantic subfields.
 - We won't introduce.
 - But the learning resources are listed at the end.



Mosky

- Python Charmer at Pinkoi.
- Has spoken at
 - PyCons in TW, MY, KR, JP, SG, HK, COSCUPs, and TEDx, etc.
- Countless hours on teaching Python.
- Own the Python packages:
 - ZIPCodeTW, MoSQL, Clime, etc.
- <http://mosky.tw/>

The Outline

- “Data”
- The Analysis Steps
- Visualization
- Preprocessing
- Dimensionality Reduction
- Statistical Models
- Machine Learning Models
- Keep Learning

The Packages

- \$ pip3 install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn
- Or
- > conda install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn

Common Jupyter Notebook Shortcuts

Esc	Edit mode → command mode.
Ctrl-Enter	Run the cell.
B	Insert cell below.
D, D	Delete the current cell.
M	To Markdown cell.
Cmd-/	Comment the code.
H	Show keyboard shortcuts.
P	Open the command palette.

Checkpoint: The Packages

- Open *00_preface_the_packages.ipynb* up.
- Run it.
- The notebooks are available on [https://github.com/moskytw/
data-science-with-python](https://github.com/moskytw/data-science-with-python).

"Data"

“Data”

- = Variables
- = Dimensions
- = Label + Features

Data in Different Types

	Nominal	{male, female}
Discrete	Ordinal Ranked	↑ & can be ordered. {great > good > fair}
	Interval	↑ & distance is meaningful. temperatures
Continuous	Ratio	↑ & 0 is meaningful. weights

Data in the X-Y Form

y

x

dependent variable

independent variable

response variable

explanatory variable

regressand

regressor

endogenous variable | endog

exogenous variable | exog

outcome

design

label

feature

- Confounding variables:
 - May affect y , but not x .
 - May lead erroneous conclusions; “garbage in, garbage out”.
 - Controlling, e.g., fix the environment.
 - Randomizing, e.g., choose by computer.
 - Matching, e.g., order by gender and then assign group.
 - Statistical control, e.g., BMI to remove height effect.
 - Double-blind, even triple-blind trials.

Get the Data

- Logs
- Existent datasets
 - The Datasets Package – StatsModels
 - Kaggle
- Experiments

The Analysis Steps

The Three Steps

1. Define Assumption
2. Validate Assumption
3. Validated Assumption?

1. Define Assumption

- Specify a *feasible* objective.
 - “Use AI to get the moon!”
- Write an *formal* assumption.
 - “The users will buy 1% items from our recommendation.” rather than “The users will love our recommendation!”
- Note the *dangerous* gaps.
 - “All the items from recommendation are free!”
 - “Correlation does not imply causation.”
- Consider the *next* actions.
 - “Release to 100% of users.” rather than “So great!”

2. Validate Assumption

- Collect *potential* data.
- List *possible* methods.
 - A plotting, median, or even mean may be good enough.
 - Selecting Statistical Tests – Bates College
 - Choosing a statistical test – HBS
 - Choosing the right estimator – Scikit-Learn
- Evaluate the metrics of methods with data.

3. Validated Assumption?

- Yes → Congrats! Report fully and take the actions! 
- No → Check:
 - The *hypotheses* of methods.
 - The *confounding variables* in data.
 - The *formality* of assumption.
 - The *feasibility* of objective.

Iterate Fast While Industry Changes Rapidly

- Resolve the small problems first.
- Resolve the **high impact/effort** problems first.
- One week to get a quick result and improve rather than one year to get the may-be-the-best result.
- Fail fast!

Checkpoint: Pick up a Method

- Think of an interesting problem.
 - E.g., the revenue is higher, but is it because of the noise?
- Pick one method from the cheatsheets.
 - Selecting Statistical Tests – Bates College
 - Choosing a statistical test – HBS
 - Choosing the right estimator – Scikit-Learn
- Remember the three analysis steps.

Visualization

Visualization

- Make Data Colorful – Plotting
 - *01_1_visualization_plotting.ipynb*
- In a Statistical Way – Descriptive Statistics
 - *01_2_visualization_descriptive_statistics.ipynb*

Checkpoint: Plot the Variables

- Star98
 - `star98_df = sm.datasets.star98.load_pandas().data`
- Fair
 - `fair_df = sm.datasets.fair.load_pandas().data`
- Howell1
 - `howell1_df = pd.read_csv('dataset_howell1.csv', sep=';')`
- Or your own datasets.
- Plot the variables that interest you.

Preprocessing

Feed the Data That Models Like

- Preprocess data for:
 - Hard requirements, e.g.,
 - corpus → vectors
 - “What kind of news will be voted down on PTT?”
 - Soft requirements (hypotheses), e.g.,
 - t-test: better when samples are normally distributed.
 - SVM: better when features range from -1 to 1.
 - More representative features, e.g., total price / units.
- Note that different models have different tastes.

