

Welcome

NCI Data Science Learning Exchange



GETTING STARTED

LEARNING RESOURCES ▾

COLLABORATION

EVE

Exploratory Data Analysis (EDA) for Clinical Datasets

An Overview of Techniques and Tools

Posted on August 6, 2020



Today's Presenter



George Zaki, PhD
Bioinformatics Manager
George.Zaki@nih.gov

Strategic and Data Science Initiatives (SDSI) Team
Biomedical Informatics and Data Science (BIDS)



<https://tinyurl.com/yxn5nwk9>

Frederick National Laboratory for Cancer Research



Exploratory Data Analysis of Clinical Data using Pandas, Scikit-learn, and Seaborn

George Zaki, Ph.D.

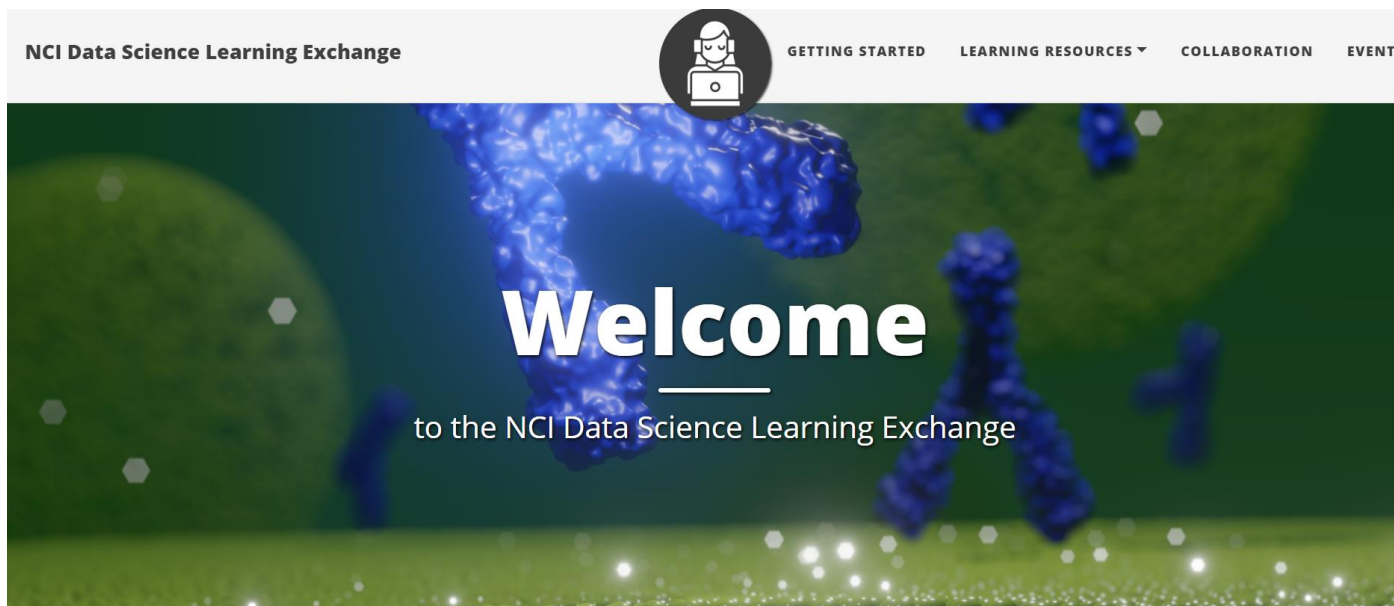
Biomedical Informatics and Data Science (BIDS)

August 6, 2020

The Frederick National Laboratory is a Federally Funded Research and Development Center operated by Leidos Biomedical Research, Inc., for the National Cancer Institute
DEPARTMENT OF HEALTH AND HUMAN SERVICES • National Institutes of Health • National Cancer Institute

NCI Data Science Learning Exchange

Frederick
National
Laboratory
for Cancer Research



Website:

<https://cbiit.github.io/p2p-datasci/>

- **Peer-to-peer community**
- **Connects NCI staff learning data science with each other**
- **See Resources & engage!**
 - **Website**
 - **Microsoft Teams**

Join the MS Team!

General Channel + 17 Topic-specific Channels!

**Frederick
National
Laboratory**
for Cancer Research

<https://bit.ly/2VjpFHn>

Intro to Data Science
Biowulf + HPC Systems
Bioinformatics
C-based languages
Command Line & Shell Scripting
Data Pipelines & Workflow Management
R
SAS
Image Analysis
Java
Machine Learning & AI
Math for ML
Python
Database

In each channel:

- **Posts** – open discussions; Q & A; recommendations; resources
- **Files**
- **Wiki**

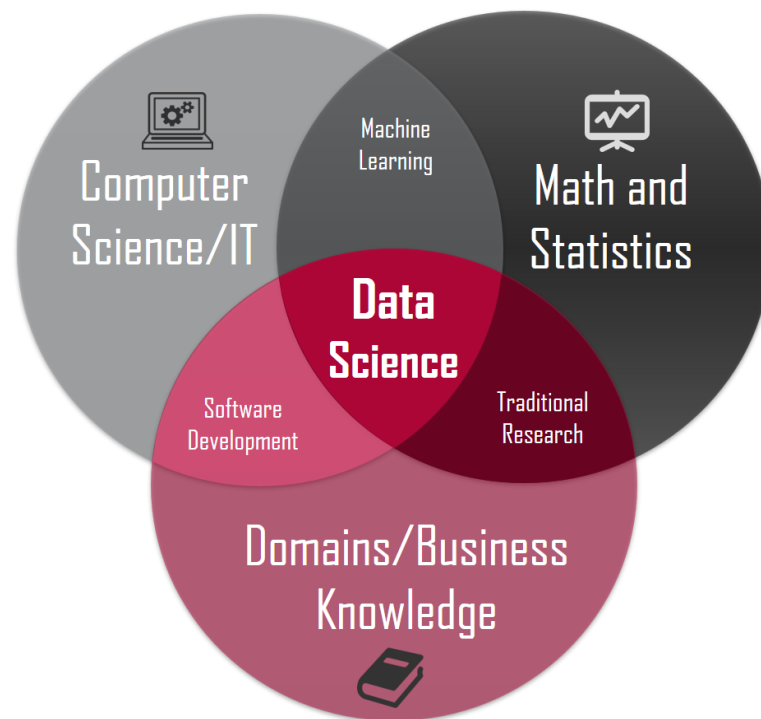
Data Science Initiative: NCI CBIIT, FNL

Leverage breakthrough advancements in scientific computing and data science to help NCI scientific staff advance basic research, understanding, and treatments in cancer.

Thanks to the team:

Lynn Borkon, Amar Khalsa,
Laurie Morrissey, Carl McCabe,
Ravi Ravichandran, Eric
Stahlberg, Andrew Weisman

george.zaki@nih.gov



Know about you

- **Have you used exploratory data analysis in your research?**
 - If yes, for what?
- **What would like to get out of this workshop?**

Exploratory Data Analysis

- What?
 - Summarize and visualize statistical characteristics of data sets.
- Why?
 - Find outliers and replicates, missing values, cleanup data
 - Understand relationships, suggest hypothesis
- How?
 - Single variable, bivariate and multivariate plots, data imputation, clustering, scaling, correlation, dimensionality reduction, etc.



‘Exploratory data analysis’ is an attitude, a state of flexibility, a willingness to look for those things that we believe are not there, as well as those we believe to be there.’

