Xxxxx = [] #List

Xxxxx={} #dictionary

import pandas as pd # use pandas， to create series, run pd.series()

import os # for reading file

import numpy as np

data\_file = “ dataset.scv”

NaN # means not a number

student\_data = os.path.join( 'new\_student\_data.csv') # read data in the same folder

student\_df = pd.read\_csv(student\_data) # read csv data as data frame

pd.DataFrame(dicts) # make dicts to a data frame

xxx\_df.head #show first 5 rows

xxx\_df[“column”] # will call column item only.

Xxx \_df.isnull() # show which cell is null by True or false

Xxx \_df.isna() # show nan values

xxx\_df.isnull().sum() # show summary of the column that has null

Xxx\_df.isnull().mean() # percentage of missing values in each column

xxx.dropna() # To drop the rows of data that have missing values

xxx.fillna() # to replace the NaNs with whatever inside the ()

xxx. duplicated () # identifying duplicated rows in a DataFrame.by true or false in the row

xxx\_df.describe() # outputs the same list of summary statistics

**student\_df = student\_df.drop\_duplicates()**

xxx..duplicated().sum() # return the number of duplicated value

xxx\_df.drop\_duplicates() #drop duplicate cells

xxx \_df.dtype # check data type

xxx\_df.max() # return max value

xxx\_df.mean() # return mean value of the dataset

Xxx\_pd[series].unique # look into series and returns all distinct values

Xxx\_df[‘series’].value\_counts()

#returns a list of all unique within the series but count how many times a unique value appears.

**student\_df['grade'] = student\_df['grade'].astype(int)**

**#** Change the type of the “grade“ column to int by using astype

**prices.loc[:, "price\_usd"] = prices.loc[:, "price\_usd"].str.replace("$", "")**

.str.replace # function to replace each instance of the "$" currency symbol string with an empty string

# using loc or iloc to specify that column

# to collect specific rows or columns of data from a dataframe

Df.loc[[row”s”][coloum] #

.iloc[start:end] # return out from start to end in a list

**prices.loc[:, "price\_usd"] = prices.loc[:, "price\_usd"].astype("float")**

# convert our price data to floating point numbers

del df[‘series’] # delete the column

df.info() # give you some info

Df[df.isna().any(axis=1)] #drop nan value that has row not column

how = 'any‘#f any NA values are present, drop that row or column

how =‘all’ # If all values are NA, drop that row or column.

Df = original \_df.set\_index(“last\_name”) # set new index to last name

Df.shape() #stores the number of rows and columns as a tuple (number of rows, number of columns)

Df[‘series’].value\_counts() #display an overview of the series column

Df.groupby([“series”])

.reset\_index() # to reset index for new grouped string

Df[df[‘first name’] == “Billy’ ]# show row first name that is billy only.

Import matplotlib.pyplot as plt

Np.arange(0, 5, 0.1) #output series from 0 to 5 increasing 0.1 each time,exclude 5

Pyplot.plot() to tell matplotlib what data to use and which plot to make

Plt.plot(x\_axis,e\_x)

Plt.show（） #show the graph created from above

Plt.xlabel(“xxx”) # label x axis

plg.savefig("xxx") # save graph as image

plt.bar(x\_axis,users,color=”r”,alpha=0.5,align=”center”) # create bar graph

tick\_locations = [value for value in x\_xaxi] # set the definition of tick location

plt.xticks(tick\_locations,[“A”,“”B”,] # place A and B to x x axis as column

Labels=[“xxx”,”xxxx”]

Sizes = [1,2,3]

Colors= [“red”,”yellow”]

Explode = (0.1,0,0,0) # tell pie to separate the first category from the other 3 category.

Plt.pie(sizes,explode=explode,labels=labels,colors=colors,autopct=”%1.1f%%”,shadow=True,startangle=140)

plt.scatter(x\_axix,data, market=”o”,facecolor=”red”,edgecolors=”black”, s=x\_axis, s)