Preprocessing

- The Dishes – Containers
 - *02_1_preprocessing_containers.ipynb*
- A Cooking Method – Standardization
 - *02_2_preprocessing_standardization.ipynb*
- Watch Out for Poisonous Data Points – Removing Outliers
 - *02_3_preprocessing_removing_outliers.ipynb*

Checkpoint: Preprocess the Variables

- Try to standardize and compare.
- Try to trim the outliers.

Dimensionality Reduction

The Model Sicks Up!

- Let's reduce the variables.
- Feed a subset → feature selection.
 - Feature selection using SelectFromModel – Scikit-Learn
- Feed a transformation → feature extraction.
 - PCA, FA, etc.
 - Another definition: non-numbers → numbers.

Dimensionality Reduction

- Principal Component Analysis
 - *03_1_dimensionality_reduction_principal_component_analysis.ipynb*
- Factor Analysis
 - *03_2_dimensionality_reduction_factor_analysis.ipynb*

Checkpoint: Reduce the Variables

- Try to PCA (*all variables*) → *the better components*, or FA.
- And then plot n-dimensional data onto 2-dimensional plane.

Statistical Models

Statistical Models

- Identify Boring or Interesting – Hypothesis Testings
 - *04_1_statistical_models_hypothesis_testings.ipynb*
 - “Hypothesis Testing With Python”
- Identify X-Y Relationships – Regression
 - *04_2_statistical_models_regression_anova.ipynb*

More Regression Models

- If y is not linear,
 - Logit or Poisson Regression | Generalized Linear Models, GLMs
- If y is correlated,
 - Linear Mixed Models, LMMs | Generalized Estimating Equation, GEE
- If x has multicollinearity,
 - Lasso or Ridge Regression
- If error term is heteroscedastic,
 - Weighted Least Squares, WLS | Generalized Least Squares, GLS
- If x is time series – predict x_t from x_{t-1} , not predict y from x ,
 - Autoregressive Integrated Moving Average, ARIMA

Checkpoint: Apply a Statistical Method

- Try to apply the analysis steps with a statistical method.
 1. Define Assumption
 2. Validate Assumption
 3. Validated Assumption?

Machine Learning Models

Machine Learning Models

- Apple or Orange? – Classification
 - *05_1_machine_learning_models_classification.ipynb*
- Without Labels – Clustering
 - *05_2_machine_learning_models_clustering.ipynb*
- Predict the Values – Regression
- Who Are the Best? – Model Selection
 - `sklearn.model_selection.GridSearchCV`

Confusion matrix, where $A = 00_2 = C[0, 0]$

		predicted - AC	predicted + BD
actual - AB	true -	A	false + B
	false -	C	true + D
actual + CD			

Common “rates” in confusion matrix

- precision = D / BD
- recall = D / CD = observed power = sensitivity
- sensitivity = D / CD = observed power = recall
- specificity = A / AB = observed confidence level
- false positive rate = B / AB = observed α
- false negative rate = C / CD = observed β

Ensemble Models

- Bagging
 - N independent models and then average their output.
 - e.g., the random forest models.
- Boosting
 - N sequential models, n -th learns from $(n-1)$ -th's error.
 - e.g., gradient tree boosting.

Checkpoint: Apply a Machine Learning Method

- Try to apply the analysis steps with a ML method.
 1. Define Assumption
 2. Validate Assumption
 3. Validated Assumption?

Keep Learning

Keep Learning

- Statistics
 - [Seeing Theory](#)
 - [Biological Statistics](#)
 - [scipy.stats + StatsModels](#)
 - [Research Methods](#)
- Machine Learning
 - [Scikit-learn Tutorials](#)
 - [Standford CS229](#)
 - [Hsuan-Tien Lin](#)
- Deep Learning
 - [TensorFlow | PyTorch](#)
 - [Standford CS231n](#)
 - [Standford CS224n](#)
- Big Data
 - [Dask](#)
 - [Hive](#)
 - [Spark](#)
 - [HBase](#)
 - [AWS](#)

The Facts

- ∵
- You can't learn *all* things in the data science!
- ∵
- “Let's learn to do” 
- “Let's do to learn” 

The Learning Flow

1. Ask a question.
 - “How does k-means work?”
2. Explore the references.
3. Digest into an answer.
 - Explore by the breadth-first way.
 - Write codes to visualize and simulate.
 - Make it work, make it right, finally make it fast.

Recap

- Let's do to learn, *not* learn to do.
- What is your **objective**?
- For the objective, what is your assumption?
- For the assumption, what method may validate it?
- For the method, how will you evaluate it with data?
- Q & A