John Tukey

General steps for Exploratory Data Analysis



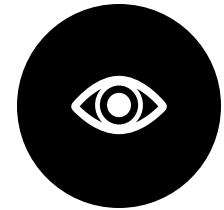
DATA
INGESTION



SUMMARY
STATISTICS



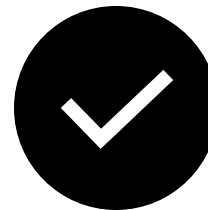
TRANSFORMATIONS



VISUALIZATION



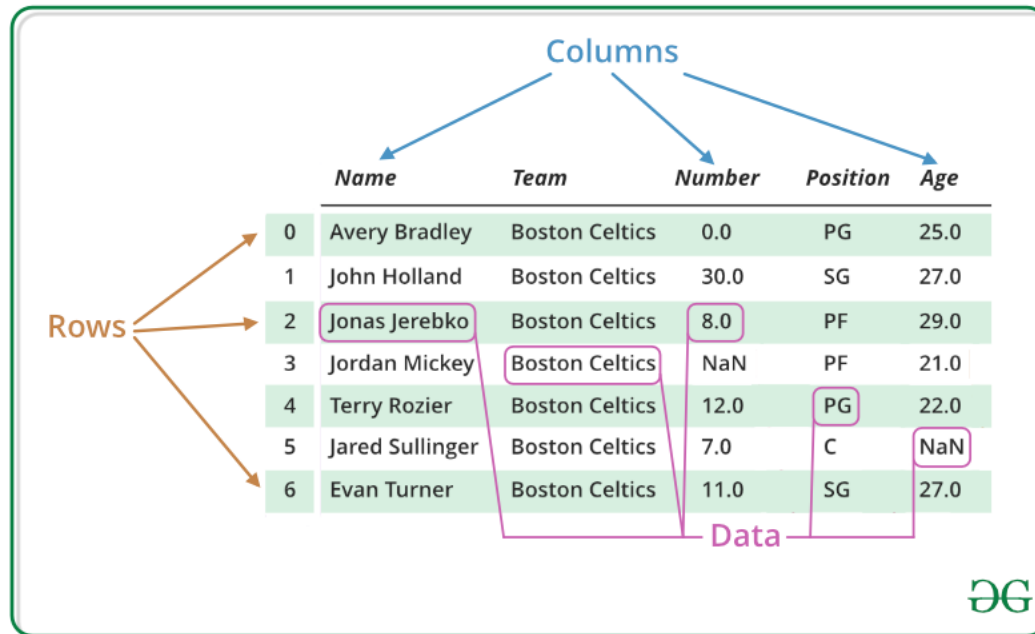
IMPUTATION



DIMENSIONALITY
REDUCTION

Data Ingestion

- Data sources: own experiments, online portal, data repositories.
- Original data might need to be processed/cleaned. Generate gene expression count, remove artifact from processing tools, etc.
- Very important to clearly document/version control what you have done
- Once the data is in a form of: Samples * features, then it can be loaded in memory as **dataframe** (e.g. Pandas)



The diagram illustrates a DataFrame structure. A table is shown with 7 rows and 6 columns. The columns are labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed from 0 to 6. The data is as follows:

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

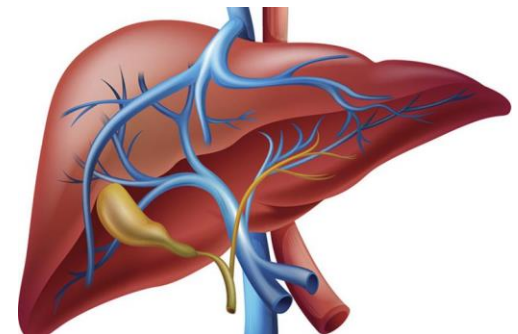
Annotations in the diagram:

- Columns:** A blue arrow points from the word 'Columns' to the column headers.
- Rows:** An orange arrow points from the word 'Rows' to the row indices (0-6).
- Data:** A pink box highlights the data cells (excluding headers and indices) for rows 2 through 6, with a pink arrow pointing to the word 'Data'.

Logo: OG

The clinical data for this workshop

- Breast Cancer Wisconsin (Diagnostic) Data Set
- Cervical cancer (Risk Factors) Data Set
- Hepatitis C Virus (HCV) for Egyptian patients Data Set
- Each one of these datasets would highlight different aspects on the application of EDA to better understand the data.
- Github examples: <https://github.com/georgezakinih/exploratory-data-analysis>



Summary Statistics

- Pandas' **head**, **describe**, **summary**, and **info**

```
1 raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1385 entries, 0 to 1384
```

```
Data columns (total 29 columns):
```

#	Column	Non-Null Count	Dtype
0	Age	1385 non-null	int64
1	Gender	1385 non-null	int64
2	BMI	1385 non-null	int64
3	Fever	1385 non-null	int64
4	Nausea/Vomting	1385 non-null	int64
5	Headache	1385 non-null	int64
6	Diarrhea	1385 non-null	int64
7	Fatigue & generalized bone ache	1385 non-null	int64
8	Jaundice	1385 non-null	int64
9	Epigastric pain	1385 non-null	int64

Categorical versus Numerical Data

- **Numerical data:**

- Can be positive integer, integer, real, have a specific domain
- Have statistics: mean, median, 25, 75 percentiles, standard deviation, minimum, maximum

- **Encoding of categorical features:**

- Usually represented as integer and unique string.
- Might need to be coded for subsequent machine learning tasks

One hot coding: `pd.get_dummies(df['Category'], prefix='Cat')`

Original column	New coded columns		
Category	Cat_A	Cat_B	Cat_C
A	1	0	0
B	0	1	0
C	0	0	1
A	1	0	0

Map, apply functions

- **df.apply**: Applies any user defined transformation, aggregation, split on a data frame or a series. Can be used row or column wise.
- **Series.map**: Map every value of a series to another values.

```
699, Uniformity of Cell Shape
699, Marginal Adhesion
699, Single Epithelial Cell Size
683, Bare Nuclei
699, Bland Chromatin
699, Normal Nucleoli
699, Mitoses
699, Class
```

```
1 def replace_NaN(x, median):
2     if np.isnan(x):
3         return median
4     else:
5         return x
```

- Calculate median: median = 1
- Replace NaN (e.g. missing) values with 1

```
1 data["Bare Nuclei"] =
2     data["Bare Nuclei"].map(lambda x: replace_NaN(x,1))
```

- When to use **df.apply** ? Here is a [good discussion](#)

Slicing and Dicing

- **Subsetting:**

- `raw_data[(raw_data.Age < 40)]`
- `raw_data[(raw_data.Age < 40)] & (raw_data.BMI < 20)]`

Returns a subset of the data frame rows that satisfy the condition

- Here is a nice [tutorial](#)

- **df.groupby:** Creates summary statistics per group in the data.

```
raw_data.groupby([ "Gender" ])[ 'Age ' ].mean()
```

