**Web Programming and Python: Assignment-11**

**Name: Leon Lobo**

**Admission Number:** U24AI043

1. Write a Pandas program to create

a) Date time object for Jan 15 2012.

b) Specific date and time of 9:20 pm.

c) Local date and time.

d) A date without time.

e) Current date.

t) Time from a date time.

g) Current local time.

**Code:**

def main():

import pandas as pd

from datetime import datetime

dt\_a=pd.Timestamp('2012-01-15')

print("Date time object for Jan 15 2012:",dt\_a)

dt\_b=pd.Timestamp('2025-04-24 21:20')

print("Specific date and time of 9:20 pm:",dt\_b)

dt\_c=pd.Timestamp.now()

print("Local date and time:",dt\_c)

dt\_d=pd.to\_datetime('2025-04-24').date()

print("A date without time:",dt\_d)

dt\_e=pd.Timestamp.today().date()

print("Current date:",dt\_e)

dt\_f=pd.Timestamp('2025-04-24 21:20').time()

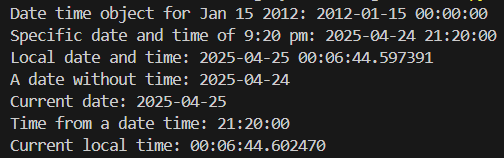
print("Time from a date time:",dt\_f)

dt\_g=datetime.now().time()

print("Current local time:",dt\_g)

if \_\_name\_\_=='\_\_main\_\_':

main()

**Output:**

2. Write a Pandas program to convert all the string values to upper, lower cases in a given

pandas series. Also find the length of the string values.

s = pd.Series ([‘X’, ‘Y’, ‘T’, ‘Aaba’, ‘Baca’, ‘CABA’, None, ‘bird’, ‘horse’, ‘dog’])

**Code:**

def main():

import pandas as pd

s=pd.Series(['X','Y','T','Aaba','Baca','CABA',None,'bird','horse','dog'])

upper\_case=s.str.upper()

print("Upper case:\n",upper\_case)

lower\_case=s.str.lower()

print("Lower case:\n",lower\_case)

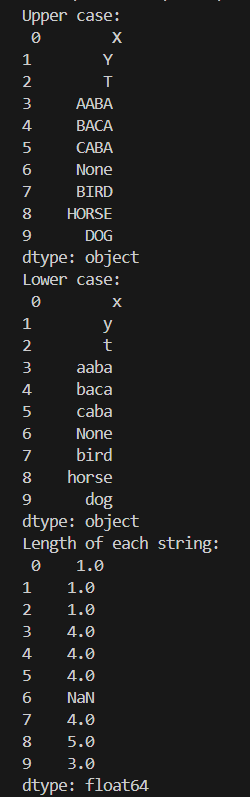
string\_length=s.str.len()

print("Length of each string:\n",string\_length)

if \_\_name\_\_=='\_\_main\_\_':

main()

**Output:**



3. After accidentally leaving an ice chest of fish and shrimp in your car for a week while you

were on vacation, you’re now in the market for a new vehicle. Your insurance didn’t cover

the loss, so you want to make sure you get a good deal on your new car.

Given a Series of car asking\_prices and another Series of car fair\_prices, determine which

cars for sale are a good deal. In other words, identify cars whose asking price is less than

their fair price.

The result should be a list of integer indices corresponding to the good deals

in asking\_prices.

**Code:**

def main():

import pandas as pd

cars=int(input("Enter no. of cars:"))

asking\_price=input("Enter asking prices: ")

values=list(map(int,asking\_price.split()))

asking\_prices=pd.Series(values)

fair\_price=input("Enter fair prices: ")

values=list(map(int,fair\_price.split()))

fair\_prices=pd.Series(values)

good\_deals=asking\_prices<=fair\_prices

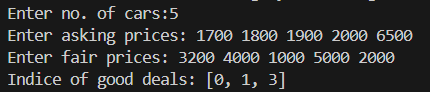
good\_deals\_indices=good\_deals[good\_deals].index.tolist()

print("Indice of good deals:", good\_deals\_indices)

if \_\_name\_\_=='\_\_main\_\_':

main()

**Output:**



4. Whenever your friends John and Judy visit you together, y’all have a party. Given a

DataFrame with 10 rows representing the next 10 days of your schedule and whether John

and Judy are scheduled to make an appearance, insert a new column

called days\_til\_party that indicates how many days until the next party.

days\_til\_party should be 0 on days when a party occurs, 1 on days when a party doesn’t

occur but will occur the next day, etc.

