BBM 101

Introduction to Programming I

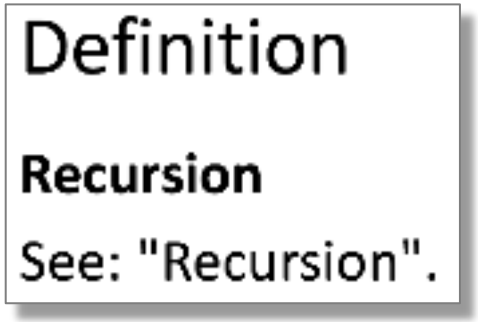
Lecture #12 – Understanding Data

HACETTEPE UNIVERSITY

Fuat Akal & Erkut Erdem // Fall 2020

Last time... **Recursion**

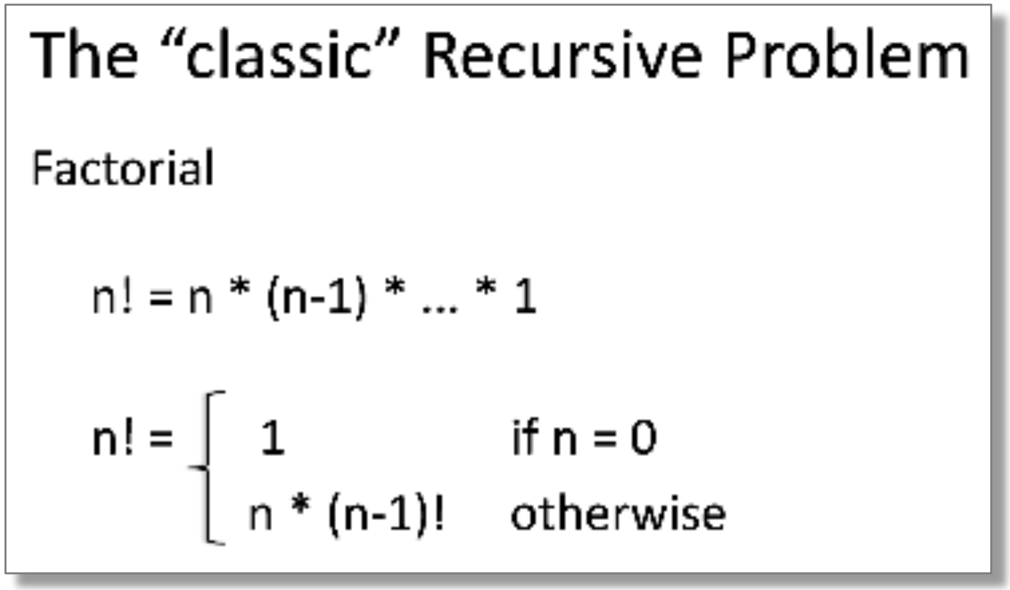
Recursion is a programming concept whereby a function invokes itself.



Definition

Recursion

See: "Recursion".

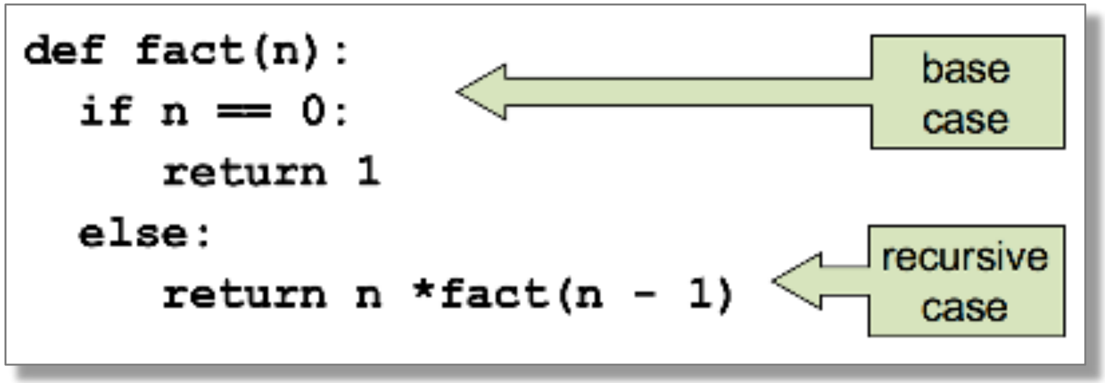


The “classic” Recursive Problem

Factorial

$$n! = n * (n-1) * \ldots * 1$$

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n * (n-1)! & \text{otherwise} \end{cases}$$



```
def fact(n):
  if n == 0:
     return 1
  else:
     return n *fact(n - 1)
```

Lecture Overview

- Introduction to Data Science
  - Data, Data Science, Data Scientist…
- Python Libraries to Analyse Data
  - Pandas
  - Numpy
  - Matplotlib
- Your -Probably the- First Data Science Project

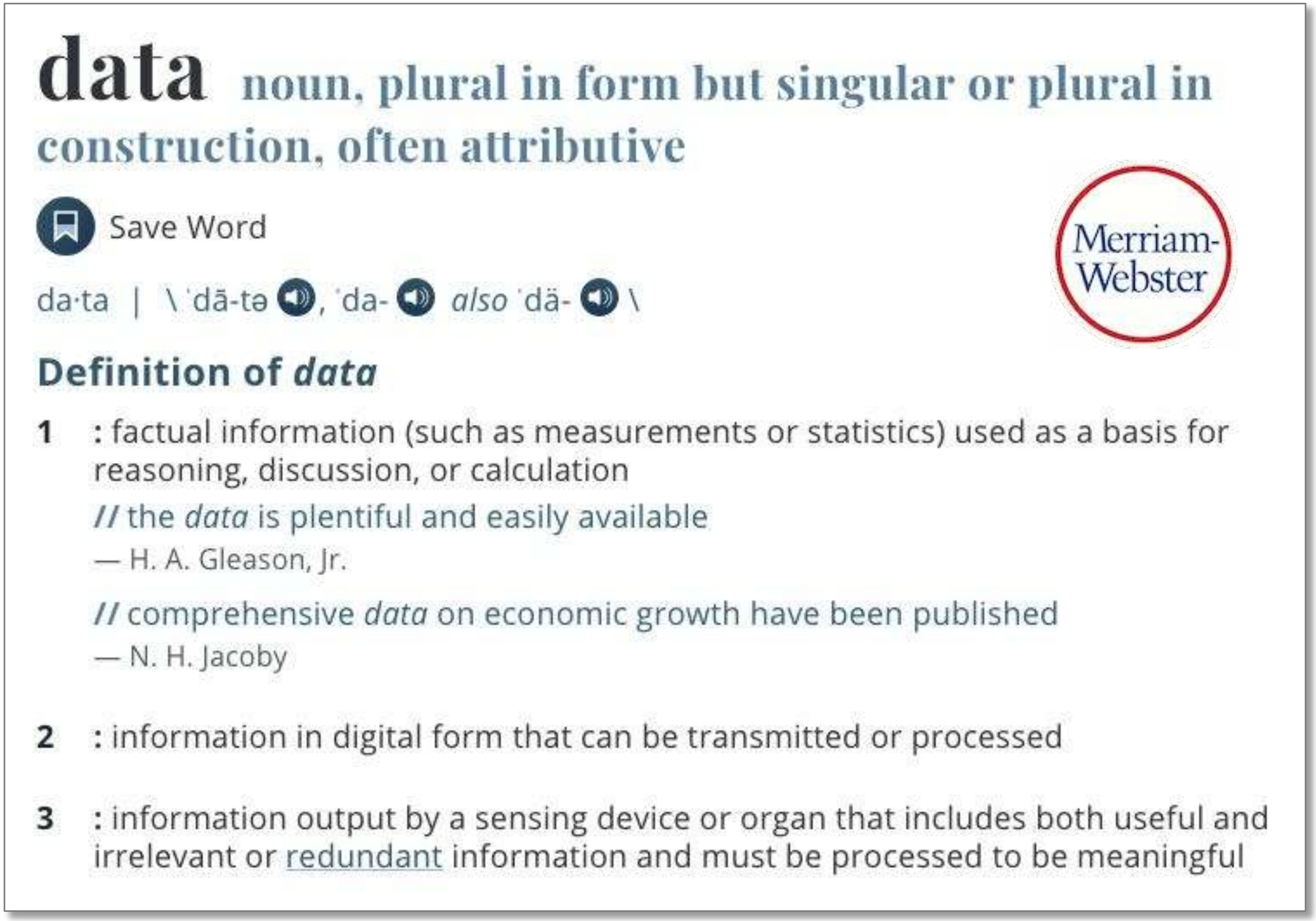
Disclaimer: Much of the material and slides for this lecture were borrowed from

- IBM Courses at Coursera, https://www.coursera.org/professional-certificates/ibm-data-science
- CS109 Data Science course at Harvard University, by Rafael A. Irizarry and Verena Kaynig-Fittkau.
- Python Numpy Tutorial by Justin Johnson.

Lecture Overview

- Introduction to Data Science
  - Data, Data Science, Data Scientist…
- Python Libraries to Analyse Data
  - Pandas
  - Numpy
  - Matplotlib
- Your -Probably the- First Data Science Project

What is Data?



data noun, plural in form but singular or plural in construction, often attributive

Save Word

da·ta | \ ˈdā-tə, ˈda- *also* ˈdä- \

Definition of *data*

1 : factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation

// the *data* is plentiful and easily available
— H. A. Gleason, Jr.

// comprehensive *data* on economic growth have been published
— N. H. Jacoby

2 : information in digital form that can be transmitted or processed

3 : information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful

What is Data Science?

- Data science is the study of data.
- It involves developing methods of recording, storing, and analyzing data to effectively extract useful information.
- The goal of data science is to gain insights and knowledge from any type of data — both structured and unstructured.

Who are Data Scientists?

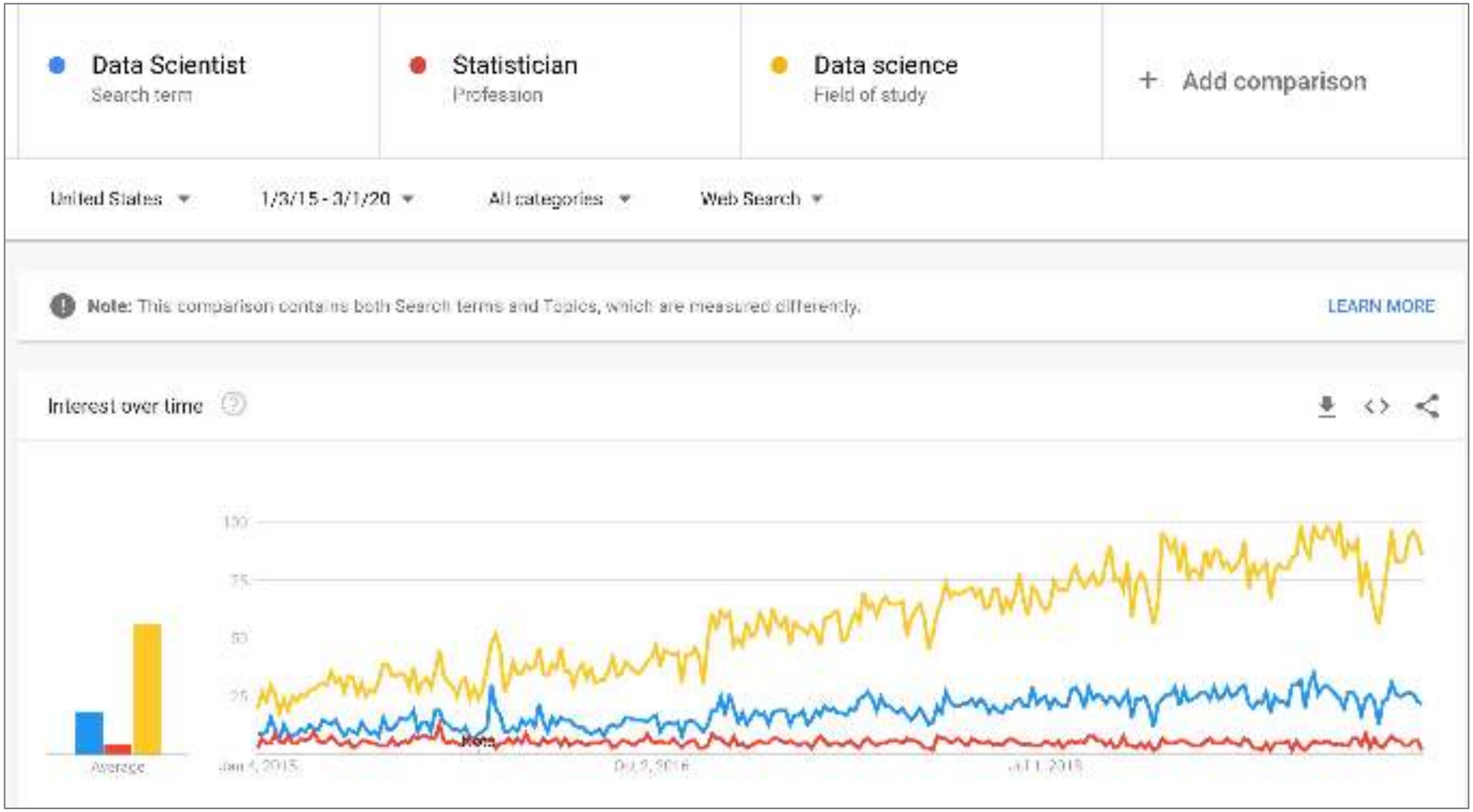
- They are part **mathematician**, part **statistician**, part **computer scientist** and part **trend-spotter**.
- They straddle both the business and IT worlds.
- They are highly **sought-after** and **well-paid**.

"I keep saying the sexy job in the next ten years will be statisticians. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s?"

- Hal Varian, Google's Chief Economist

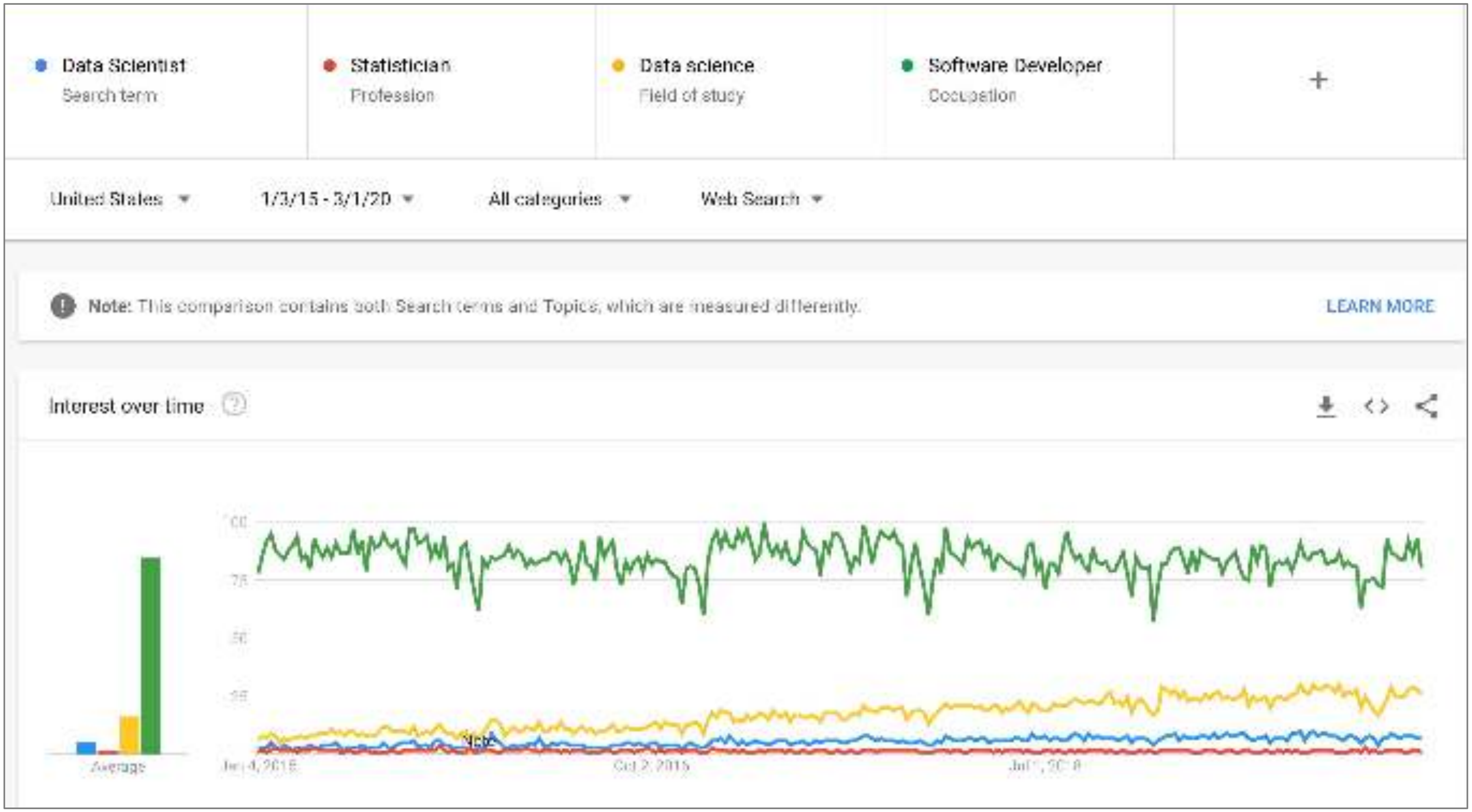
Data Scientist is Sexy?

Google Trends



Maybe not?