Gender

1 46.404526

2 46.230088

Name: Age , dtype: float64

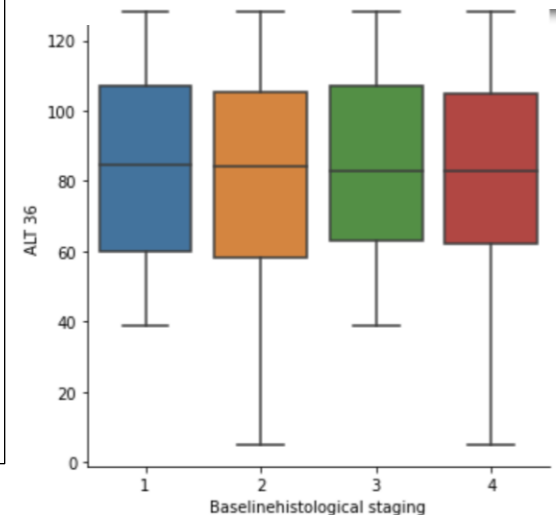
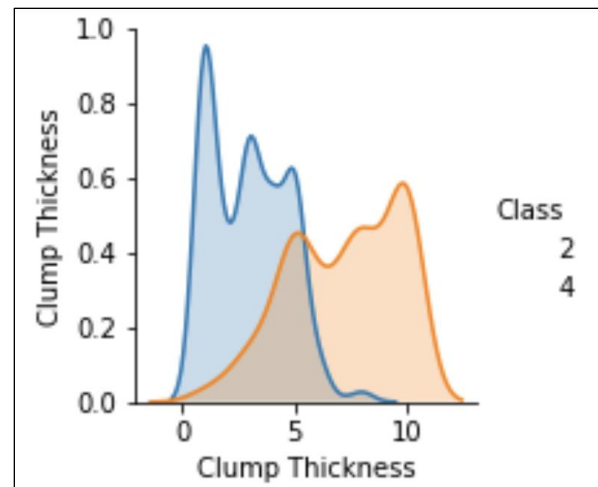
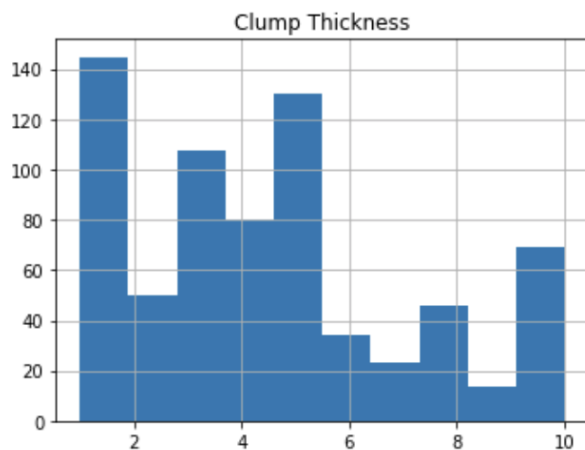
Boundary checks

- Sometimes, the values will not fit a meaningful range, the distribution might be not be normal.
- `pd.crosstab` and `df.describe` functions can help spot inconsistent data.

col_0	% observations		Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size
Smokes		count	699.000000	699.000000	699.000000	699.000000	699.000000
0.0	0.841492	mean	4.417740	3.134478	3.207439	2.806867	3.216023
1.0	0.143357	std	2.815741	3.051459	2.971913	2.855379	2.214300
?	0.015152	min	1.000000	1.000000	1.000000	1.000000	1.000000
# of unique values 3		25%	2.000000	1.000000	1.000000	1.000000	2.000000
		50%	4.000000	1.000000	1.000000	1.000000	2.000000
		75%	6.000000	5.000000	5.000000	4.000000	4.000000
		max	10.000000	10.000000	10.000000	10.000000	10.000000

Univariate plots

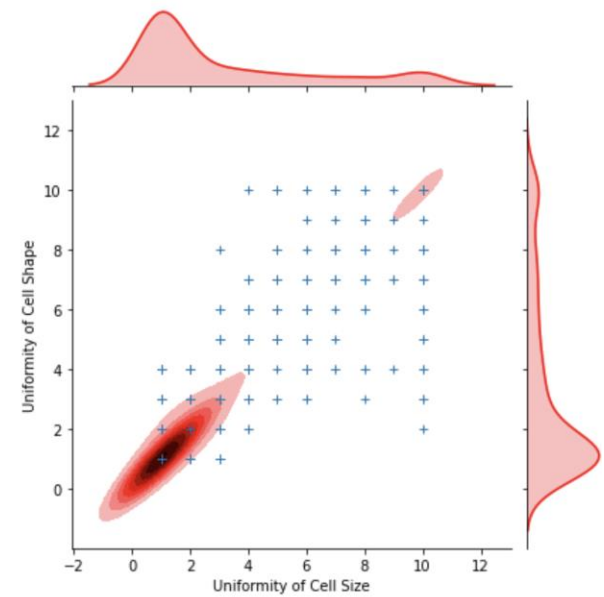
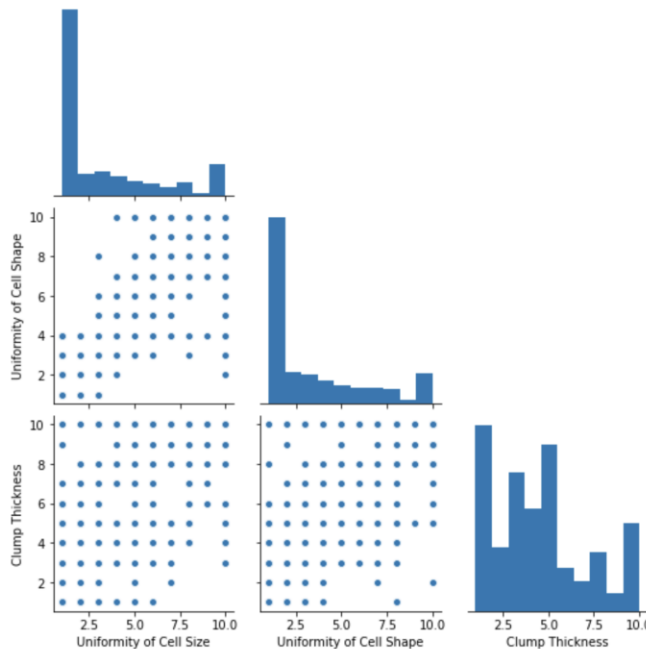
- These are methods to show the distribution of a single variable.
- Popular methods are histograms, dot plots, box plots, and kernel density plots: `df.hist`, `seaborn.pairplot`, `seaborn.catplot`



- These plots help in understanding the assumptions in a model (e.g., normal probability plot) and check the limitations where a model may not fit well the data.

Bivariate plots

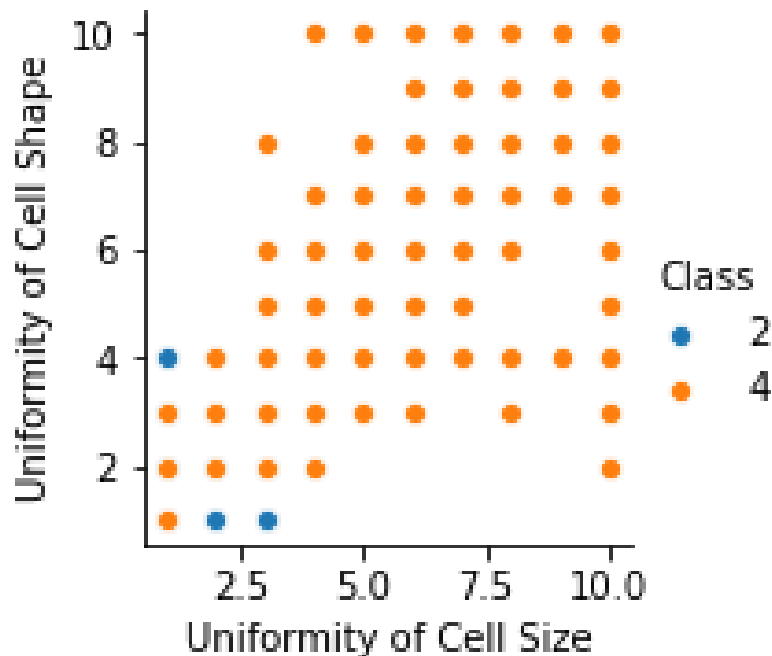
- Scatter plots can highlight the relationship between two variables and possible trends. **seaborn.jointplot**, **seaborn.pairplot**
- The components of the trend are: (a) **direction** (positive or negative), (b) **form** (linear or curvilinear), and (c) **strength** (degree of variability around the trend).