**Code:**

import pandas as pd

import numpy as np

john\_data=[]

judy\_data=[]

print("Enter whether John and Judy are visiting as 'True' or 'False':")

for i in range(10):

john\_val=input(f"Day {i+1} - John: ").strip().capitalize()

judy\_val=input(f"Day {i+1} - Judy: ").strip().capitalize()

john\_data.append(john\_val=="True")

judy\_data.append(judy\_val=="True")

df=pd.DataFrame({'John':john\_data,'Judy':judy\_data})

party\_days=(df['John']&df['Judy']).values

days\_til\_party=np.full(len(df),fill\_value=-1)

next\_party=None

for i in reversed(range(len(df))):

if party\_days[i]:

next\_party=i

days\_til\_party[i]=0

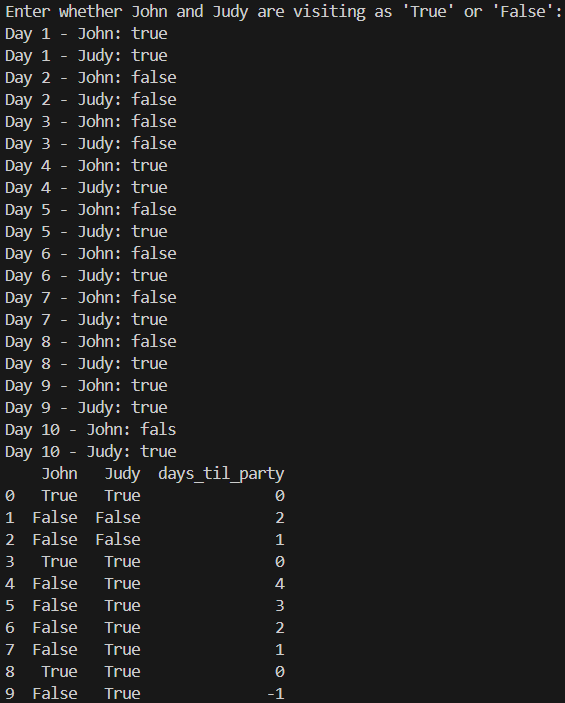
elif next\_party is not None:

days\_til\_party[i]=next\_party-i

df['days\_til\_party']=days\_til\_party.astype(int)

print(df)

**Output:**



5. Given a dataset of concerts, count the number of concerts per (artist, venue), per year

month. Make the resulting table be a wide table - one row per year month with a column

for each unique (artist, venue) pair. Use the cross product of the artists and venues Series

to determine which (artist, venue) pairs to include in the result.

**Code:**

import pandas as pd

df = pd.DataFrame({'artist':['A','B','C','C','C','A','B'],'venue':['Z','X','Y', 'Y','Z','X','Z'],'date':pd.to\_datetime(['2024-01-10','2024-01-15','2024-01-20','2024-02-05','2024-02-20','2024-02-25','2024-03-01'])})

df['year\_month']=df['date'].dt.to\_period('M')

artists=df[['artist']]

venues=df[['venue']]

cross=artists.merge(venues,how='cross')

grouped=(df.groupby(['year\_month','artist','venue']).size().rename('count').reset\_index())

all\_months=df[['year\_month']].drop\_duplicates()

full\_index=all\_months.merge(cross,how='cross')

full\_data=(full\_index.merge(grouped,on=['year\_month','artist','venue'],how='left').fillna(0))

pivot=full\_data.pivot\_table(index='year\_month',columns=['artist','venue'],values='count',fill\_value=0)

pivot.columns=['\_'.join(col).strip() for col in pivot.columns.values]

print(pivot)

**Output:**

