**Functions**

1. Making numpy array as matrices and vectors

import numpy as np

vector = np.array([10, 20, 30])

matrix = np.array([[5, 10, 15], [20, 25, 30], [35, 40, 45]])

1. Dimension of array

vector = numpy.array([1, 2, 3, 4])

print(vector.shape)

1. Reading a dataset in Numpy

world\_alcohol = np.genfromtxt("world\_alcohol.csv", delimiter=",")

print(type(world\_alcohol))

world\_alcohol = np.genfromtxt("world\_alcohol.csv", delimiter=",", dtype="U75", skip\_header=1)

print(world\_alcohol)

world\_alcohol.dtype

1. Selecting all rows and specific columns

countries = world\_alcohol[:,2]

alcohol\_consumption = world\_alcohol[:,4]

# slice of one column

first\_two\_columns = world\_alcohol[:,0:2]

first\_ten\_years = world\_alcohol[0:10,0]

first\_ten\_rows = world\_alcohol[0:10,:]

# Slice of both row and column

first\_twenty\_regions = world\_alcohol[0:20,1:3]

1. Computatio with Numpy

years\_1984 = (world\_alcohol[:,0] == "1984")

countries\_canada = (world\_alcohol[:,2] == "Canada")

#Extracting rows with conditions and multiple conditions

country\_is\_algeria = world\_alcohol[:,2] == "Algeria"

country\_algeria = world\_alcohol[country\_is\_algeria,:]

is\_algeria\_and\_1986 = (world\_alcohol[:,0] == "1986") & (world\_alcohol[:,2] == "Algeria")

rows\_with\_algeria\_and\_1986 = world\_alcohol[is\_algeria\_and\_1986,:]

#Changing row values with other values

world\_alcohol[:,0][world\_alcohol[:,0] == '1986'] = '2014'

world\_alcohol[:,3][world\_alcohol[:,3] == 'Wine'] = 'Grog'

## Filling null values

is\_value\_empty = world\_alcohol[:,4] == ''

world\_alcohol[is\_value\_empty, 4] = '0'

#We can convert the data type of an array with the [astype()](http://docs.scipy.org/doc/numpy-1.10.1/reference/generated/numpy.ndarray.astype.html) method.

alcohol\_consumption = world\_alcohol[:,4]

alcohol\_consumption = alcohol\_consumption.astype(float)

#Data row operations

total\_alcohol = alcohol\_consumption.sum()

average\_alcohol = alcohol\_consumption.mean()

## All Operations

is\_canada\_1986 = (world\_alcohol[:,2] == "Canada") & (world\_alcohol[:,0] == '1986')

canada\_1986 = world\_alcohol[is\_canada\_1986,:]

canada\_alcohol = canada\_1986[:,4]

empty\_strings = canada\_alcohol == ''

canada\_alcohol[empty\_strings] = "0"

canada\_alcohol = canada\_alcohol.astype(float)

total\_canadian\_drinking = canada\_alcohol.sum()

totals = {}

is\_year = world\_alcohol[:,0] == "1989"

year = world\_alcohol[is\_year,:]

for country in countries:

is\_country = year[:,2] == country

country\_consumption = year[is\_country,:]

alcohol\_column = country\_consumption[:,4]

is\_empty = alcohol\_column == ''

alcohol\_column[is\_empty] = "0"

alcohol\_column = alcohol\_column.astype(float)

totals[country] = alcohol\_column.sum()

print(totals)

highest\_value = 0

highest\_key = None

for country in totals:

consumption = totals[country]

if highest\_value < consumption:

highest\_value = consumption

highest\_key = country

PANDAS

1. Data manipulation with pandas

import pandas

food\_info = pandas.read\_csv("food\_info.csv")

#list of all column names

col\_names = food\_info.columns.tolist()

print(col\_names)

print(food\_info.head(3))

1. Operating on Columns

div\_1000 = food\_info["Iron\_(mg)"] / 1000

add\_100 = food\_info["Iron\_(mg)"] + 100

sub\_100 = food\_info["Iron\_(mg)"] - 100

mult\_2 = food\_info["Iron\_(mg)"]\*2

sodium\_grams = food\_info["Sodium\_(mg)"] / 1000

sugar\_milligrams = food\_info["Sugar\_Tot\_(g)"] \* 1000

water\_energy = food\_info["Water\_(g)"] \* food\_info["Energ\_Kcal"]

print(water\_energy[0:5])

# The largest value in the "Energ\_Kcal" column.

max\_calories = food\_info["Energ\_Kcal"].max()

# Divide the values in "Energ\_Kcal" by the largest value to normalize.

normalized\_calories = food\_info["Energ\_Kcal"] / max\_calories

1. Add processed column to the DataFrame as a new column instead

food\_info["Normalized\_Protein"] = normalized\_protein

food\_info["Normalized\_Fat"] = normalized\_fat

1. DataFrame objects have a [sort\_values() method](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.sort_values.html) that we can use to sort the entire DataFrame.

food\_info.sort\_values("Sodium\_(mg)")

# Sorts the DataFrame in-place, rather than returning a new DataFrame.

food\_info.sort\_values("Sodium\_(mg)", inplace=True)

# Sorts by descending order, rather than ascending.

food\_info.sort\_values("Sodium\_(mg)", inplace=True, ascending=False)