- Existence of **clusters** can also be identified in a scatter plot.

Multivariate

- Exponential number of plots: 3 variables: N^3 , 4 variables: N^4
- To limit the number of plots, use insights from the bivariate plots and select few candidates for multivariate you will investigate.
- In Seaborn, we can use 2D plots + the semantics of **hue**, **size**, and **style** to add up to three more variables.



Note that in this scatter plot, the dots with same values are **overlapping**. The data is not that imbalanced.

Missing Data Imputation

- **Missing features is a typical phenomena in clinical data.**
- **Techniques to handle missing features are:**
 - Detect and quantify, find systematic bias in missing data
 - Can I predict that data is missed based on other features in the sample?
 - Remove this feature from all the samples
 - Might miss important signal
 - Remove samples with missing features
 - Might introduce bias the samples left (e.g. a specific medical test can be done in sever cases)
 - Keep the feature, but impute its value when missing

Sklearn's **SimpleImputer**, **IterativeImputer**,
MissingIndicator, **KNNImputer**



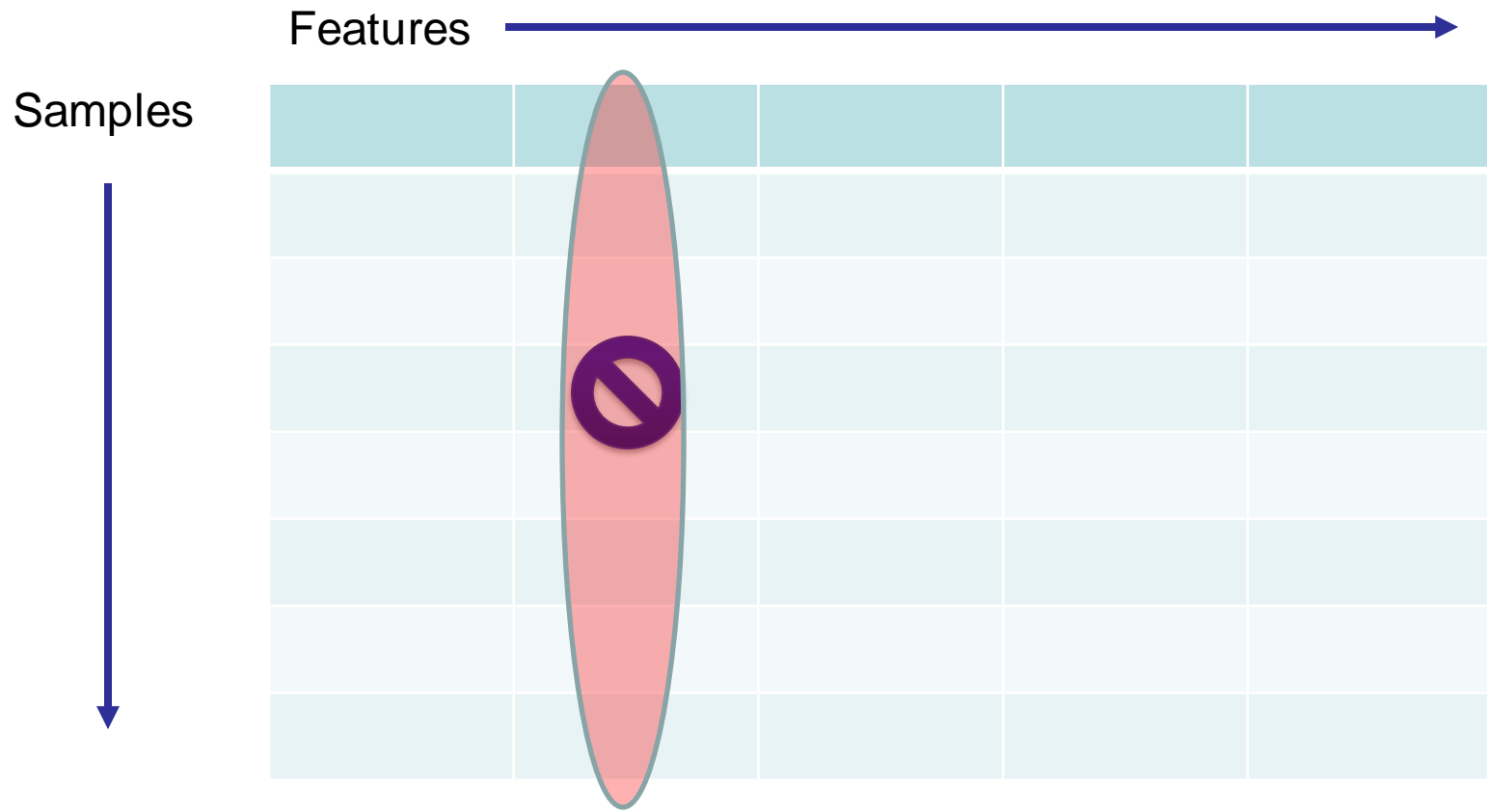
Missing Data Imputation

- **Keep the feature, but impute its value when missing:**
 - Numerical: median, mean, most frequent, constant
 - Categorical: most frequent, create a "missing" category
 - Add extra column indicating when the variable has been imputed
 - Be careful that a ML algorithm can learn this information
 - Impute the value from other features in the same sample
 - Impute the value randomly from the closest set of samples

Sklearn's **SimpleImputer**, **IterativeImputer**,
MissingIndicator, **KNNImputer**

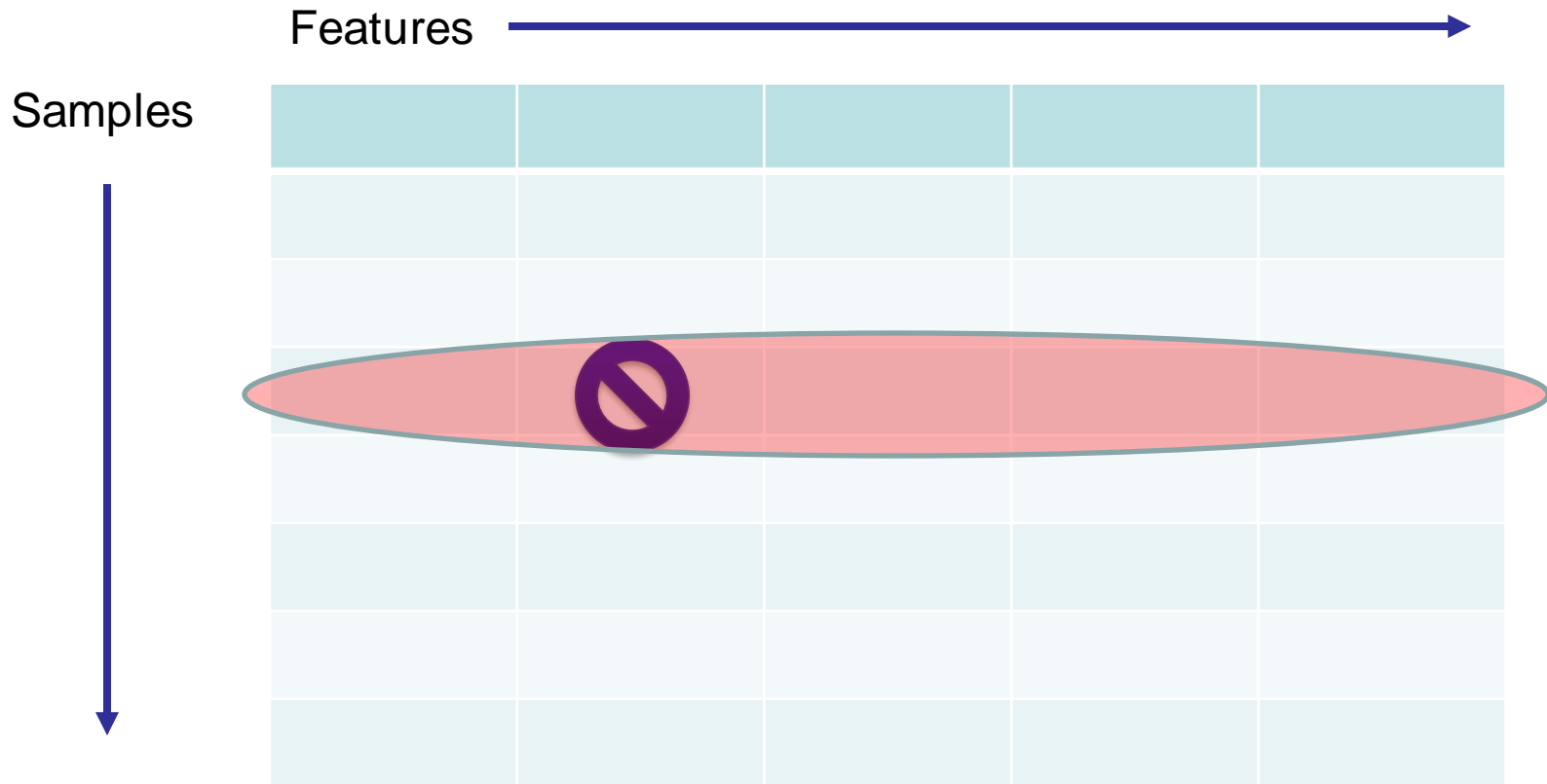
- **Warning!** Imputation might introduce correlation in some samples. Make sure you understand/quantify the implication of the imputation technique you choose.