Google Trends



● Data Scientist
Search term

● Statistician
Profession

● Data science
Field of study

● Software Developer
Occupation

+

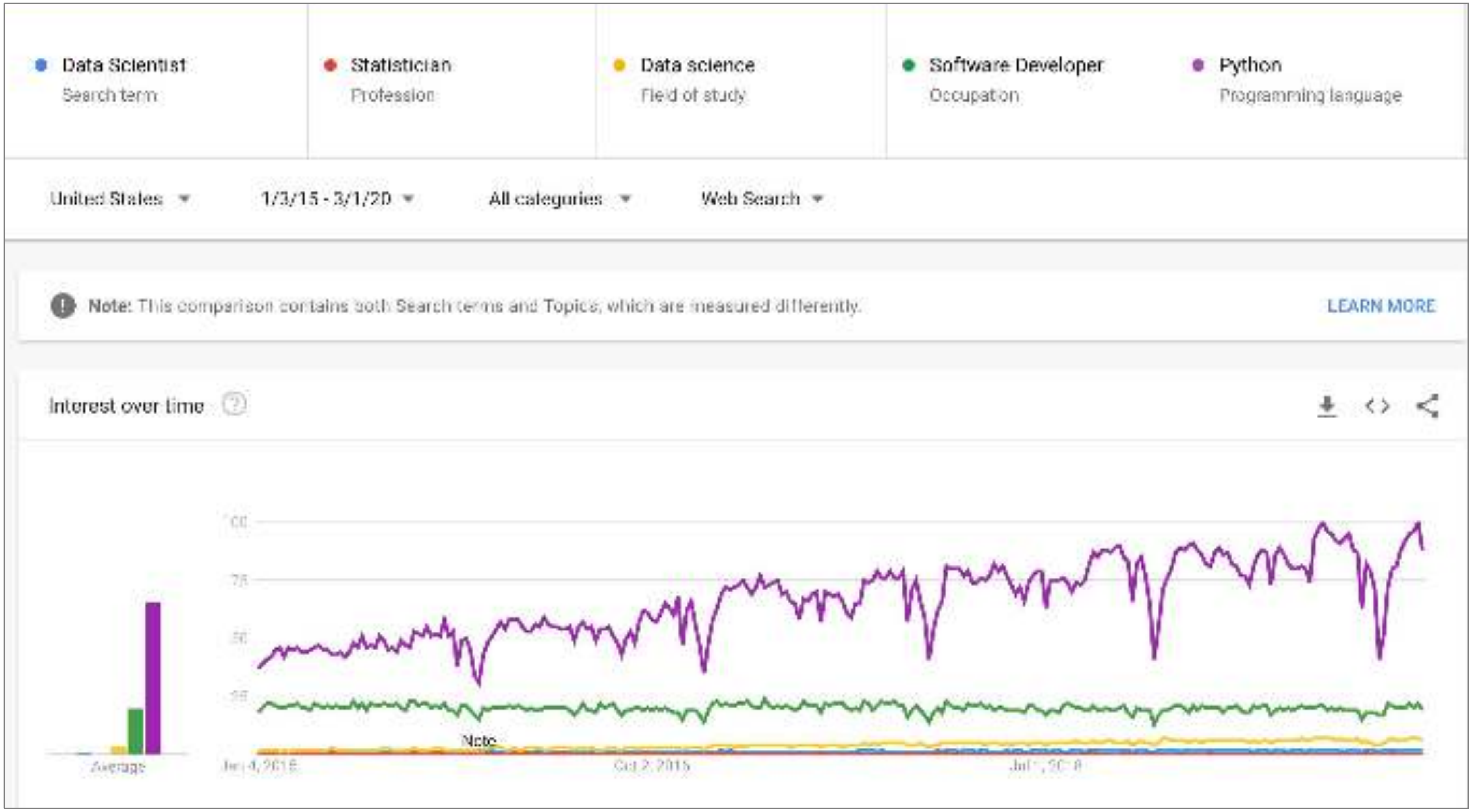
United States ▾ 1/3/15 - 3/1/20 ▾ All categories ▾ Web Search ▾

Note: This comparison contains both Search terms and Topics, which are measured differently. LEARN MORE

Interest over time

Maybe It is

Google Trends



● Data Scientist
Search term

● Statistician
Profession

● Data science
Field of study

● Software Developer
Occupation

● Python
Programming language

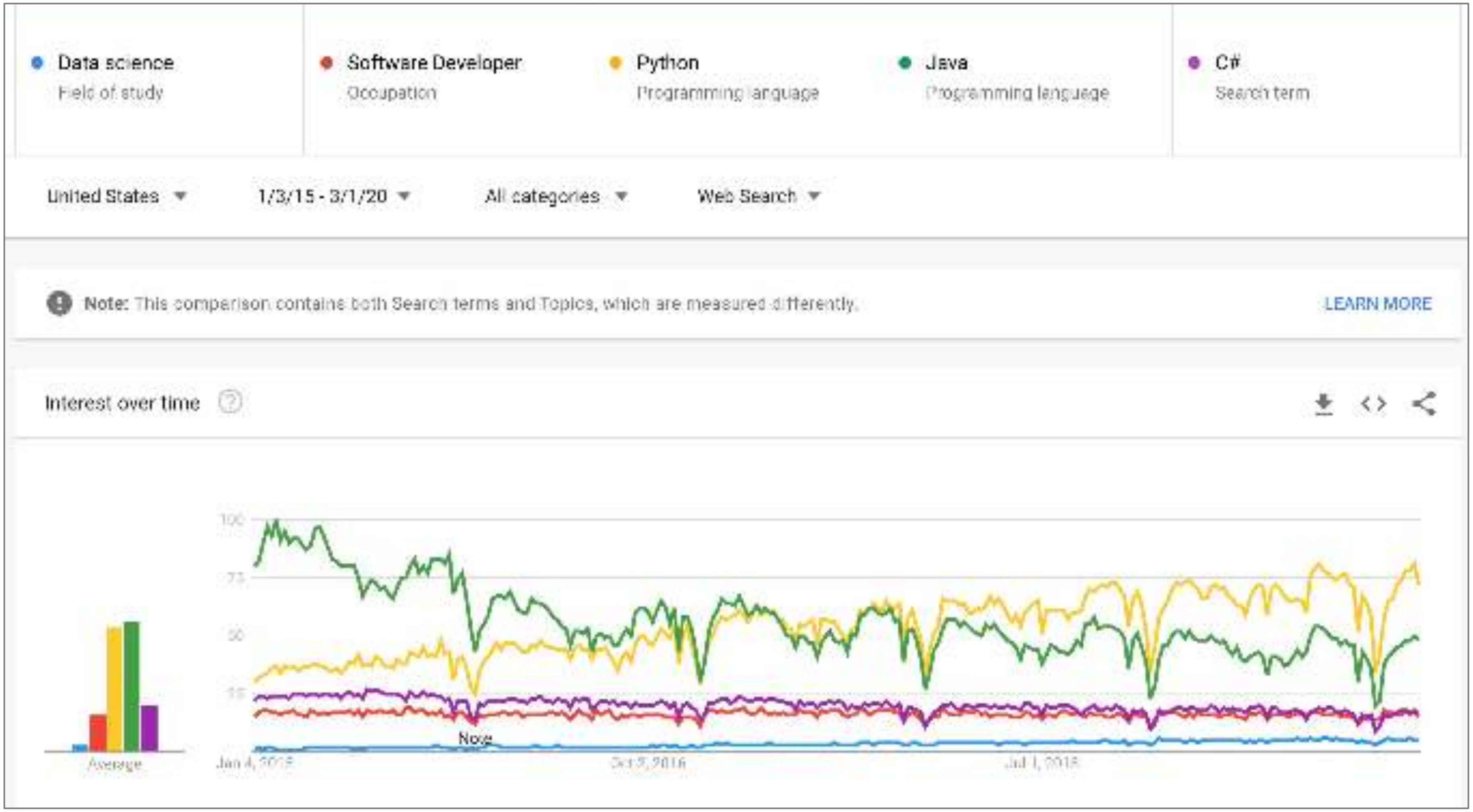
United States ▾ 1/3/15 - 3/1/20 ▾ All categories ▾ Web Search ▾

Note: This comparison contains both Search terms and Topics, which are measured differently. LEARN MORE

Interest over time

Not Quite Sure

Google Trends



● Data science
Field of study

● Software Developer
Occupation

● Python
Programming language

● Java
Programming language

● C#
Search term

United States ▾ 1/3/15 - 3/1/20 ▾ All categories ▾ Web Search ▾

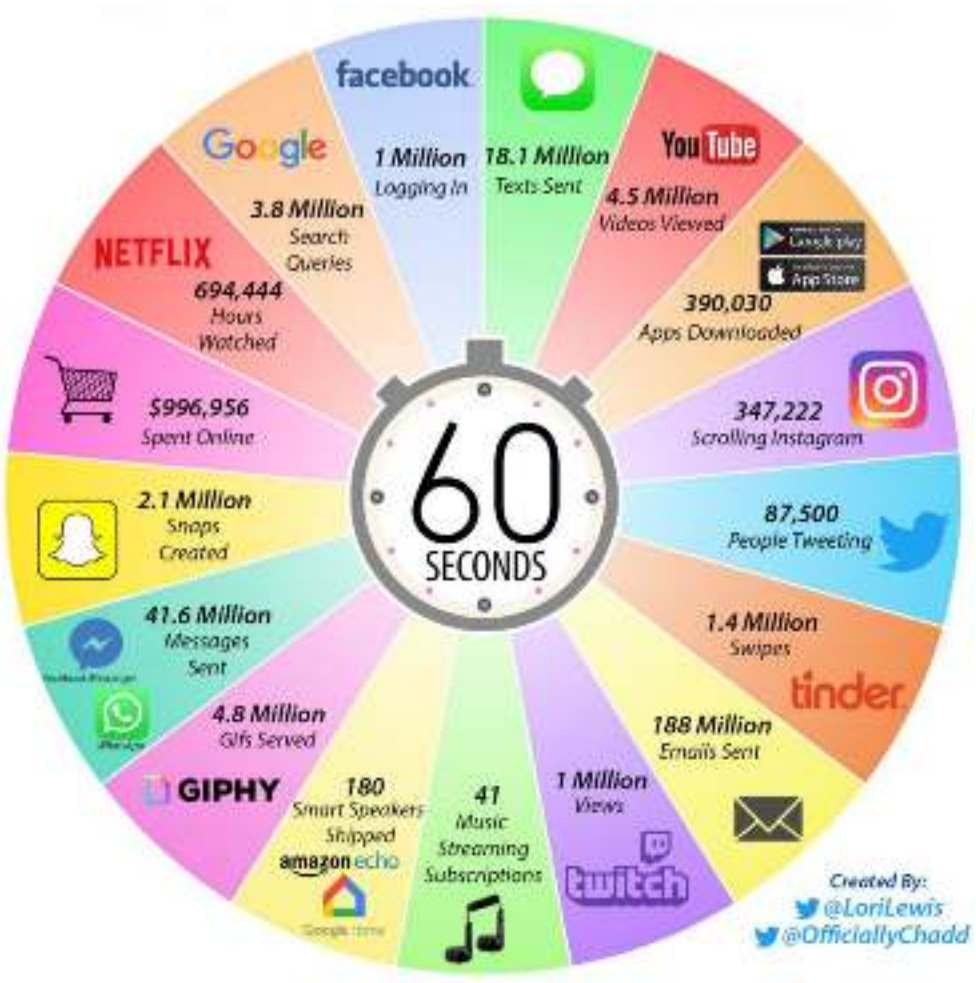
Note: This comparison contains both Search terms and Topics, which are measured differently. LEARN MORE

Interest over time

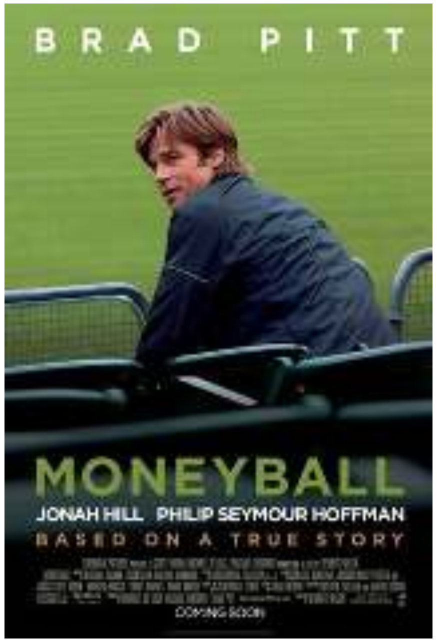
Everybody is Talking about Data

Why Data Science Now?

- We are producing more and more data every minute via
  - Sensors
  - Video Surveillance Cameras
  - Browsing Web
  - Medical Instruments
  - …
- The biggest data source we have today is Internet
  - Currently at Exabytes
- Getting insights out of data is crucial as we want to
  - Build better football teams
  - Sell more products
  - Avoid fraud
  - Find treatments
  - …

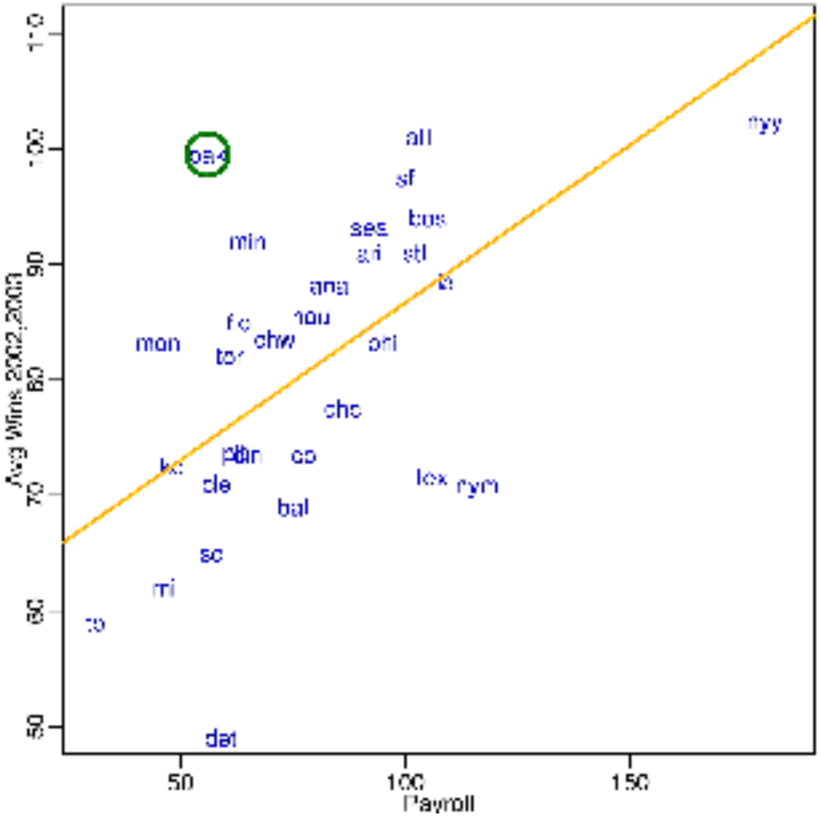


Example Data Science Project "Moneyball"

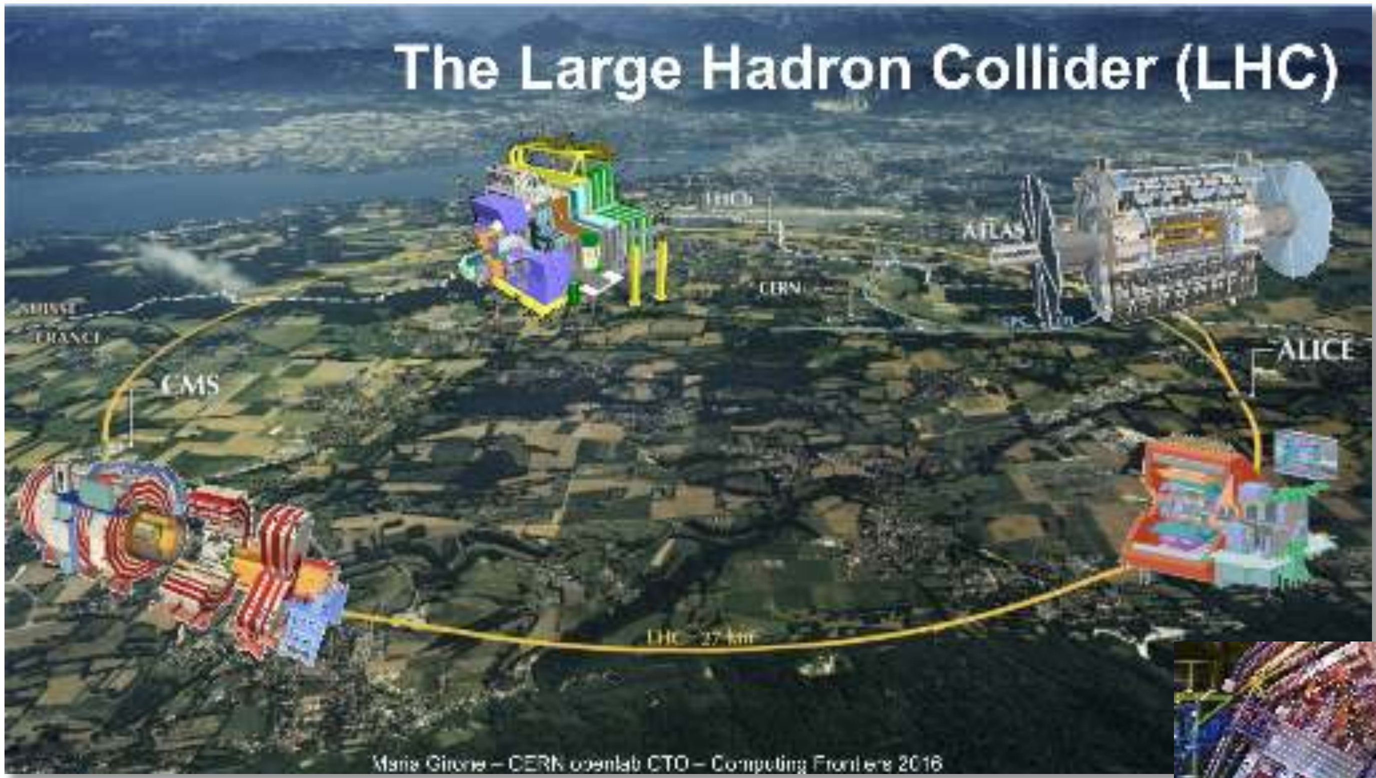


The Oakland A's picked players that scouts thought no good but data said otherwise.

Oakland A's general manager Billy Beane's successful attempt to assemble a baseball team on a lean budget by employing computer-generated analysis to acquire new players.



