Библиотека Pandas – позволяет работать с данными в виде таблиц.

Import pandas as pd

Frame = pd.DataFrame({‘numbers’:range(10), ‘chars’: [‘a’]\*10}) Получим таблицу

Frame = pd.read.csv(‘dataset.tsv’, header=0, sep = ‘\t’) Считываем файл в таблицу

Frame.columns для выведения имен столбцов

Frame.shape для вывода размера таблицы

Frame = Frame.append(new\_line, ignore\_index=True) Для добавления новой строки к таблице

Frame[‘IsStudent’]=[False]\*3+[true]\*2 Для добавления столбца

Frame.drop([5,6],axis=0, inplace=True) Для удаления строк

Frame.drop(‘IsStudent’,axis=0, inplace=True) Для удаления столбца

Frame.to\_csv(‘update\_dataset.tsv’, header=True, sep = ‘,’, index=None) Для записи в файл

Frame.Birth = frame.Birth.apply(pd.to\_datetime) для изменения типа данных в столбце

Frame.dtypes для определения типа данных

Frame.info() немного информации о таблице

Frame.fillna(‘разнорабочий’, inplace=True) для до-заполнения пропущенных ячеек

Frame.Position

frame[[‘position’]] для вывода одного столбца

Frame.head(3)

frame[:3] для вывода первых трех строк

Frame.loc[[1,3,5],[‘Name’,’City’]]

Frame.iloc[[1,3,5],[1,3]]

Frame.ix для вывода опред. строк и столбцов

Frame[frame.Birth >= pd.datetime(1985,1,1)] фильтрация по дате

Библиотека numpy

Import numpy as np

X = [2,3,4,5]

Y = np.array(x) создание массива

Y[y>3] фильтрация

Y \* 5 умножение каждого элемента массива на 5

Np\_array[1,2] извлечение 2 элемента из 1 массива двумерной матрицы

Np.random.randn() рандомное число нормального распределения

Np.arange(0,8,0.1) массив чисел с шагом

Np.range(0,8,1) с целым шагом

b = numpy.array([1, 2, 3, 4, 5], dtype=float)

d = np.arange(start=10, stop=20, step=2)

создаст вектор

numpy.linalg.norm(a, ord=2) норма

numpy.dot(a, b) скалярное произведение

np.linalg.det(a) определитель матрицы

vals, vecs = np.linalg.eig(a) собственный вектор, значение

norm(a - b, ord=1) расстояние

a.transpose() транспонирование

Библиотека scipy

From scipy import optimize

Def f(x):

…..

X\_min = optimize.brute(f,((-5,5),(-5,5)) )

x\_min = optimize.minimize(f,[5,5])

X\_min = optimize.differential\_evolution(f,((-5,5),(-5,5)) )

для получения минимального значения функции и соответственных аргументов

From scipy import linalg

X = linalg.solve(a,b) решение системы уравнений

Np.dot(a,x) перемножение двух матриц

U, v, d = linalg.svd(x) сингулярное разложение матрицы

scipy.spatial.distance.cdist(b, a, metric='euclidean') расстояние между векторами

scipy.optimize.anneal метод оптимизации (имитации отжига)

scipy.optimize.diffentialevolution метод оптимизации (дифференциальной эволюции)

scipy.optimize.minimize метод оптимизации (Нелдера-Мида)

P(A+B)=P(A)+P(B)−P(AB) P(B\A)=P(B)−P(AB) P(A|B)=P(AB)P(B)

Формула полной вероятности: P(A)=P(A|B)P(B)+P(A| ̄B)P( ̄B).

Формула Байеса: P(A|B)=P(A)P(B|A)P(B)

P(X=k) =Cknpk(1−p)n−k биномиальное распределение

Распределение Пуассона: P(X=k) =λke−λk!, λ >0, k= 0,1,2,...

Равномерное распределение: f(x) =1b−a, x∈[a,b],0,x /∈[a,b].

f(x) =1√2πσ2exp(−(x−μ)22σ2) нормальное распределение

Частоты соответствующих событий на выборке (по закону больших чисел): ̄pk=1nn∑i=1[Xi=ak]. Если непрерывная случайная величина задается с помощью функции распределения, то ее можно оценить с помощью эмпирической функции распределения: X∼F(x), Fn(x) =1nn∑i=1[Xi6x].

Ядерная оценка плотности имеет вид: fn(x) =1nhn∑i=1K(Xi−xh).

Матожидание: EX=∑iaipi, X дискретная, +∞∫−∞xf(x)dx, X непрерывна.

Квантилем порядка α∈(0,1) называется величина Xα такая, что: P(X6Xα)>α, P(X>Xα)>1−α. Медиана это квантиль порядка 0,5:P(X6medX)>0,5, P(X>med X)>0,5.

Еще одной характеристикой среднего является мода - самое вероятное значение случайной величины (в нестрогом смысле):modeX=aargmaxpii, X дискретна,argmaxf(x)x, Xнепрерывна.

Дисперсия: DX=E((X−EX)2).Часто используется величина √DX, называемая среднеквадратическое отклонение. Еще одна характеристика разброса интерквантильный размах: IQR =X0,75−X0,25.

Выборочная дисперсия оценивает дисперсию и имеет следующий вид: S2=1n−1n∑i=1(Xi− ̄X)2.

Выборочный интерквартильный размах определяется следующим образом: IQRn=X([0,75n])−X([0,25n]).

Центральной предельной теоремы: X∼F(x), Xn=(X1,X2,...,Xn)⇒ ̄Xn≈∼N(EX,DXn).