Data Imputation



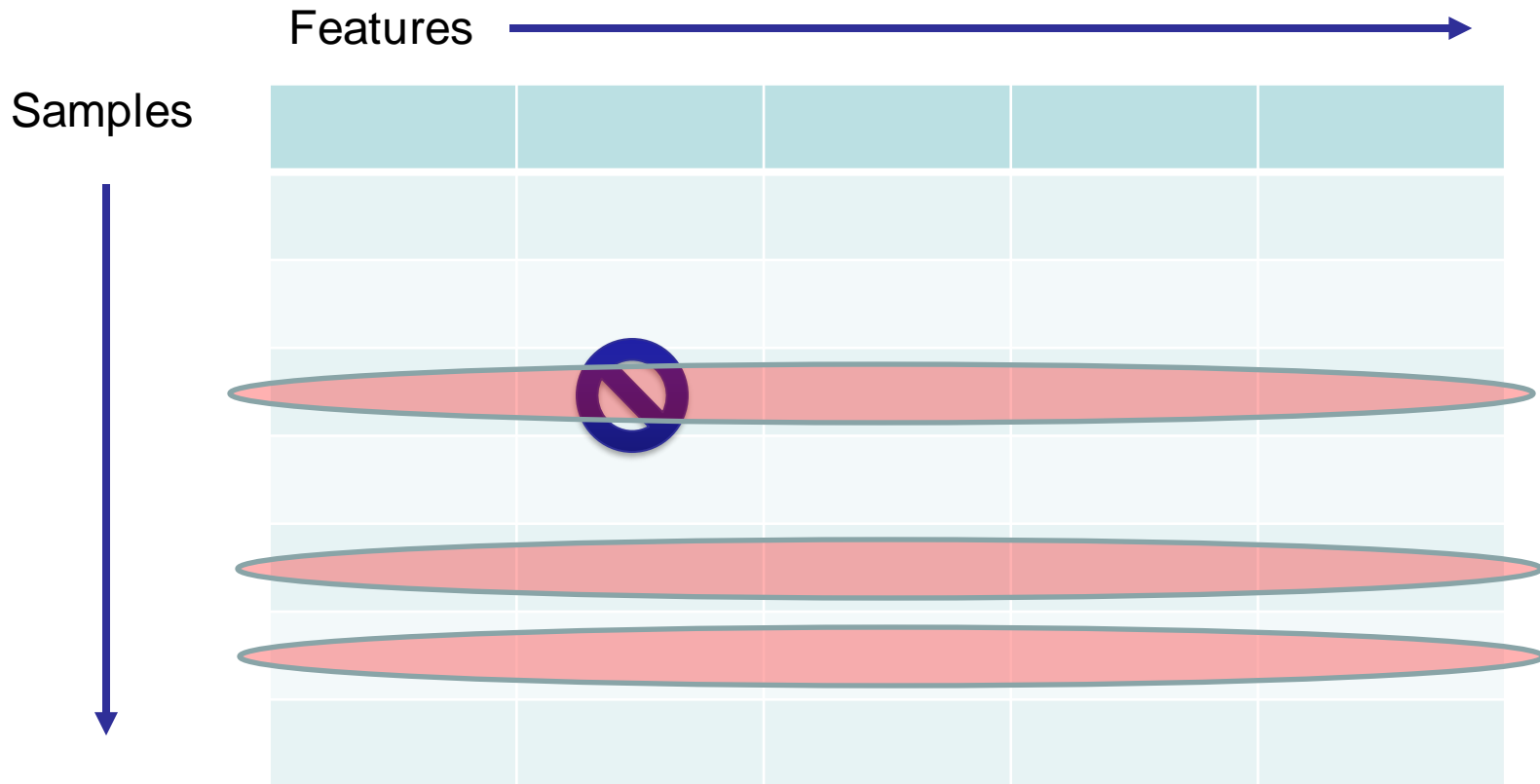
Impute using statistics of the features from other samples: mean, median

Data Imputation



Infer the missing values from other **features** in the sample

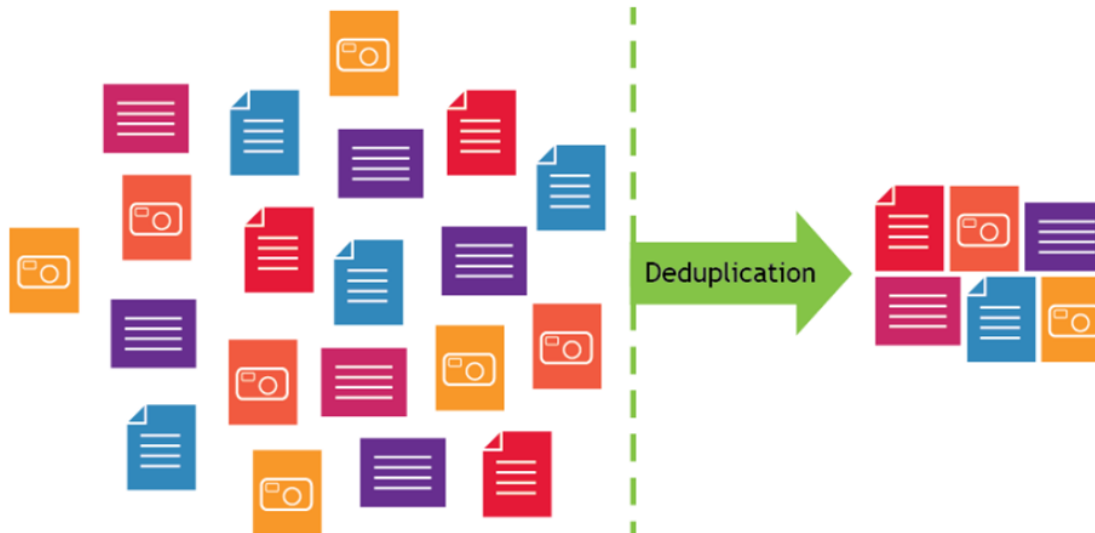
Data Imputation



Infer the missing values from other **close samples** in the dataset.