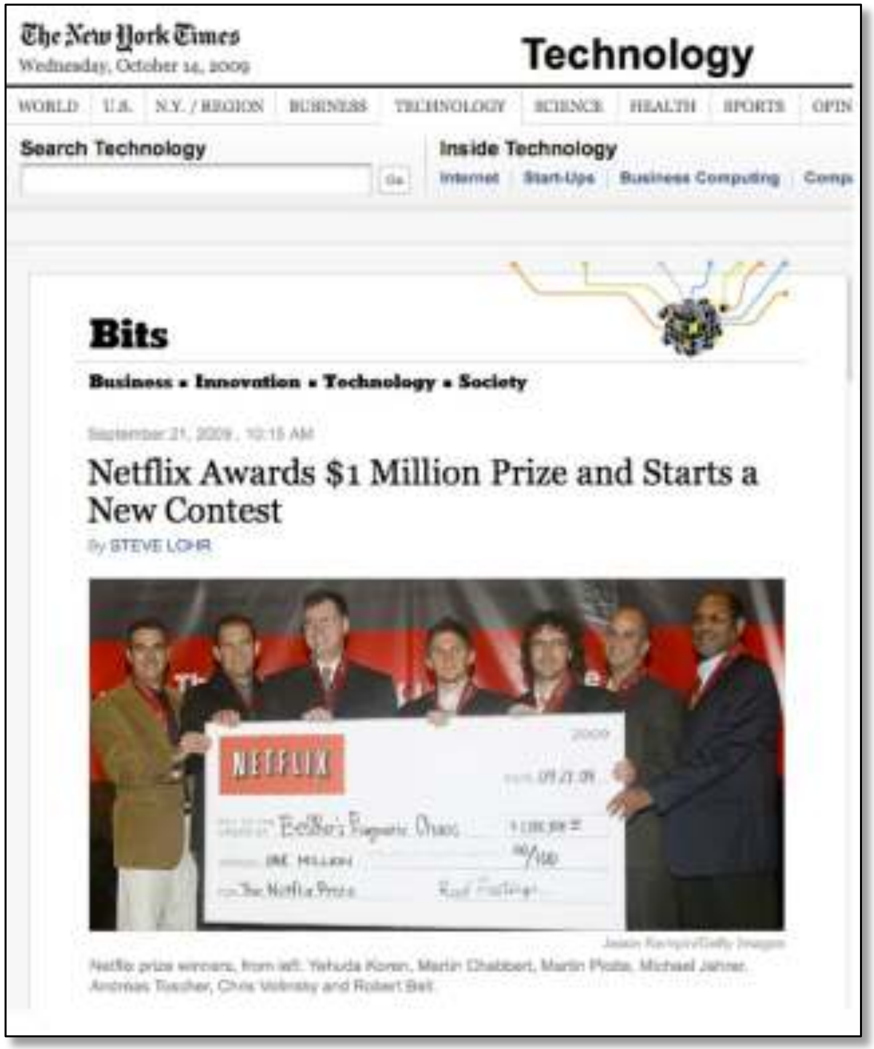
Data Analysis in Physics


The Large Hadron Collider (LHC)
LHCb
ATLAS
CERN
SUISSE
FRANCE
CMS
ALICE
LHC 27 km
Maria Girone – CERN openlab CTO – Computing Frontiers 2016

50 Petabytes of data per year!

Netflix Challenge

- October 2006: Netflix offers $1M for an improved recommender algorithm.
- 6 years of data for training: 2000-2005
- $1M grand prize for 10% improvement



The New York Times
Wednesday, October 14, 2009

Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPIN

Search Technology

Inside Technology
Internet | Start-Ups | Business Computing | Comp

Bits

Business • Innovation • Technology • Society

September 21, 2009, 10:15 AM

Netflix Awards $1 Million Prize and Starts a New Contest

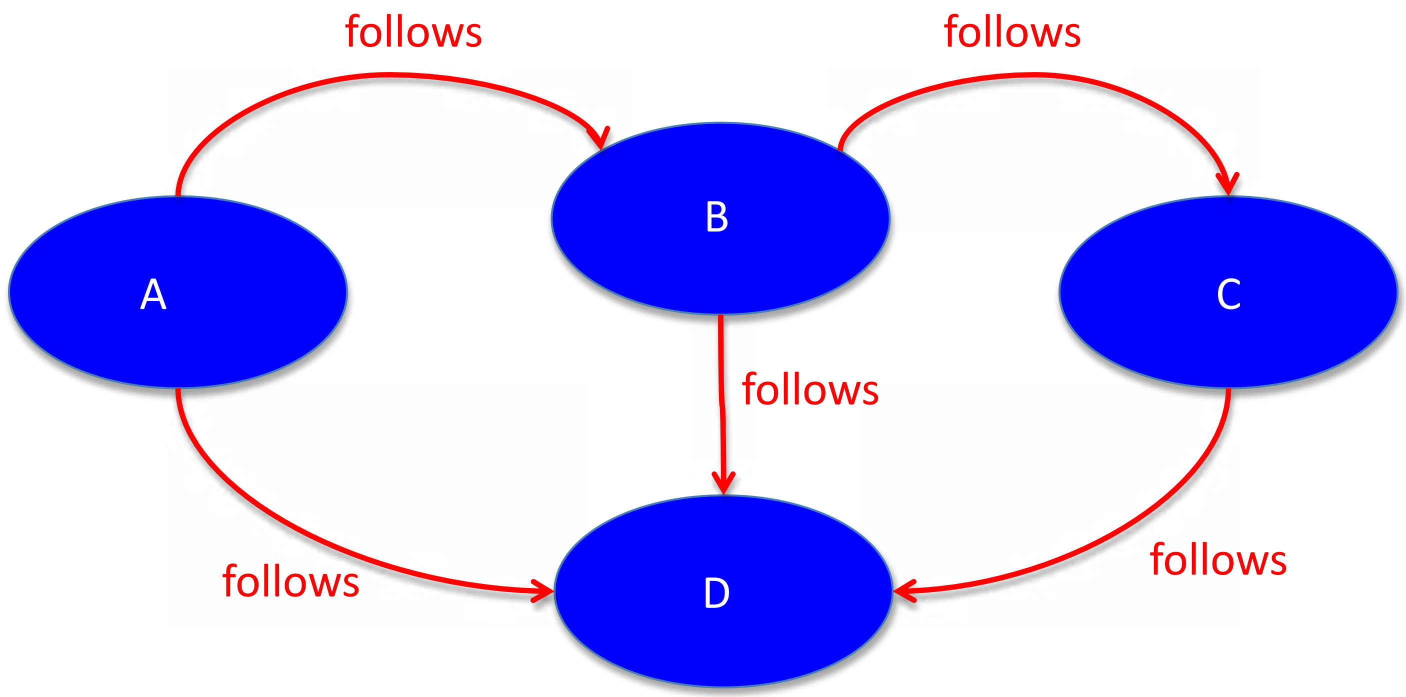
By STEVE LOHR

Jason Kempin/Getty Images

Netflix prize winners, from left: Yehuda Koren, Martin Chabbert, Martin Piotte, Michael Jahrer, Andreas Toscher, Chris Volinsky and Robert Bell

Discover Relationships

What would you think about person D?



Facebook Bought WhatsApp for $19 Billion in 2014.

Can you tell why?

WhatsApp with Facebook's $19B offer?

In a play to dominate messaging on phones and the Web, Facebook has acquired WhatsApp for $19 billion.

Btw, WhatsApp had 55 employees in 2014. Currently, the number is 120.

How Do We Do Data Science?

- **Science**: determining what questions can be answered with data and what are the best datasets for answering them
- **Computer Programming**: using computers to analyze data
- **Data Wrangling**: getting data into analyzable form on our computers
- **Statistics**: separating signal from noise
- **Machine Learning**: making predictions from data
- **Communication:** sharing findings through visualization, stories and interpretable summaries

Lecture Overview

- Introduction to Data Science
  - Data, Data Science, Data Scientist…
- Python Libraries to Analyse Data
  - Pandas
  - Numpy
  - Matplotlib
- Your -Probably the- First Data Science Project

Python Libraries to Analyse Data

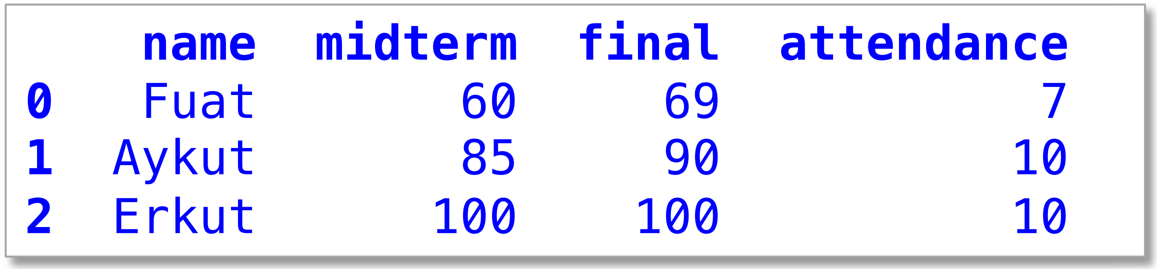
- Pandas
  - Provides data structures and operations for data (e.g. tables and time series) manipulation and analysis.
- Numpy
  - Provides means to work with multidimensional arrays.
- Matplotlib
  - A plotting library used to create high-quality graphs, charts, and figures.

Pandas

- A library that contains high-performance, easy-to-use data structures and data analysis tools.
- Some important aspects of Pandas
  - A fast and efficient DataFrame object for data manipulation with integrated indexing.
  - Tools for reading and writing data in different formats, e.g. csv, Excel, SQL Database.
  - Slicing, indexing, subsetting, merging and joining of huge datasets.
- Typically imported as `import pandas as pd` in Python programs

Create DataFrames using Dictionaries

```
import pandas as pd

data = { 'name': ['Fuat', 'Aykut', 'Erkut'],
          'midterm': [60, 85, 100],
          'final': [69, 90, 100],
          'attendance': [6, 10, 10]
}

df_bbm101 = pd.DataFrame(data)

print(df_bbm101.head())# Prints top 5 rows
```



```
     name  midterm  final  attendance
0    Fuat       60     69           7
1   Aykut       85     90          10
2   Erkut      100    100          10
```

Same Thing, in Another Way

```
names = ['Fuat', 'Aykut', 'Erkut']
midterms = [60, 85, 100]
finals = [69, 90, 100]
attendances = [6, 10, 10]

list_labels = ['name', 'midterm', 'final', 'attendance']
list_cols = [names, midterms, finals, attendances]

zipped = list(zip(list_labels, list_cols))

print(zipped)     # [('name', ['Fuat', 'Aykut', 'Erkut']),
                  # ('midterm', [60, 85, 100]),
                  # ('final', [69, 90, 100]),
                  # ('attendance', [6, 10, 10])]

data = dict(zipped)

df_bbm101 = pd.DataFrame(data)
```

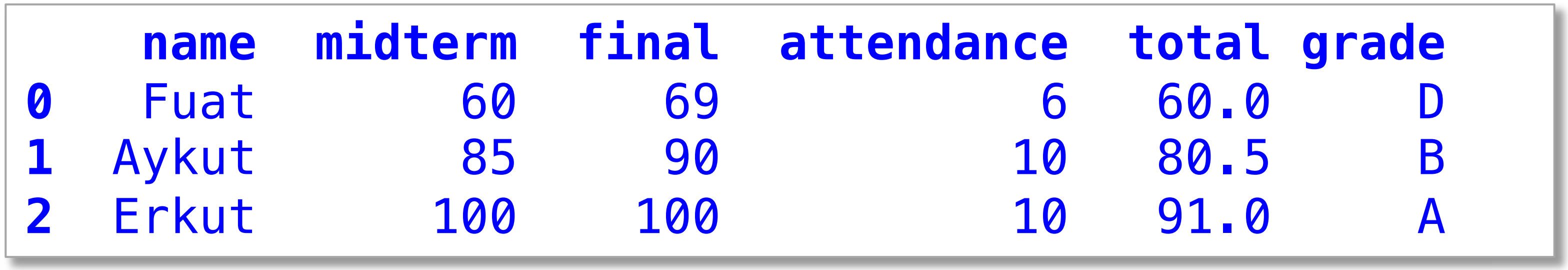
Broadcasting

```
df_bbm101['total'] = 0
# Adds new column to df and
# broadcasts 0 to entire column

print(df_bbm101.head())
```

```
   name    midterm  final  attendance  total
0 Fuat          60     69           6      0
1 Aykut         85     90          10      0
2 Erkut        100    100          10      0
```

Compute Columns

```
df_bbm101['total'] = df_bbm101['midterm']*0.3 + \
                      df_bbm101['final']*0.6 + \
                      df_bbm101['attendance']*0.1

df_bbm101.loc[(df_bbm101['total'] >= 60) &
          (df_bbm101['total'] < 70), 'grade'] = 'D’
…            # Code to compute Bs and Cs comes here
df_bbm101.loc[df_bbm101['total'] >= 90, 'grade'] = 'A’

print(df_bbm101.head())
```



```
     name  midterm  final  attendance  total grade
0    Fuat       60     69           6   60.0     D
1  Aykut       85     90          10   80.5     B
2  Erkut      100    100          10   91.0     A
```

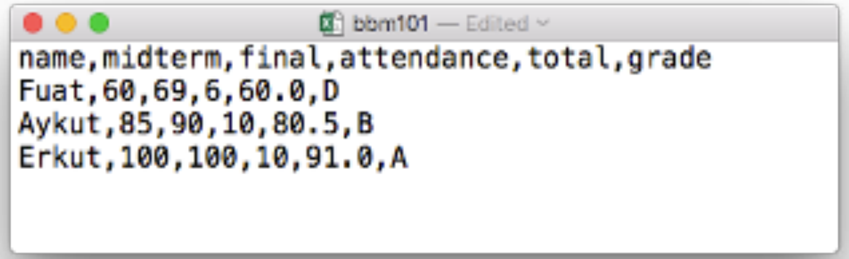
Beware that Fuat would not make it if he missed just one more lecture ;-)

Subsetting/Slicing Data

```
print(df_bbm101[['name', 'grade']])

print(df_bbm101.iloc[:, [0, 5]])

print(df_bbm101.iloc[:, [True, False, False, False,
                          False, True]])

# They all return the same thing
# name and grade columns of the df
# Same principle can be applied to rows as well
```

```
     name  grade
0    Fuat      D
1   Aykut      B
2   Erkut      A
```

DataFrames from CSV Files



```
name,midterm,final,attendance,total,grade
Fuat,60,69,6,60.0,D
Aykut,85,90,10,80.5,B
Erkut,100,100,10,91.0,A
```

file name: bbm101.csv

```
df_bbm101 = pd.read_csv('bbm101.csv')
print(df_bbm101.head())
```

```
    name  midterm  final  attendance  total grade
0   Fuat       60     69           6   60.0     D
1  Aykut       85     90          10   80.5     B
2  Erkut      100    100          10   91.0     A
```

Indexing DataFrames

```
df_bbm101 = pd.read_csv('bbm101.csv’, index_col ='name')
print(df_bbm101.head())
```

```
        midterm  final  attendance  total grade
  name
  Fuat       60     69           6   60.0     D
 Aykut       85     90          10   80.5     B
 Erkut      100    100          10   91.0     A
```

```
print(df_bbm101.loc['Fuat’])
```

```
midterm        60
final          69
attendance      6
total          60
grade           D
Name: Fuat, dtype: object
```

```
print(df_bbm101.
   loc[['Aykut', 'Erkut']])
```

```
        midterm  final  attendance  total grade
  name
 Aykut       85     90          10   80.5     B
 Erkut      100    100          10   91.0     A
```

Numpy

- A library for the Python programming language, adding support for large **multi-dimensional arrays and matrices**,
  - along with a large collection of high-level mathematical functions to operate on these arrays.

- A numpy array is a grid of values, **all of the same type**, and is indexed by a tuple of nonnegative integers.

- The number of dimensions is the **rank** of the array.

- The **shape** of an array is a tuple of integers giving the size of the array along each dimension.

- Typically imported as `import numpy as np` in Python programs

Creating Numpy Arrays

```
import numpy as np

a = np.array([1,2,3])     # Create a rank 1 array
print(type(a))            # <class 'numpy.ndarray'>
print(a.shape)            # (3,)
print(a)                  # [1 2 3]
print(a[0], a[1], a[2])   # 1 2 3


b = np.array([[1,2,3],[4,5,6]])    # Create a rank 2 array
print(b.shape)                     # (2, 3)
print(b)                           # [[1 2 3]
                                   #  [4 5 6]]

print(b[0, 0], b[0, 1], b[1, 0])   # 1 2 4
```

Miscellaneous Ways to Create Arrays

```
a = np.zeros((2,2))    # Create an array of all zeros
print(a)               # [[ 0.  0.]
                       #  [ 0.  0.]]

b = np.ones((1,2))     # Create an array of all ones
print(b)               # [[ 1.  1.]]

c = np.full((2,2), 7)  # Create a constant array
print(c)               # [[ 7.  7.]
                       #  [ 7.  7.]]

d = np.eye(2)          # Create a 2x2 identity matrix
print(d)               # [[ 1.  0.]
                       #  [ 0.  1.]]

e = np.random.random((2,2))  # Create an array filled with
                             # random values
print(e)                     # Might print
                             # [[ 0.91940167  0.08143941]
                             #  [ 0.68744134  0.87236687]]
```

Indexing Arrays

- Slicing
- Integer Indexing
- Boolean (or, Mask) Indexing

Slicing

- Similar to slicing Python lists.
- Since arrays may be multidimensional, you must specify a slice for each dimension of the array.
- Slices are views (not copies) of the original data.

Slicing Examples

```
a = np.array([[1, 2, 3, 4],     # Create a rank 2 array
              [5, 6, 7, 8],     # with shape (3, 4)
              [9, 10, 11, 12]])


print(a)                        # [[ 1  2  3  4]
                                #  [ 5  6  7  8]
                                #  [ 9 10 11 12]]
b = a[:2, 1:3]
print(b)                        # [[ 2  3 ]
                                #  [ 6  7 ]

print(a[1, :])                  # [5 6 7 8]

print(a[:, :-2])                # [[ 1  2]
                                #  [ 5  6]
                                #  [ 9 10]]
```

Integer Indexing

- NumPy arrays may be indexed with other arrays.
- Index arrays must be of integer type.
- Each value in the array indicates which value in the array to use in place of the index.
- Returns a copy of the original data.

Integer Indexing Examples

```
a = np.array([1, 2, 3, 4, 5, 6])
print(a)                                       # [1 2 3 4 5 6]
print(a[[1, 3, 5]])                            # [2 4 6]


a = np.array([[1, 2], [3, 4], [5, 6]])
print(a)                                       # [[ 1  2 ]
                                               #  [ 3  4 ]
                                               #  [ 5  6 ]]


# The returned array will have shape (3,)
print(a[[0, 1, 2], [0, 1, 0]])                        # [1 4 5]
print(np.array([a[0, 0], a[1, 1], a[2, 0]]))  # [1 4 5]

# The same element from the source array can be reused
print(a[[0, 0], [1, 1]])                              # [2 2]
print(np.array([a[0, 1], a[0, 1]]))                   # [2 2]
```

Boolean (or, Mask) Indexing

- Boolean array indexing lets you pick out arbitrary elements of an array.
- Frequently used to select the elements of an array that satisfy some condition.
  - Thus, called the mask indexing.

Boolean (or, Mask) Indexing Examples

```
a = np.array([1, 2, 3, 4, 5, 6])

bool_idx = (a > 2)
# Find the elements of a that are bigger than 2;
# this returns a numpy array of Booleans of the same
# shape as a, where each slot of bool_idx tells
# whether that element of a is > 2.

print(bool_idx)                # [False False  True
                               #             True  True  True]

# We use boolean array indexing to construct a rank 1 array
# consisting of the elements of a corresponding to the True
# values of bool_idx
print(a[bool_idx])                # [3 4 5 6]

# We can do all of the above in a single concise statement:
print(a[a > 2])                   # [3 4 5 6]
```

Array Math

- Basic mathematical functions operate elementwise on arrays.

```
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])
# [[1 2]    [[5 6]
#  [3 4]]    [7 8]]

# Elementwise sum
print(x + y)
print(np.add(x, y))
# [[ 6  8]
#  [10 12]]

# Elementwise product
print(x * y)
print(np.multiply(x, y))
# [[ 5 12]
#  [21 32]]
```

Same principle holds for “`np.divide`, /” and “`np.subtract`, –”

Array Math (Cont'd)

```
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])

v = np.array([9, 10]
w = np.array([11, 12])
```

```
# Inner product of vectors;
# both produce 219
print(v.dot(w))
print(np.dot(v, w))

# Matrix / vector product;
# both produce the rank 1
# array [29 67]
print(x.dot(v))
print(np.dot(x, v))
```

```
# Matrix / matrix product;
# both produce a rank 2 array
# [[19 22]
#  [43 50]]
print(x.dot(y))
print(np.dot(x, y))

# Transpose of x
# [[1 3]
#  [2 4]]
print(x.T)
```

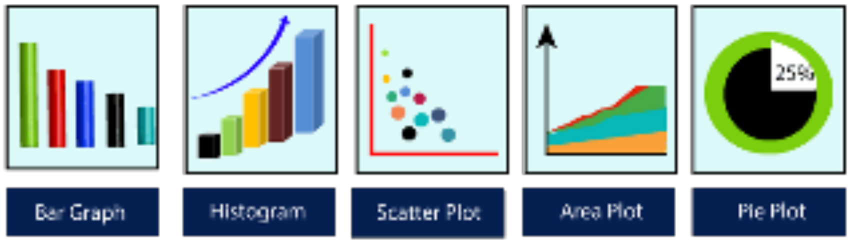
Matplotlib

- Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments.

- Typically imported as `import matplotlib.pyplot as plt` in Python programs.

- Pyplot is a module of Matplotlib which provides simple functions to add plot elements like lines, images, text, etc.

- There are many plot types. Some of are more frequently used.

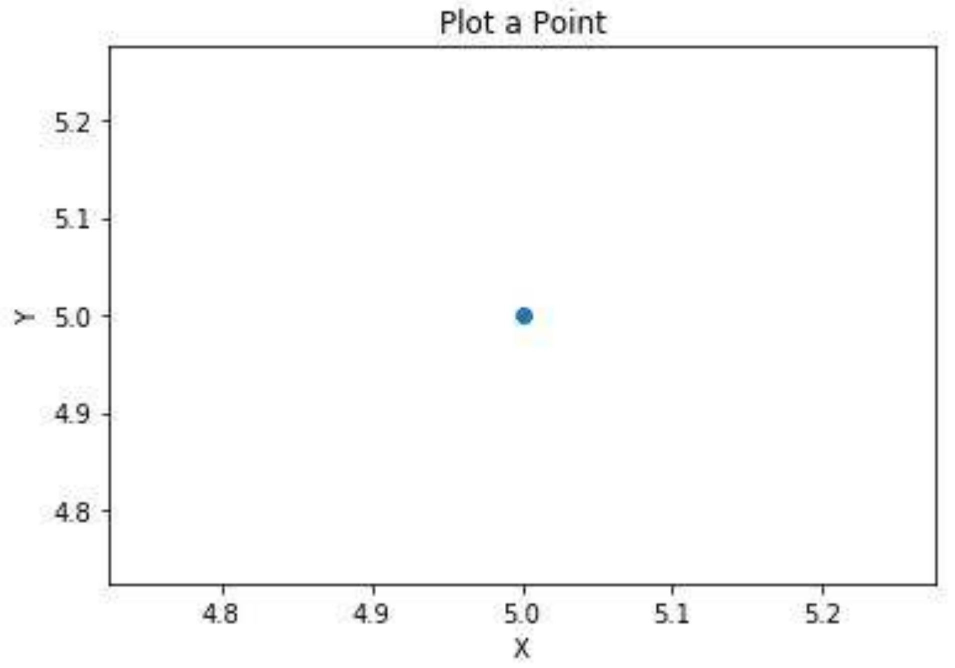


Why Build Visuals?

- For exploratory data analysis
- Communicate data clearly
- Share unbiased representation of data
- A picture is worth a thousand words ☺

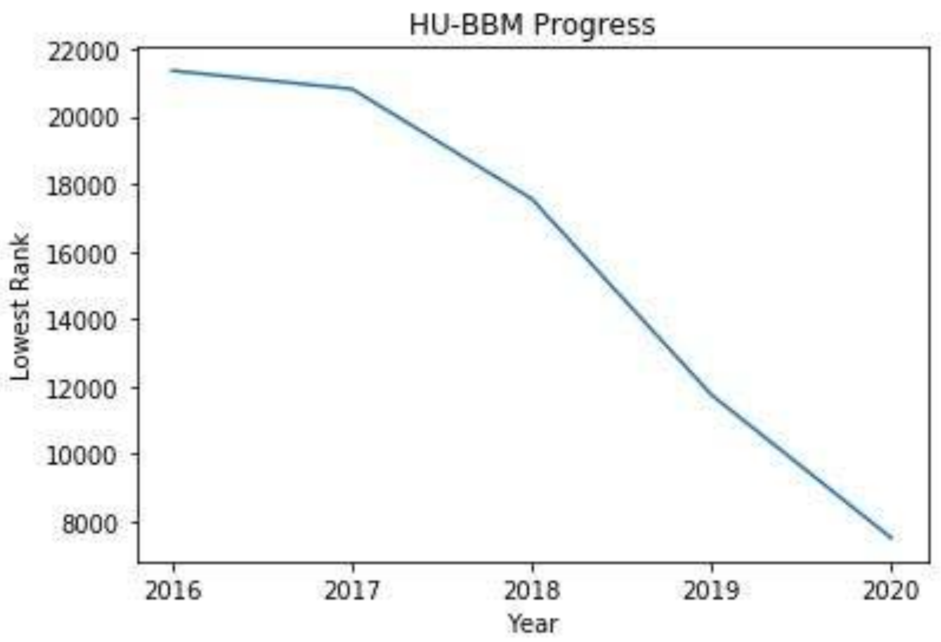
Make a Simple Plot

```
import matplotlib.pyplot as plt

plt.plot(5, 5, 'o')

plt.title("Plot a Point")

plt.xlabel("X")
plt.ylabel("Y")

plt.show()
```



Plot a Simple Line

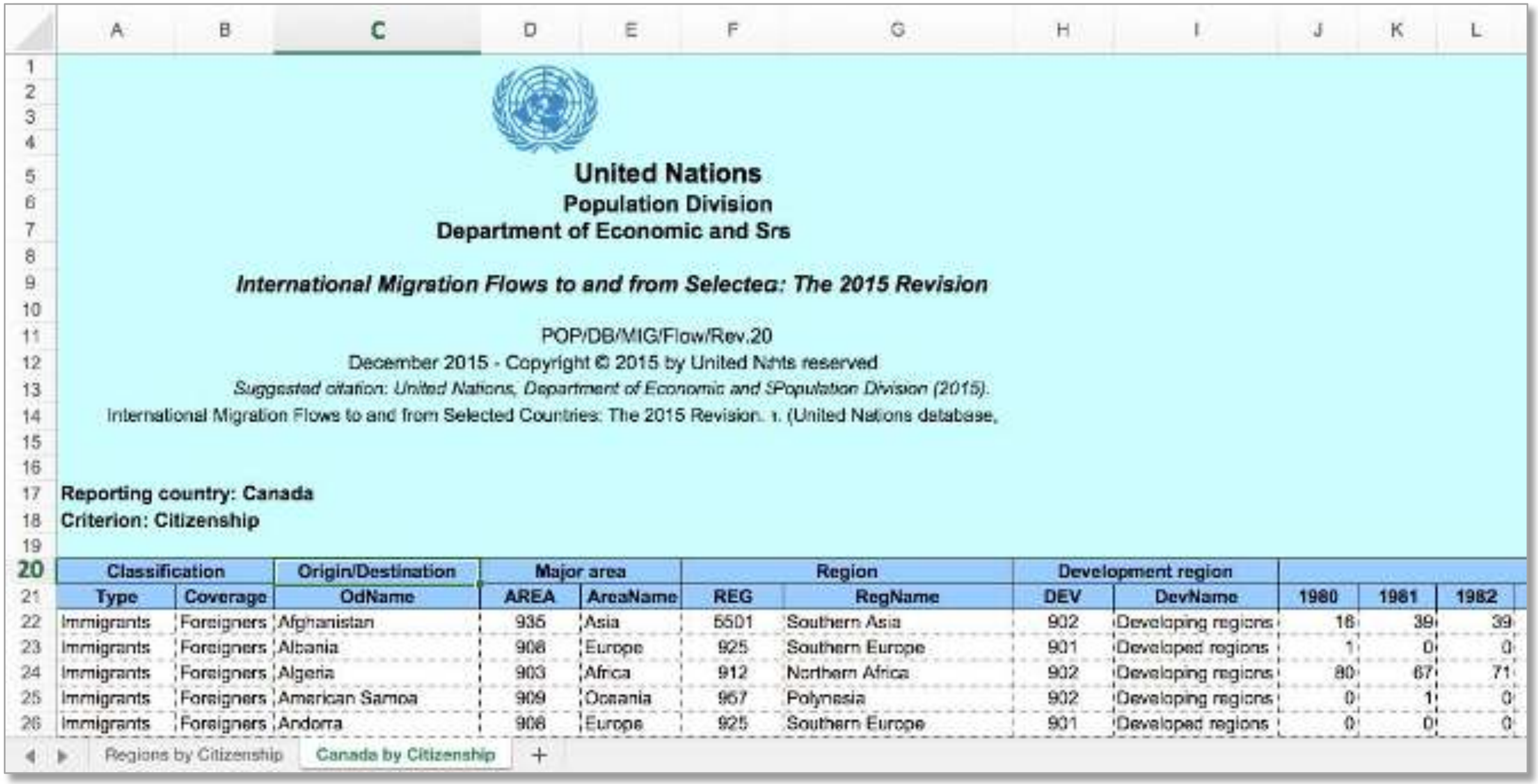
```
import matplotlib.pyplot as plt

year = ['2016', '2017', '2018', '2019', '2020']
lowest_rank = [21358, 20816, 17555, 11743, 7500]

plt.plot(year, lowest_rank)

plt.title(”HU–BBM Progress")
plt.xlabel('Year')
plt.ylabel('Lowest Rank')

plt.show()
```



Dataset to Use for the Rest of This Section

- The Population Division of the United Nations compiled data pertaining to 45 countries.
- For each country, annual data on the flows of international migrants is reported in addition to other metadata.
- We will work with data on Canada.
- You can get the original data at:
  - https://www.un.org/en/development/desa/population/migration/data/empirical2/migrationflows.asp#
  - It is also available at bbm101's web page.

Immigration Data to Canada



United Nations
Population Division
Department of Economic and Srs

International Migration Flows to and from Selectea: The 2015 Revision

POP/DB/MIG/Flow/Rev.20
December 2015 - Copyright © 2015 by United Nhts reserved
Suggested citation: United Nations, Department of Economic and SPopulation Division (2015).
International Migration Flows to and from Selected Countries: The 2015 Revision. 1. (United Nations database,

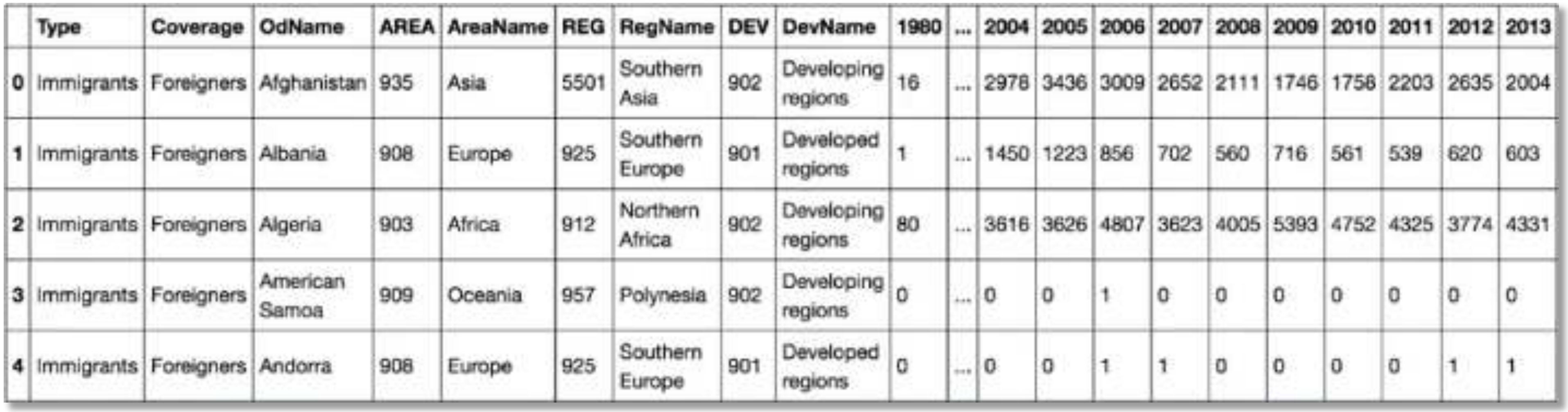
Reporting country: Canada
Criterion: Citizenship

Classification		Origin/Destination	Major area		Region		Development region				
Type	**Coverage**	**OdName**	**AREA**	**AreaName**	**REG**	**RegName**	**DEV**	**DevName**	**1980**	**1981**	**1982**
Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	39	39
Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	0	0
Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	67	71
Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	1	0
Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	0	0

Regions by Citizenship | Canada by Citizenship | +

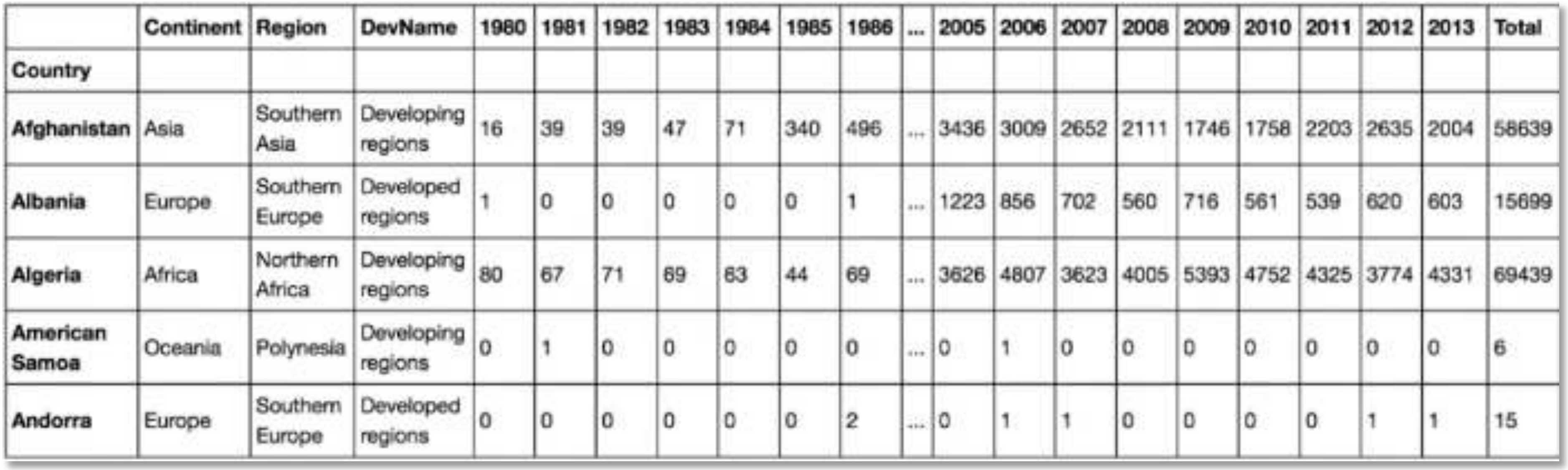
Read Data into Pandas Dataframe

```
df = pd.read_excel
   ('http://www.un.org/…/Canada.xlsx',
      sheet_name='Canada by Citizenship',
      skiprows=range(20),
      skip_footer=2)

print(df.head())
```



	Type	Coverage	OdName	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
0	Immigrants	Foreigners	Afghanistan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	2004
1	Immigrants	Foreigners	Albania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620	603
2	Immigrants	Foreigners	Algeria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	4331
3	Immigrants	Foreigners	American Samoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0	0
4	Immigrants	Foreigners	Andorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1	1

After Little Preprocessing

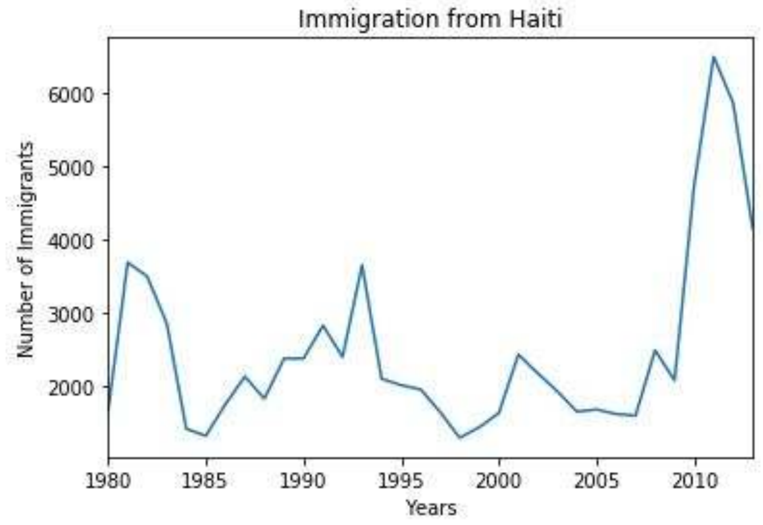


	Continent	Region	DevName	1980	1981	1982	1983	1984	1985	1986	...	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
Country																					
Afghanistan	Asia	Southern Asia	Developing regions	16	39	39	47	71	340	496	...	3436	3009	2652	2111	1746	1758	2203	2635	2004	58639
Albania	Europe	Southern Europe	Developed regions	1	0	0	0	0	0	1	...	1223	856	702	560	716	561	539	620	603	15699
Algeria	Africa	Northern Africa	Developing regions	80	67	71	69	63	44	69	...	3626	4807	3623	4005	5393	4752	4325	3774	4331	69439
American Samoa	Oceania	Polynesia	Developing regions	0	1	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	6
Andorra	Europe	Southern Europe	Developed regions	0	0	0	0	0	0	2	...	0	1	1	0	0	0	0	1	1	15

Line Plots

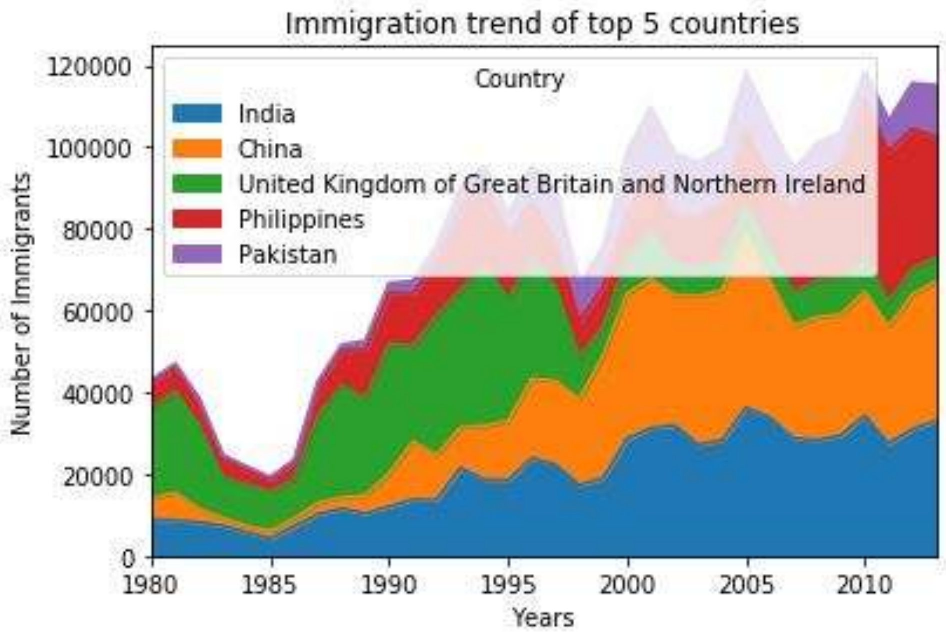
A line plot displays information as a series of data points called 'markers' connected by straight line segments.

```
years = list(range(1980, 2014))
df_canada.loc['Haiti', years].plot(kind = 'line')

plt.title('Immigration from Haiti')
plt.xlabel('Years')
plt.ylabel('Number of Immigrants')

plt.show()
```



Area Plots

Commonly used to represent cumulated totals using numbers or percentages over time.