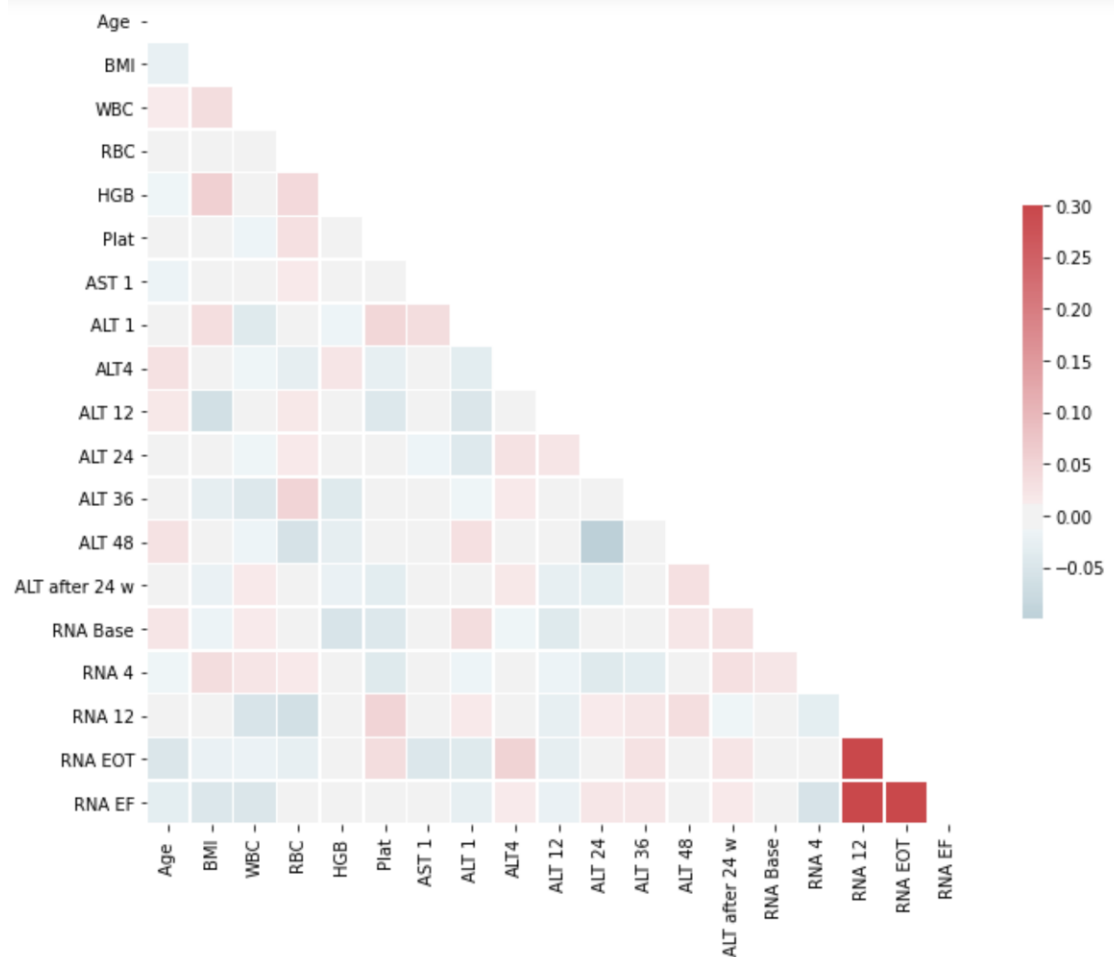
Data Duplication

- Duplication of samples can be a result of repeating experiments, or error in the data.
- It is important to identify and quantify duplication specifically if machine learning models will be built later. **DataFrame.duplicated**
- More importantly, find inconsistency in the data if the features are the same, but the outcomes are different.
 - In this case regression outcomes of duplicate samples can be summarized as mean and variance.



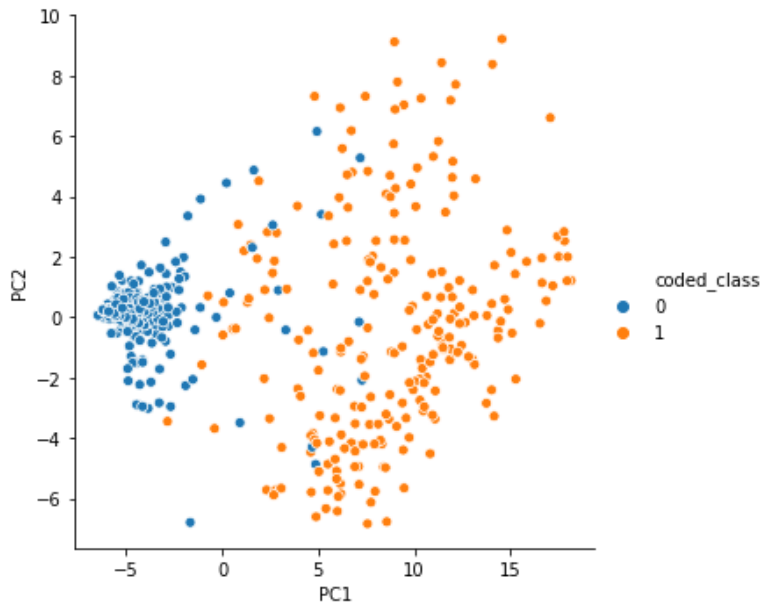
Correlation

- One common step in feature selection is to remove correlated variables.
- Correlation can be computed using `df.corr` and visualized using `seaborn.diverging_palette`

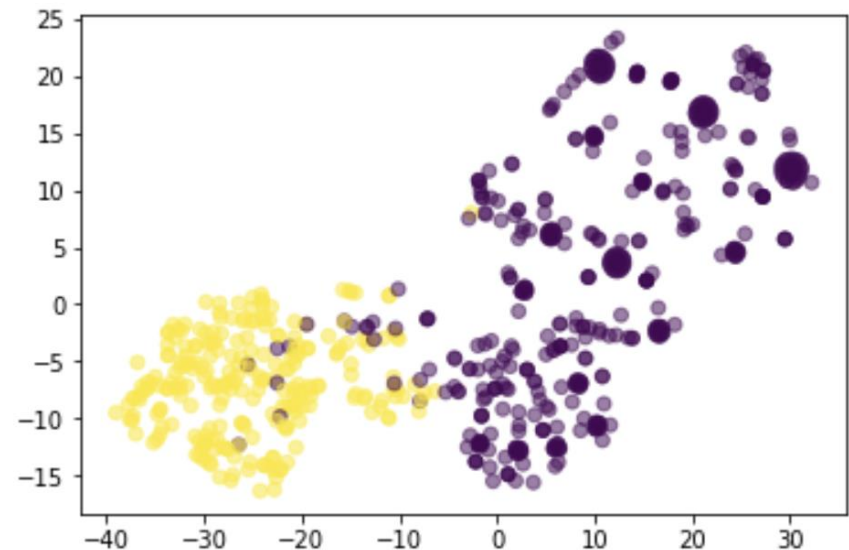


Dimensionality Reduction

- To visualize raw numeric data on a 2D plot, we need to reduce the dimensions.
- Two popular techniques: Principal Component Analysis (PCA) (linear), and TSNE (non linear)