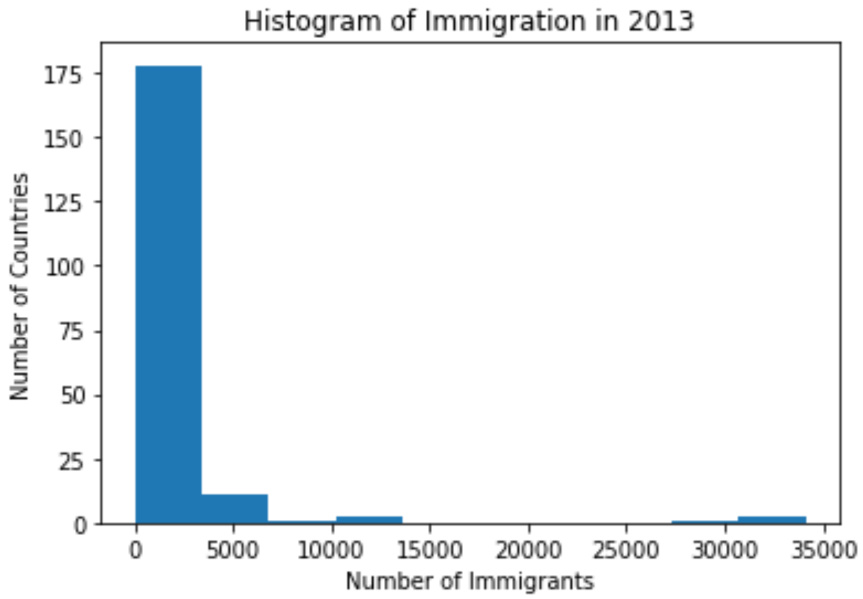


```
df_canada.sort_values(['Total'], ascending=False,
                       axis=0, inplace=True)
df_top5 = df_canada.head()
df_top5 = df_top5[years].transpose()
df_top5.plot(kind='area')

plt.title('Immigration trend of top 5 countries')
plt.xlabel('Years')
plt.ylabel('Number of Immigrants')

plt.show()
```

Histogram

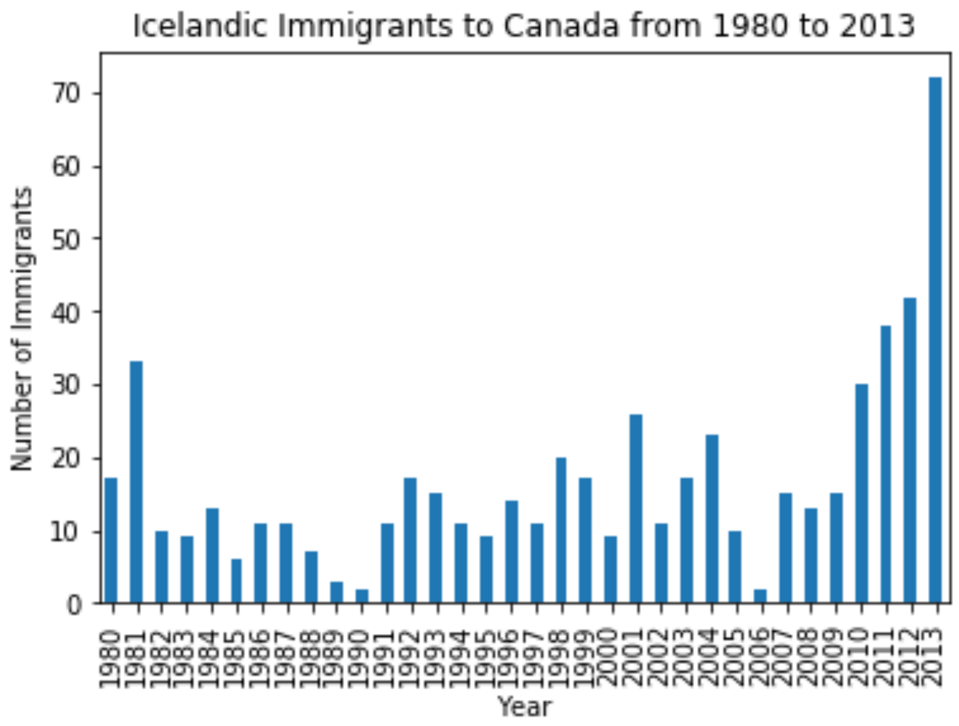
Histogram is a way of representing the frequency distribution of a variable.

```
df_canada[2013].plot(kind='hist')

plt.title('Histogram of Immigration in 2013')
plt.xlabel('Number of Immigrants')
plt.ylabel('Number of Countries')

plt.show()
```



Bar Chart

Unlike a histogram, a bar chart is commonly used to compare the values of a variable at a given point.

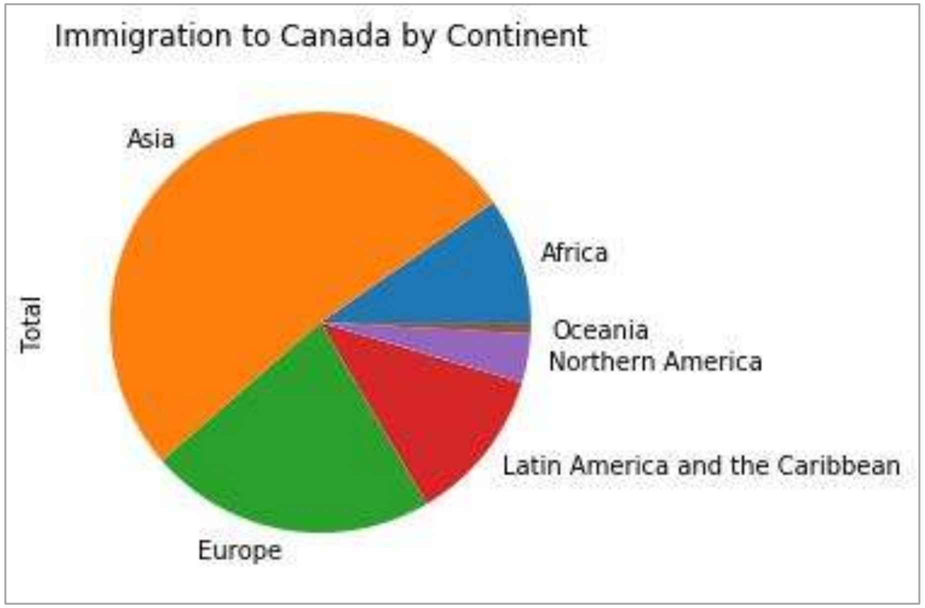


```
df_iceland = df_canada.loc['Iceland', years]
df_iceland.plot(kind='bar')

plt.title('Icelandic Immigrants to Canada from 1980 to 2013')
plt.xlabel('Year')
plt.ylabel('Number of Immigrants')

plt.show()
```

Pie Chart

A pie chart is a circular statistical graphic divided into slices to illustrate numerical proportion.

```
df_continents = df_canada.groupby('Continent', axis=0).sum()
df_continents['Total'].plot(kind='pie')

plt.title('Immigration to Canada by Continent')

plt.show()
```



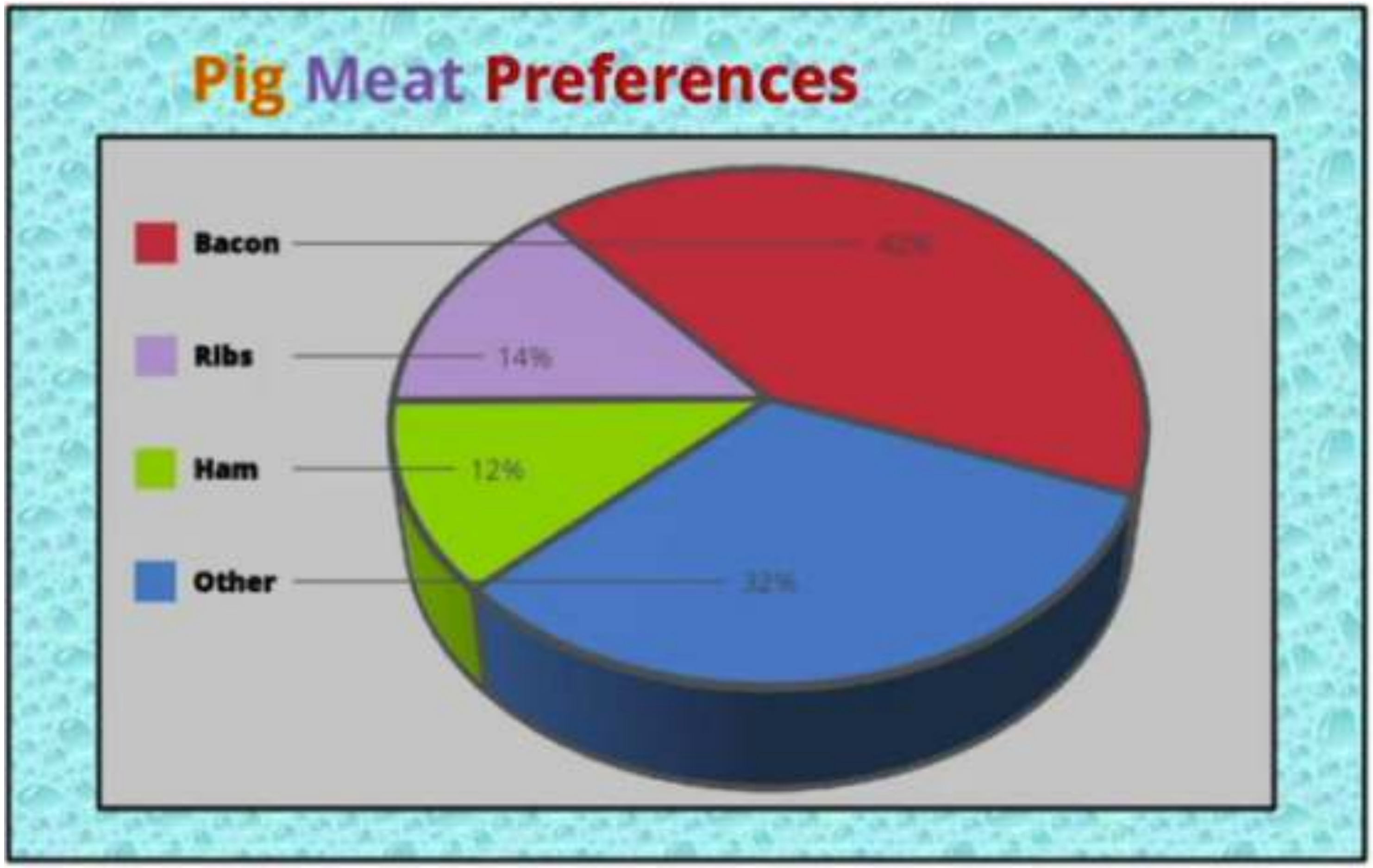
Best Practices, before we close this section

- When creating a visual, always remember:
  - Less is more effective
  - Less is more attractive
  - Less is more impactive

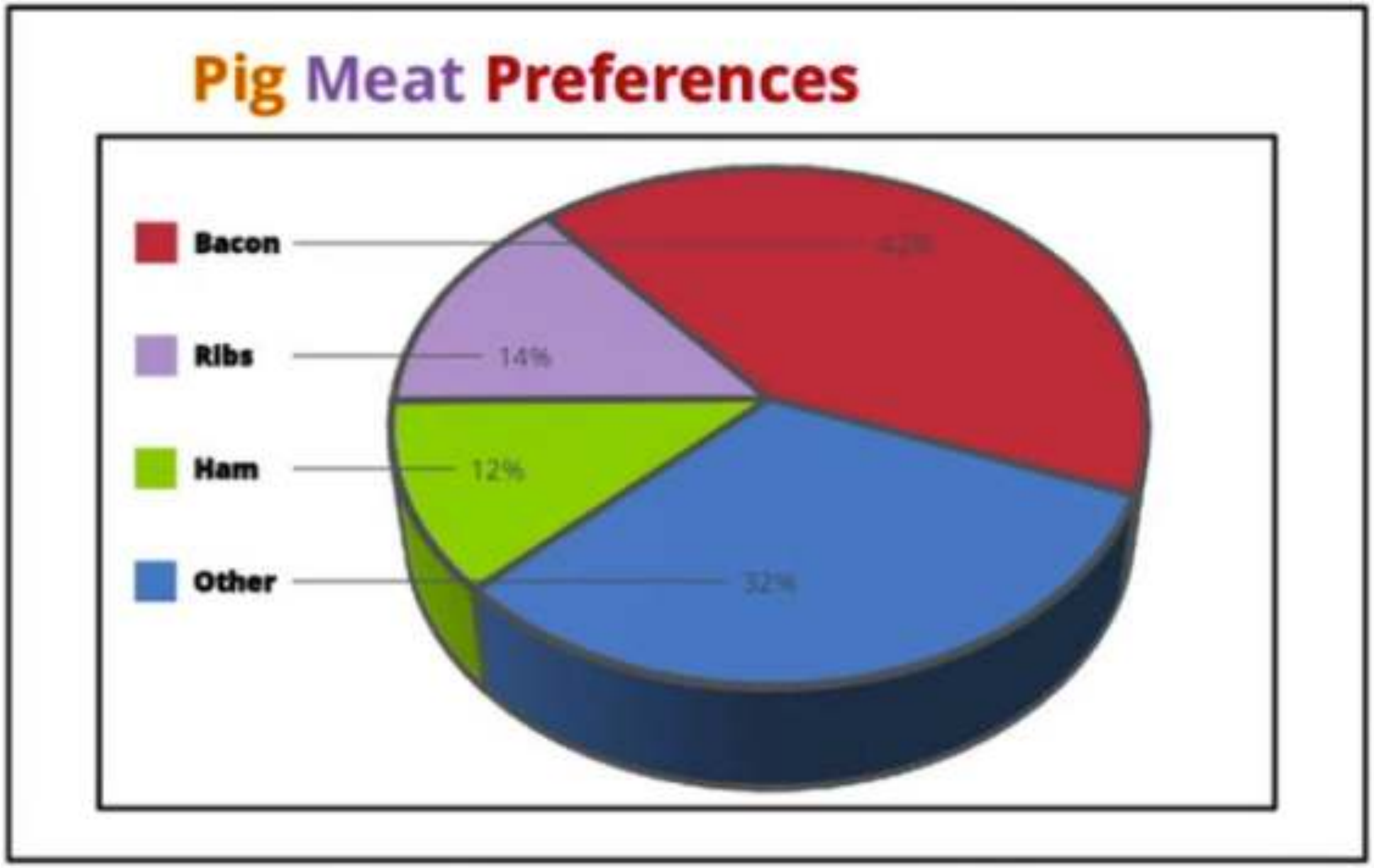
DARKHORSE ANALYTICS

www.darkhorseanalytics.com/blog/salvaging-the-pie

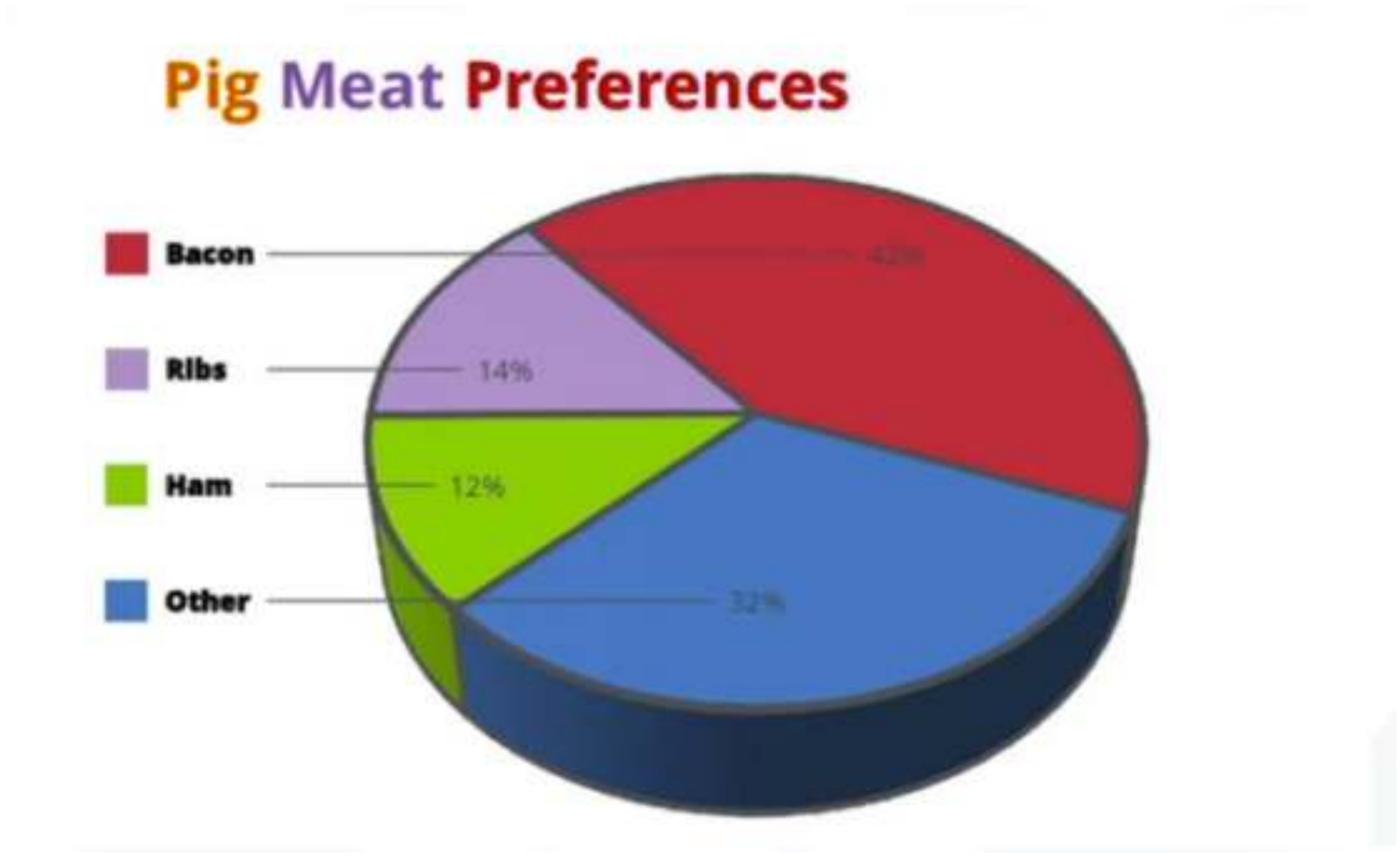
Salvaging the Pie Chart



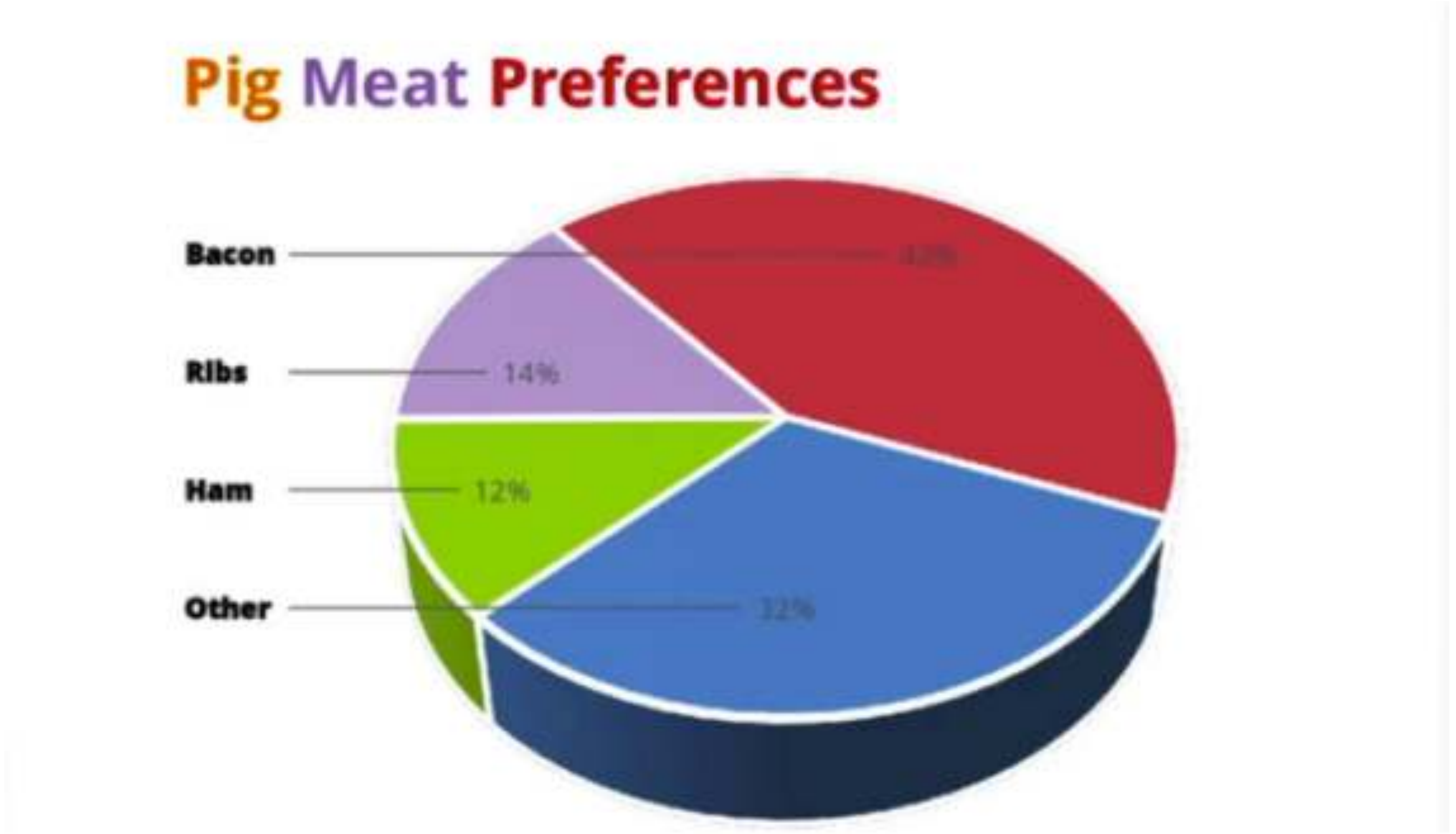
Remove Background



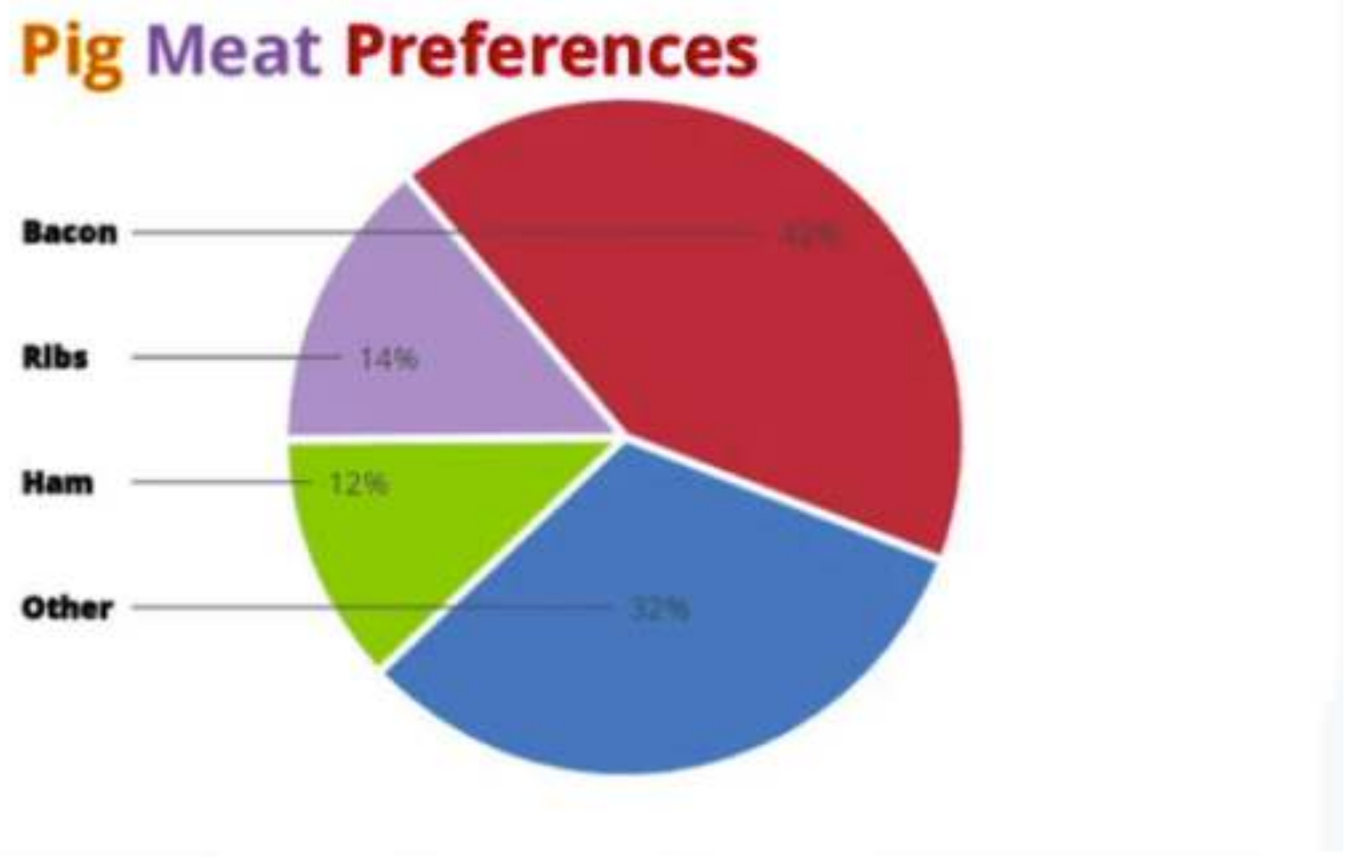
Remove Borders



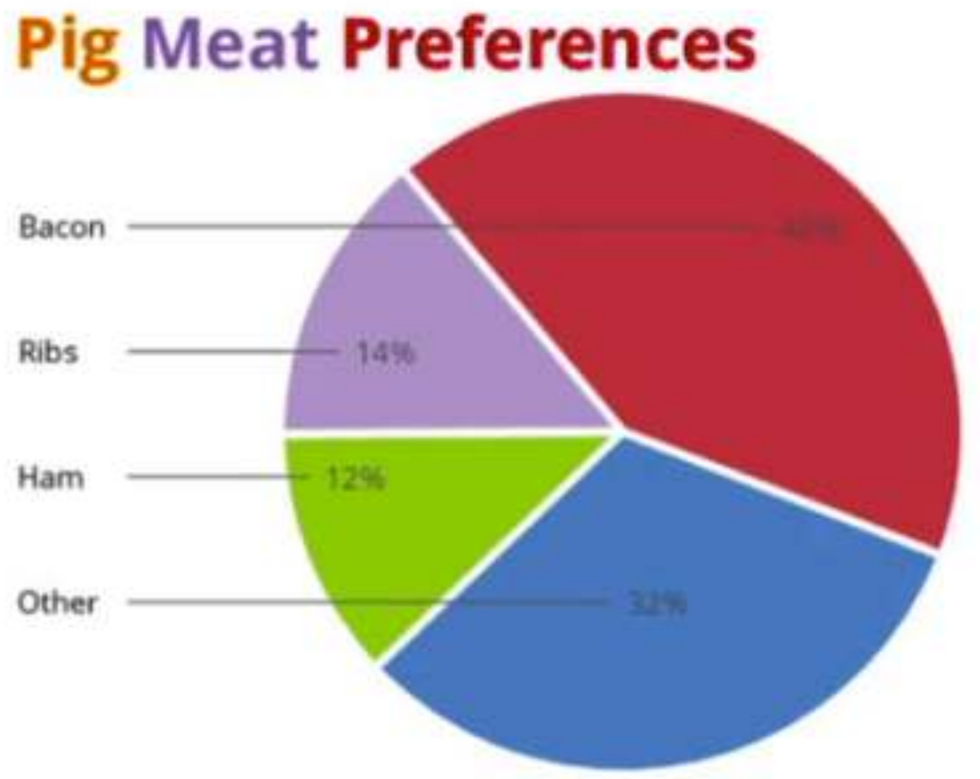
Remove Redundant Legend



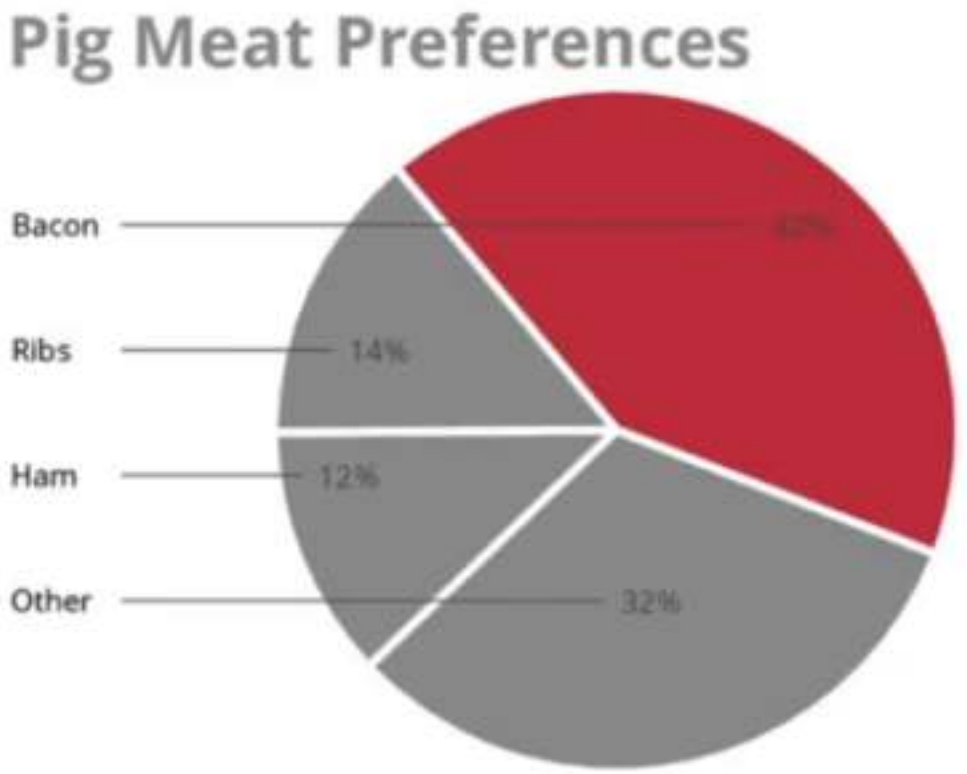
Remove 3D



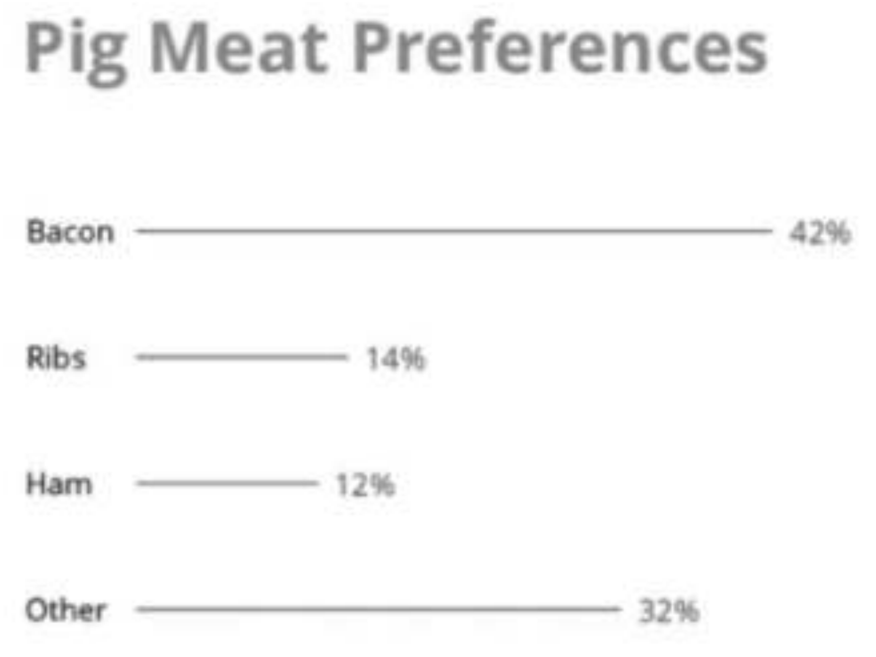
Remove Text Bolding



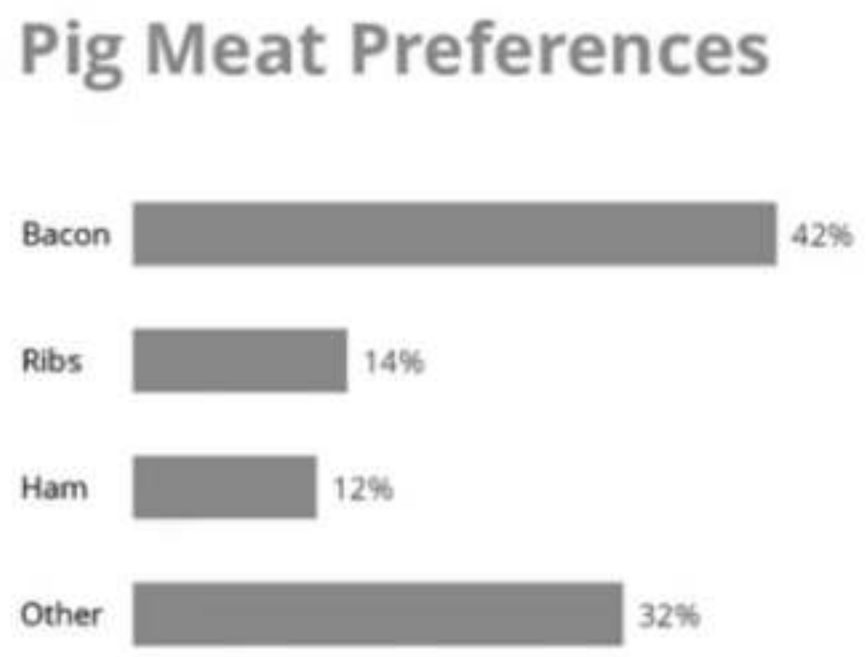
Reduce Color



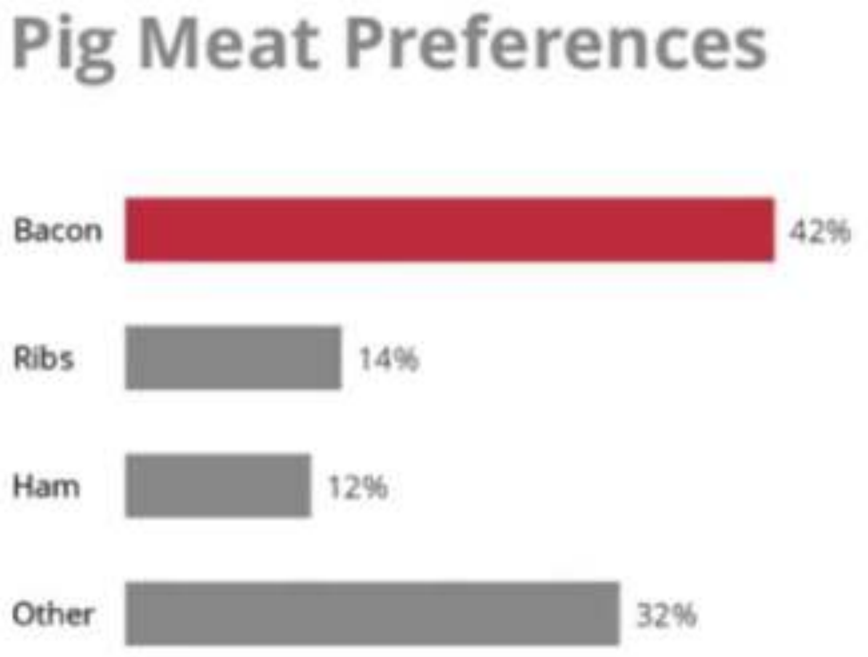
Remove Wedges



Thicken Lines

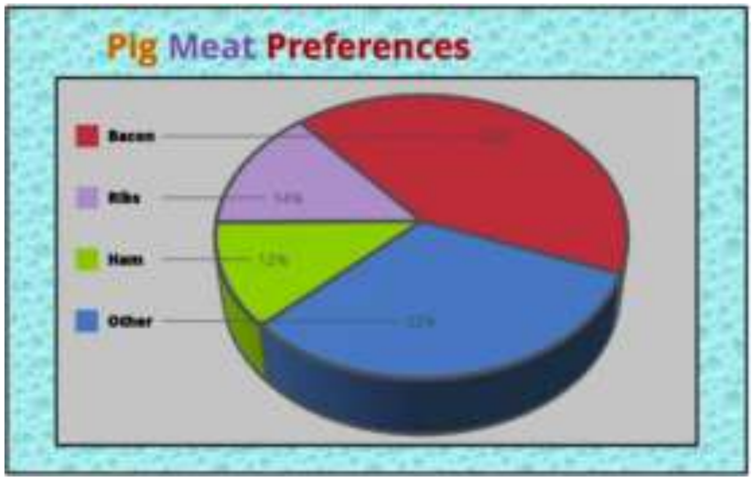


Emphasize Bacon

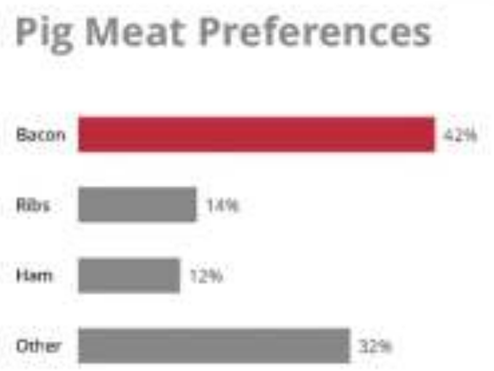


Comparison

Before



After



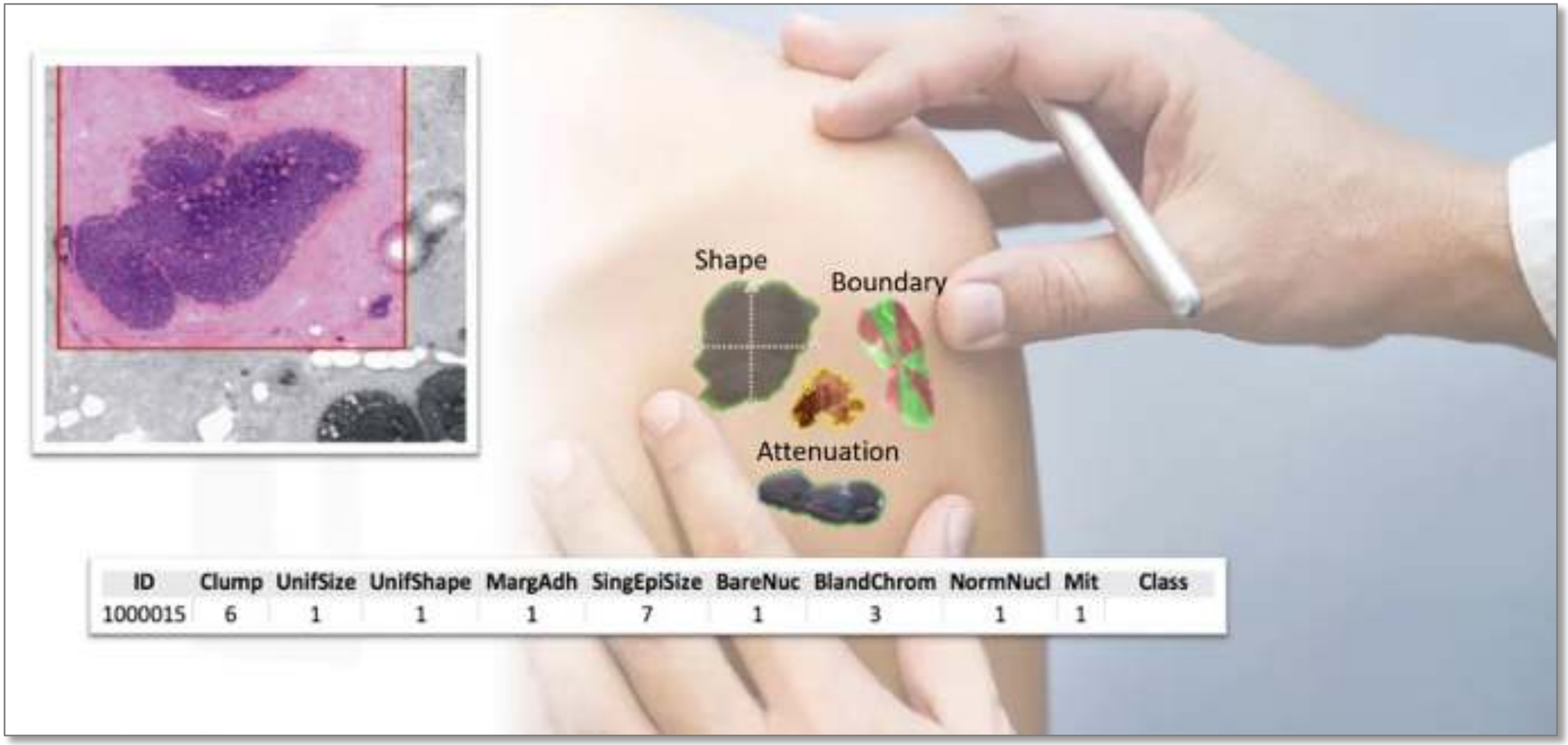
Lecture Overview

- Introduction to Data Science
  - Data, Data Science, Data Scientist…
- Python Libraries to Analyse Data
  - Pandas
  - Numpy
  - Matplotlib
- Your -Probably the- First Data Science Project

Your -Probably the- First Data Science Project

- In this small project, you will try to detect breast cancer.
- Base on the given data, you will predict if a cell is benign or malignant.
- Before that, let’s talk about machine learning little bit.

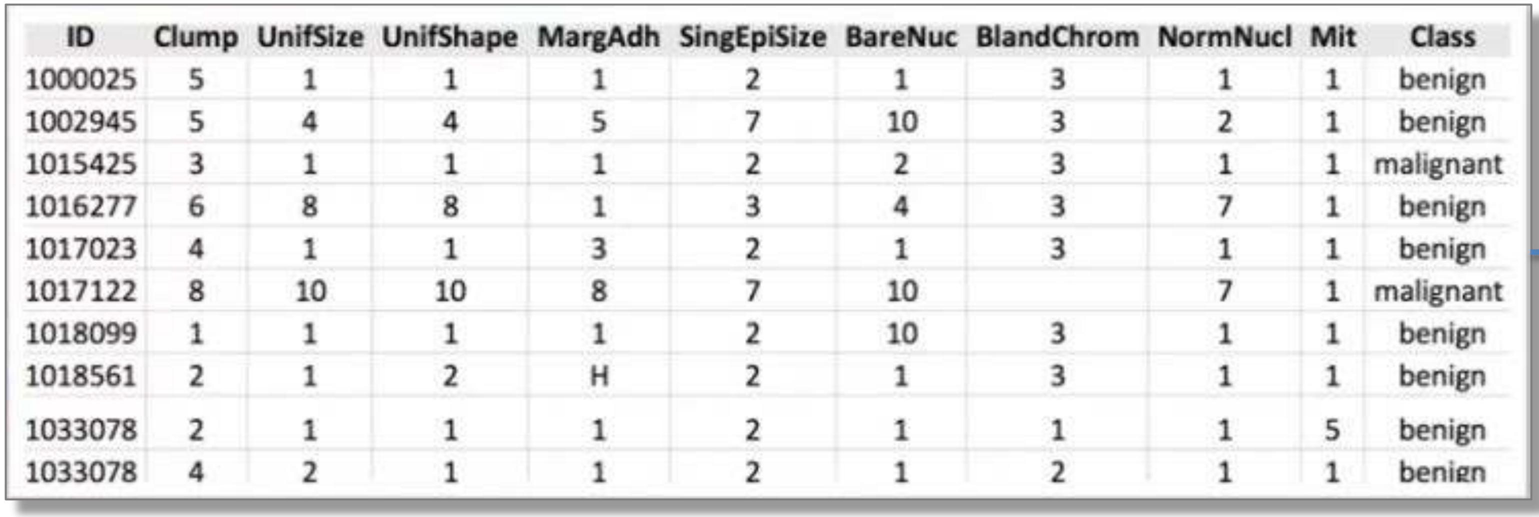
Is This a Benign or Malignant Cell?



ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000015	6	1	1	1	7	1	3	1	1	

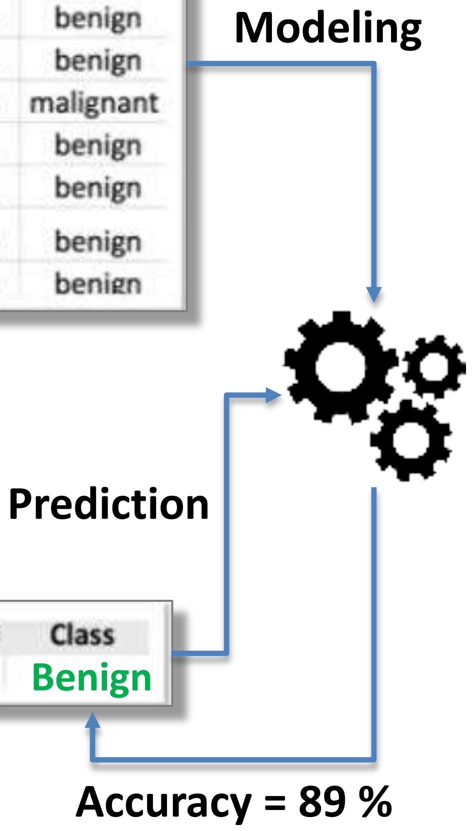
What is Machine Learning?

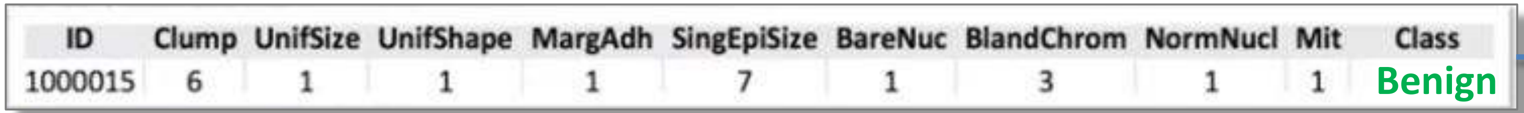
A dataset containing characteristics of human cell samples extracted from patients.



ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

Analysis shows that many of the characteristics differed significantly between benign and malignant samples.





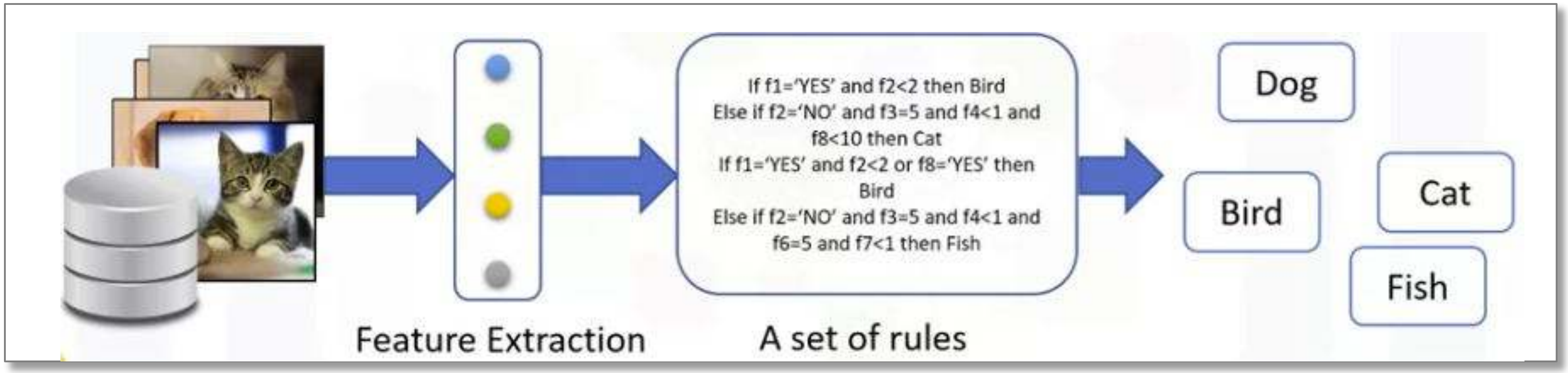
ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000015	6	1	1	1	7	1	3	1	1	Benign

Those characteristics can be used to predict whether a new sample might be benign or malignant.

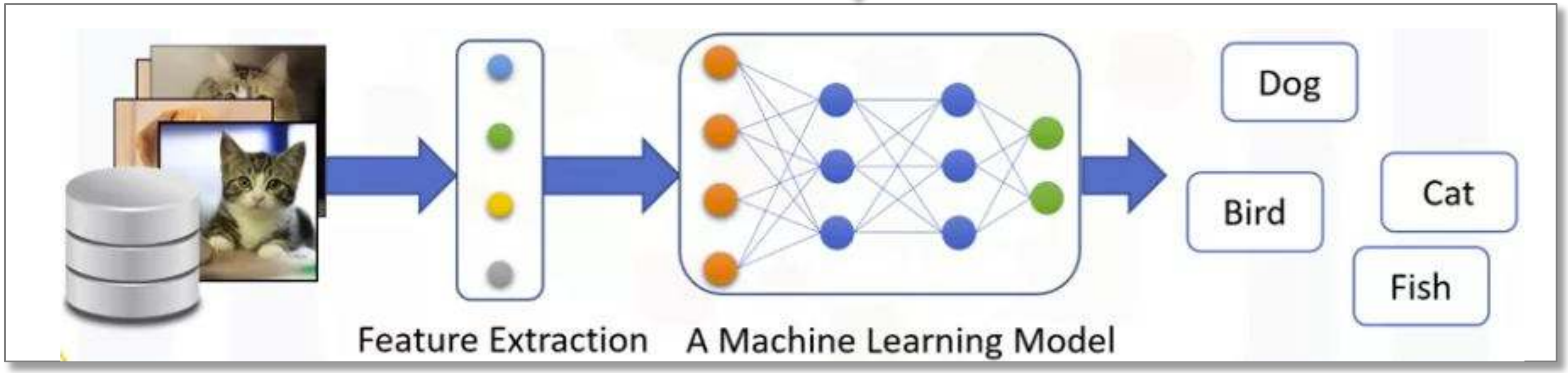
Formal Definition of Machine Learning

Machine learning is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed."

Computers to Find Hidden Insights


If f1='YES' and f2<2 then Bird
Else if f2='NO' and f3=5 and f4<1 and f8<10 then Cat
If f1='YES' and f2<2 or f8='YES' then Bird
Else if f2='NO' and f3=5 and f4<1 and f6=5 and f7<1 then Fish
Dog
Bird
Cat
Fish
Feature Extraction
A set of rules

Needs a lot of rules, highly dependent on the current dataset, and not generalized enough to detect out-of-sample cases.



Machine learning model looks at all the feature sets, and their corresponding type of animals, and learns the pattern of each animal. Without being **explicitly** programmed.