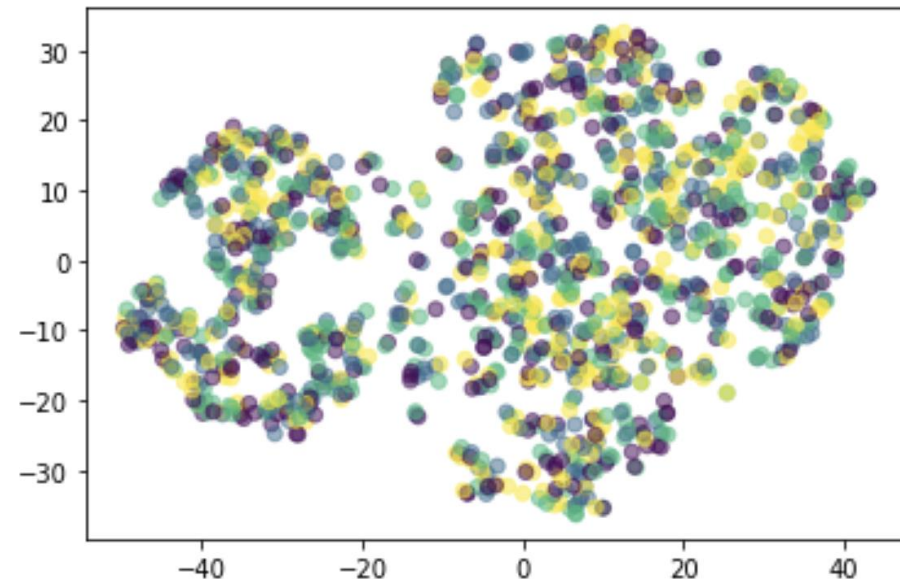
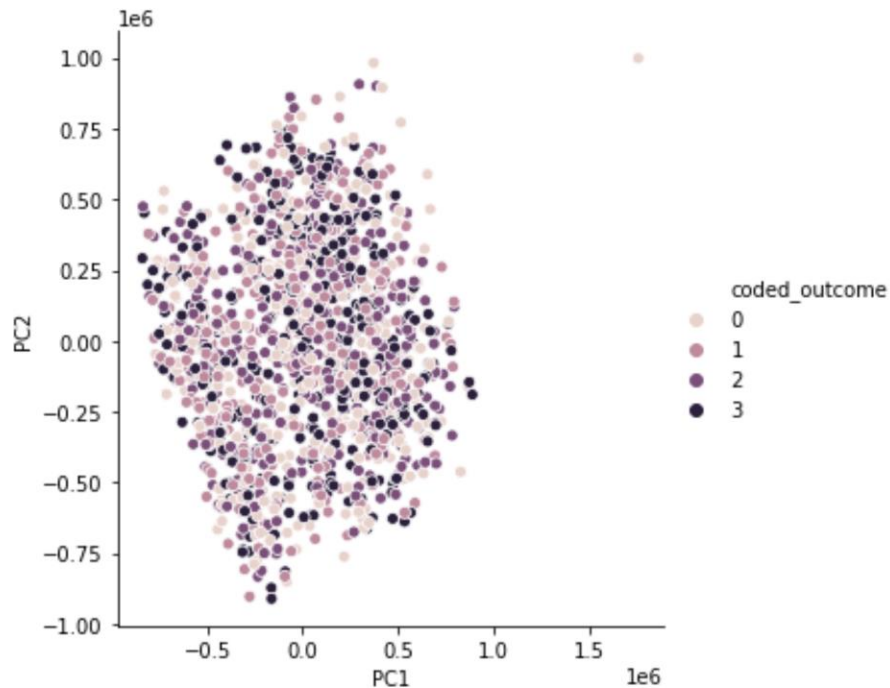
`sklearn.decomposition.PCA`



`sklearn.manifold.TSNE`

PCA and TSNE

- In some data, clusters might not always show up.
- Neural network techniques like Autoencoders may help in this task

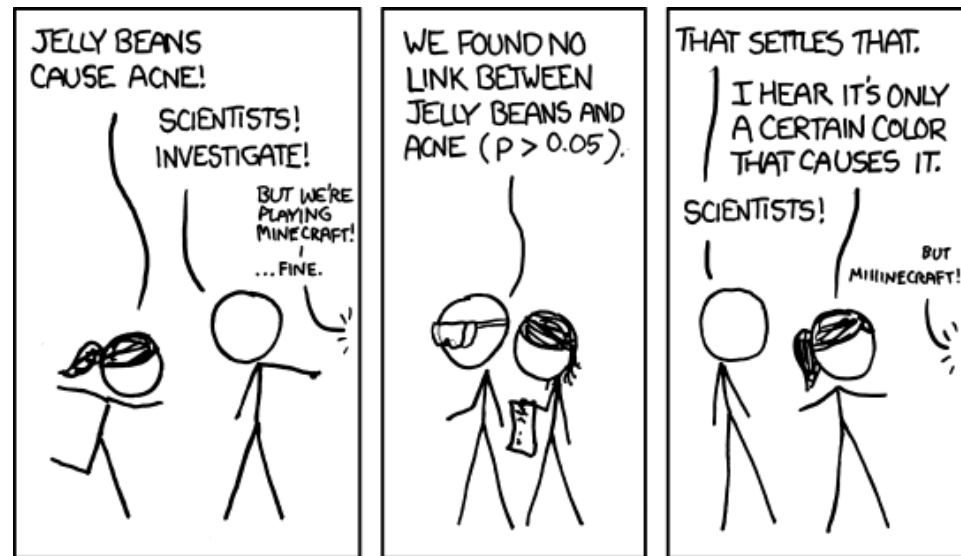


Avoid Data Dredging, p-hacking

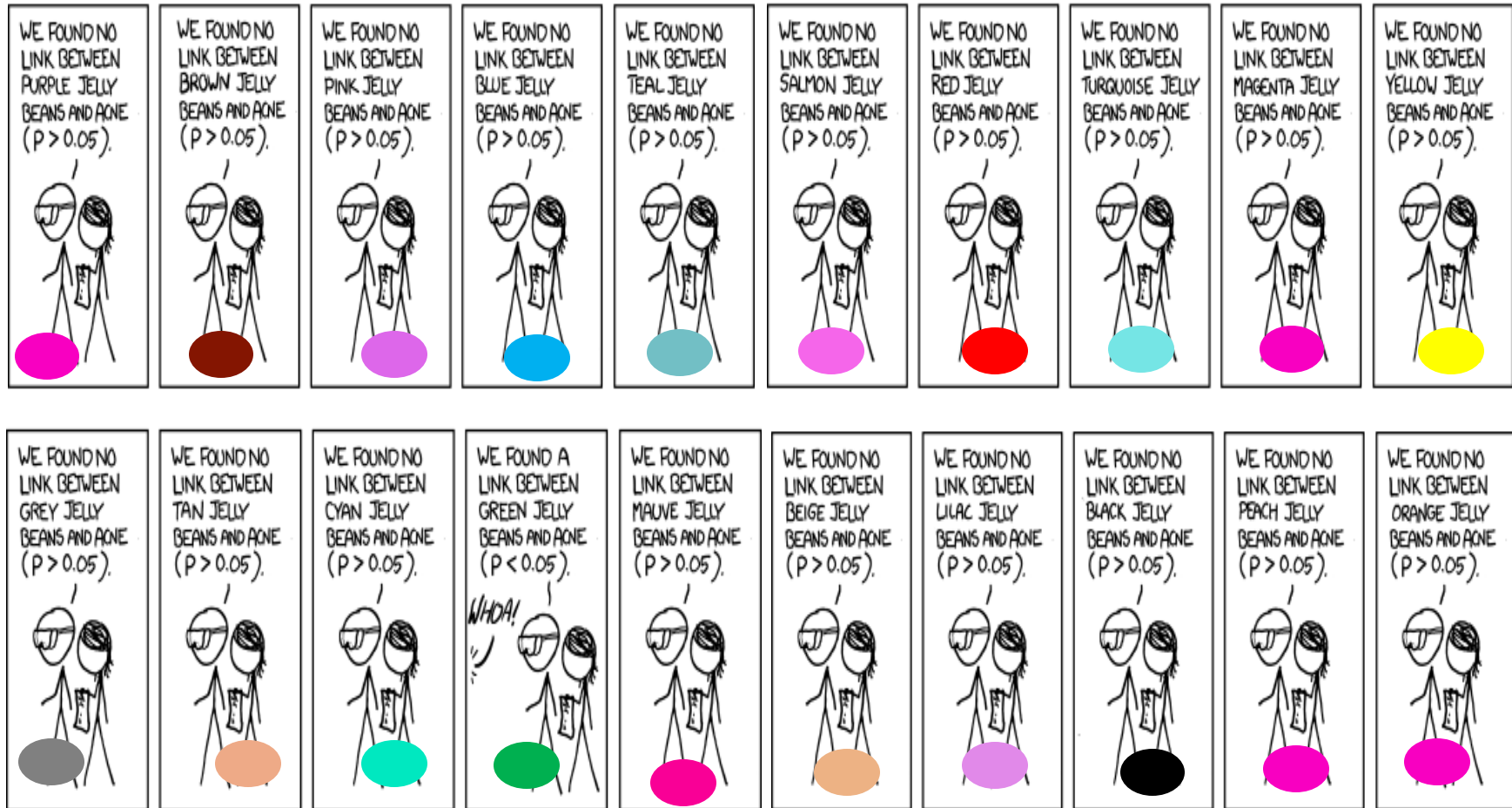
- The purpose of EDA is discovery, whereas the purpose of confirmatory research is validation
- While checking statistical significance between 1000 variables, p value of 5%, with many tests, by chance, 5% will be reported significant.

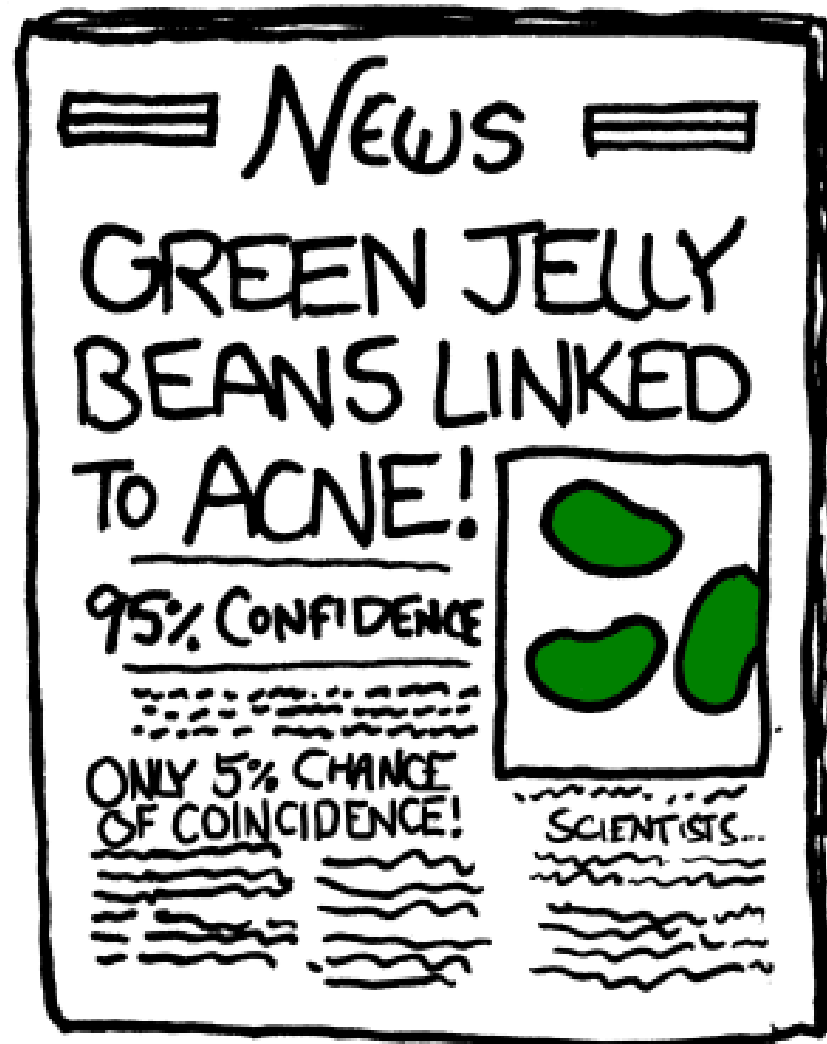
- **Mitigation:**

- Out of sample tests
- Cross validation
- Bonferroni correction



P-hacking





Summary

- It is important to gather facts about the data before applying machine learning.
- EDA tools and analysis techniques help in identifying trends, problems, and possible hypothesis from the data.
- Many python libraries like pandas, sklearn, and seaborn provide very nice tools to conduct EDA.
- It is important not to torture the data and conduct confirmatory data analysis after hypothesis are generated.

Your feedback is valuable to us!

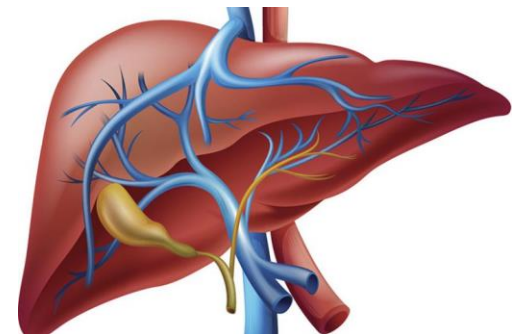
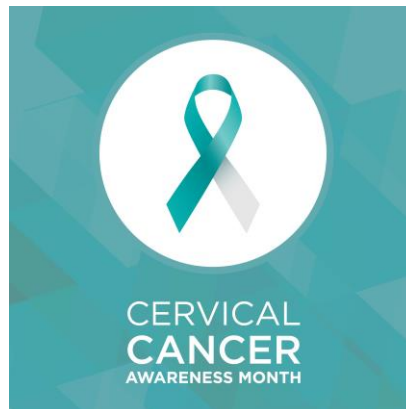


<https://tinyurl.com/yxn5nwk9>

Thank You!

Hands On

- Breast Cancer Wisconsin (Diagnostic) Data Set
- Cervical cancer (Risk Factors) Data Set
- Hepatitis C Virus (HCV) for Egyptian patients Data Set
- Each one of these datasets would highlight different aspects on the application of EDA to better understand the data.
- Github examples: <https://github.com/georgezakinih/exploratory-data-analysis>



References & Examples

- [Pandas](#), [Seaborn](#), [Matplotlib](#)
- [Exploratory Data Analysis](#), Oxford Bibliographis
- [An extensive guide to EDA](#)
- [Heart attack risk prediction](#)
- Andrew T. Jebb, Scott Parrigon, Sang Eun Woo, ***“Exploratory data analysis as a foundation of inductive research”***, Human Resource Management Review, Volume 27, Issue 2, 2017
- Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). ***“False-Positive Psychology: Undisclosed Flexibility in Data Collection and Analysis Allows Presenting Anything as Significant.”*** Psychological Science, 22(11), 1359–1366.