73

Python Libraries for Machine Learning

python™

NumPy

matplotlib

scikit learn

pandas

SciPy

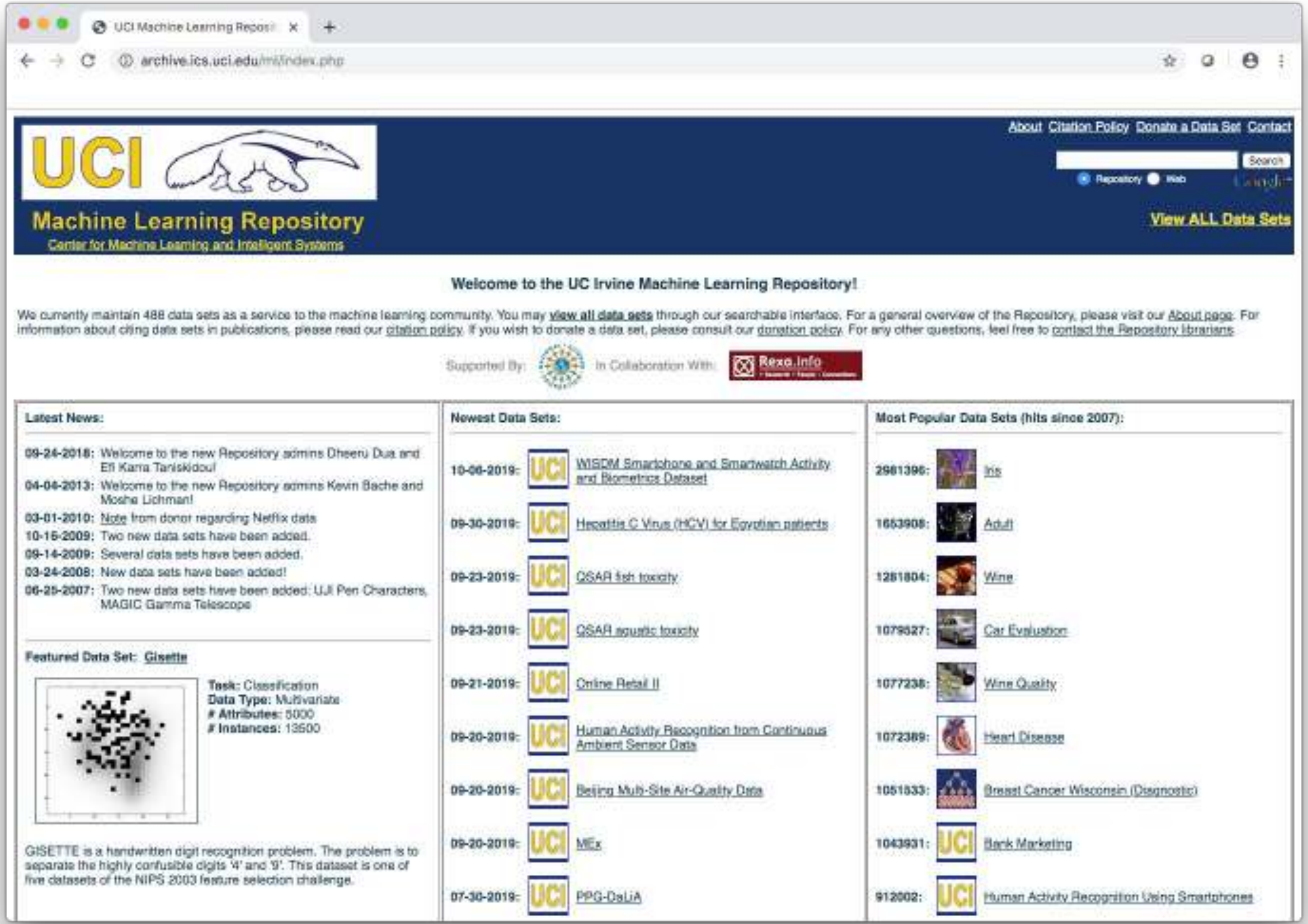
Methodology for Machine Learning Applications

- Obtain Data
- Understand, Clean and Transform Data
- Build a Machine Learning Model
- Train/Test Your Model
- Predict

Obtain Data

- Collect data by yourself
- Borrow from friends
- Generate Synthetically
- Google it
- …

Beware of “Kişisel Verilerin Korunması Kanunu”
www.kvkk.gov.tr


UCI Machine Learning Reposi
archive.ics.uci.edu/ml/index.php
About Citation Policy Donate a Data Set Contact
Search
Repository Web
View ALL Data Sets
UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 488 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult our donation policy. For any other questions, feel free to contact the Repository librarians.

Supported By: In Collaboration With: Rexa.info

Latest News:

09-24-2018:	Welcome to the new Repository admins Dheeru Dua and Efi Karra Taniskidou!
04-04-2013:	Welcome to the new Repository admins Kevin Bache and Moshe Lichman!
03-01-2010:	Note from donor regarding Netflix data
10-16-2009:	Two new data sets have been added.
09-14-2009:	Several data sets have been added.
03-24-2008:	New data sets have been added!
06-25-2007:	Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope

Featured Data Set: Gisette

Task: Classification
Data Type: Multivariate
# Attributes: 5000
# Instances: 13500

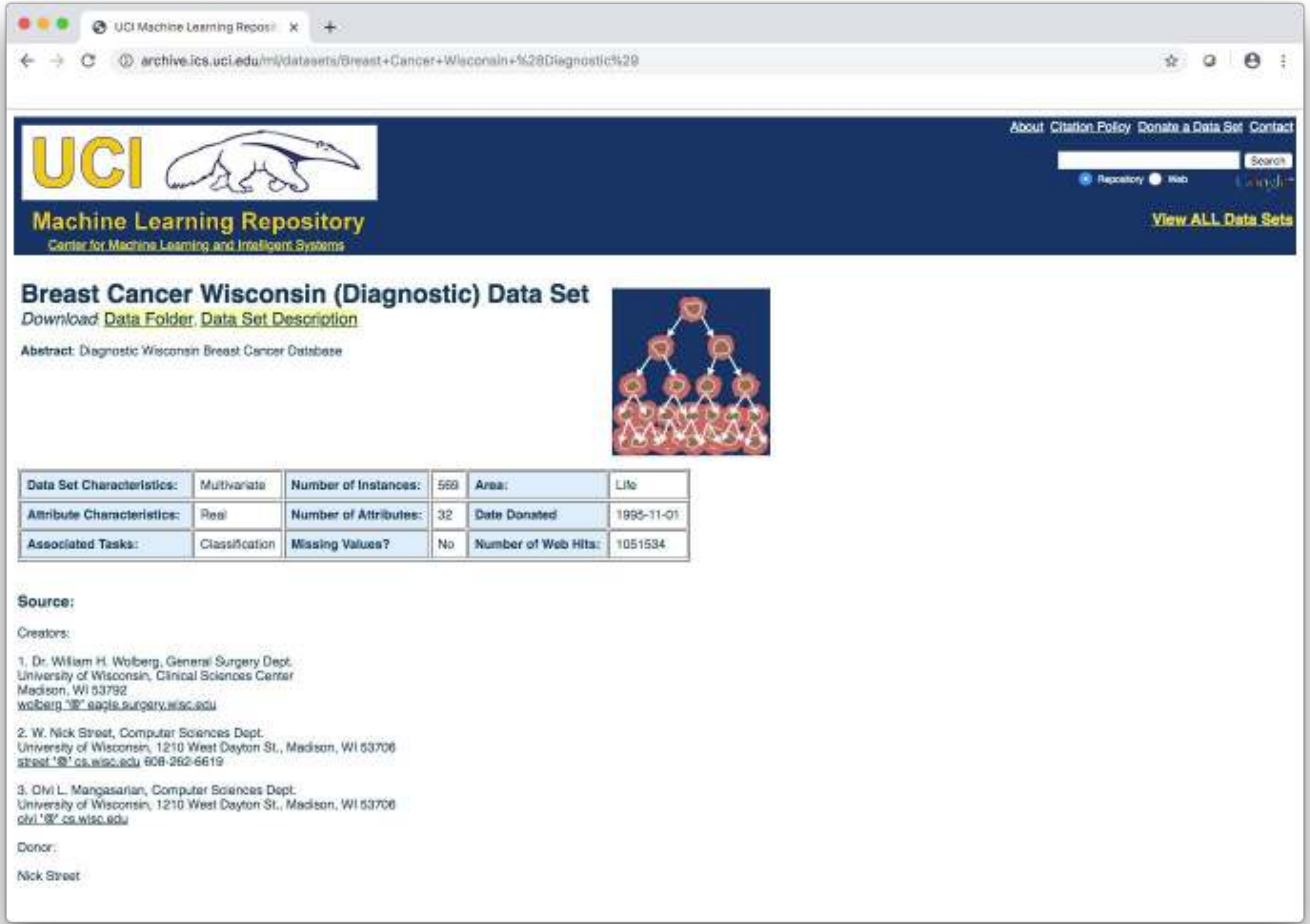
GISETTE is a handwritten digit recognition problem. The problem is to separate the highly confusible digits '4' and '9'. This dataset is one of five datasets of the NIPS 2003 feature selection challenge.

Newest Data Sets:

10-06-2019:	WISDM Smartphone and Smartwatch Activity and Biometrics Dataset
09-30-2019:	Hepatitis C Virus (HCV) for Egyptian patients
09-23-2019:	QSAR fish toxicity
09-23-2019:	QSAR aquatic toxicity
09-21-2019:	Online Retail II
09-20-2019:	Human Activity Recognition from Continuous Ambient Sensor Data
09-20-2019:	Beijing Multi-Site Air-Quality Data
09-20-2019:	MEx
07-30-2019:	PPG-DaLiA

Most Popular Data Sets (hits since 2007):

2981396:	Iris
1653908:	Adult
1281804:	Wine
1079527:	Car Evaluation
1077238:	Wine Quality
1072389:	Heart Disease
1051533:	Breast Cancer Wisconsin (Diagnostic)
1043931:	Bank Marketing
912002:	Human Activity Recognition Using Smartphones



UCI Machine Learning Reposi
archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
About Citation Policy Donate a Data Set Contact
Search
Repository Web
Machine Learning Repository
Center for Machine Learning and Intelligent Systems
View ALL Data Sets

Breast Cancer Wisconsin (Diagnostic) Data Set

Download: Data Folder, Data Set Description

Abstract: Diagnostic Wisconsin Breast Cancer Database

Data Set Characteristics:	Multivariate	**Number of Instances:**	569	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	32	**Date Donated**	1995-11-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1051534

Source:

Creators:

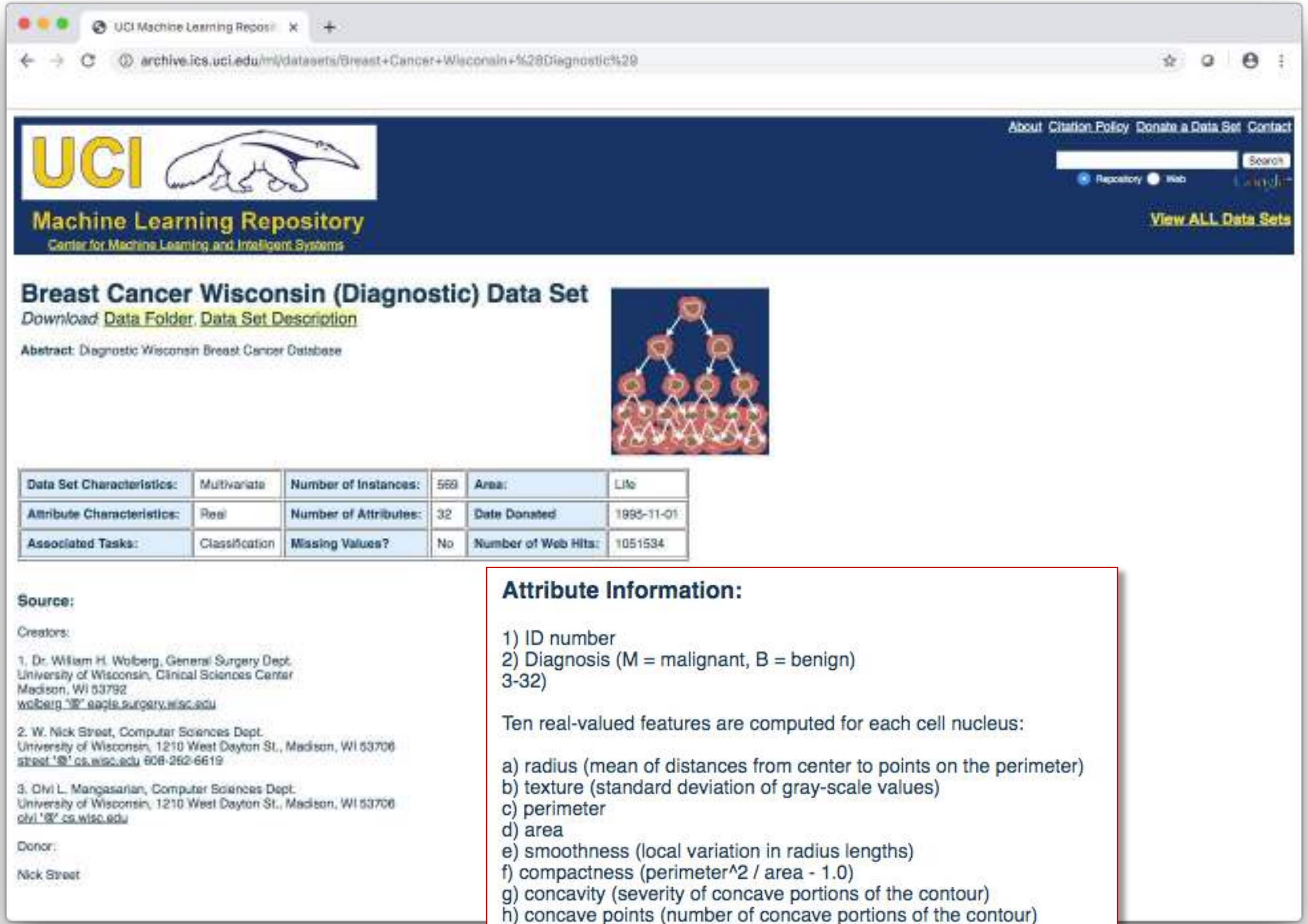
1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu

2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619

3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

Nick Street


UCI Machine Learning Reposi
archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems
About Citation Policy Donate a Data Set Contact
Search
Repository Web
View ALL Data Sets

Breast Cancer Wisconsin (Diagnostic) Data Set

Download: Data Folder, Data Set Description

Abstract: Diagnostic Wisconsin Breast Cancer Database

Data Set Characteristics:	Multivariate	**Number of Instances:**	569	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	32	**Date Donated**	1995-11-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1051534

Source:

Creators:

1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu

2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619

3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

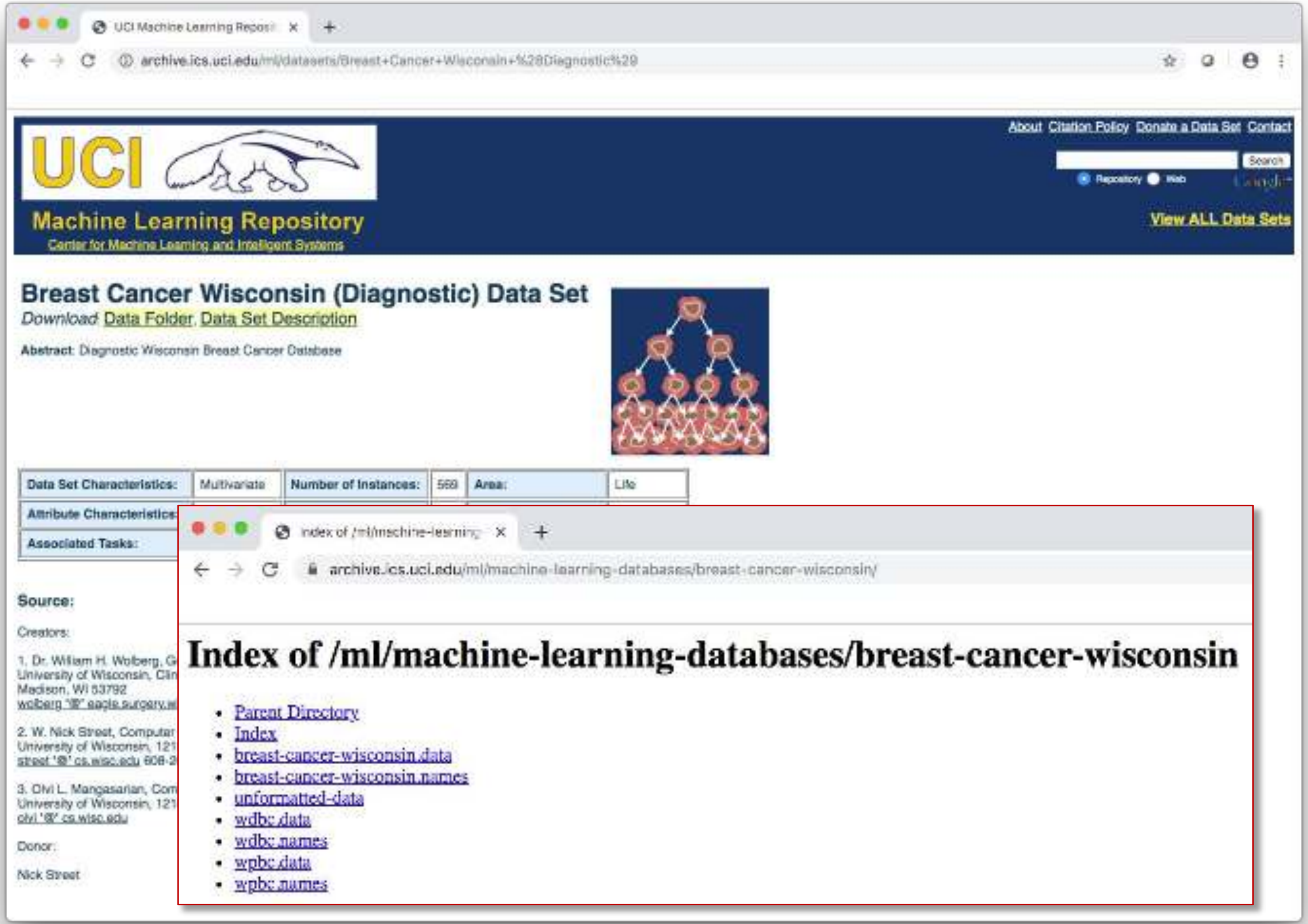
Nick Street

Attribute Information:

1) ID number
2) Diagnosis (M = malignant, B = benign)
3-32)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)



UCI Machine Learning Reposi

archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems
About Citation Policy Donate a Data Set Contact
Search
Repository Web
View ALL Data Sets

Breast Cancer Wisconsin (Diagnostic) Data Set

Download: Data Folder, Data Set Description

Abstract: Diagnostic Wisconsin Breast Cancer Database

Data Set Characteristics:	Multivariate	**Number of Instances:**	569	**Area:**	Life
Attribute Characteristics:					
Associated Tasks:					

Source:

Creators:

1. Dr. William H. Wolberg, G
University of Wisconsin, Clin
Madison, WI 53792
wolberg '@' eagle.surgery.w

2. W. Nick Street, Computer
University of Wisconsin, 121
street '@' cs.wisc.edu 608-2

3. Olvi L. Mangasarian, Com
University of Wisconsin, 121
olvi '@' cs.wisc.edu

Donor:

Nick Street

Index of /ml/machine-learning

archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/

Index of /ml/machine-learning-databases/breast-cancer-wisconsin

- Parent Directory
- Index
- breast-cancer-wisconsin.data
- breast-cancer-wisconsin.names
- unformatted-data
- wdbc.data
- wdbc.names
- wpbc.data
- wpbc.names

Understand, Clean and Transform Data

- Determine important features
- Look for correlations
- Remove duplicated data
- Handle missing data
  - Remove rows that contain missing data
  - Impute missing values somehow
- Transform data when necessary
  - e.g. convert categorical data into numbers

Build a Machine Learning Model

- Determine the type of the task at hand
  - Classification, regression, clustering
- Choose a proper algorithm to build the model

Train/Test Your Model

- Divide your data into train and test sets
  - Typically 75/25% split
- Train your model by using the train data
- Test your model by using the test data
- If the performance is not satisfying
  - Tune your model by switching parameters
  - Pick another algorithm if tuning does not help

Predict

- Use new and unlabeled data to predict
  - It is typically not 100% accurate
- Hopefully your model is accurate enough to catch problems as early as possible

Short Demo: Breast Cancer Detection

- Look at course's web page for pdf handouts and the Jupyter Notebook.
  - https://web.cs.hacettepe.edu.tr/~bbm101/