U24AI041 PYTHON lab 11

Q1:

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:

import pandas as pd

date='2012-01-15'

dobj=pd.to\_datetime(date)

print("date object created:")

print( dobj)

dt='2012-01-15 9:20:00 pm'

print("Particular date and time:")

dtobj=pd.to\_datetime(dt)

print(dtobj)

lt=pd.Timestamp.now()

print("Local date & time:")

print(lt)

print("Date without time:")

dateWithoutTime=lt.date()

print(dateWithoutTime)

print("Current date:")

localTime=lt.time()

print("Local time:")

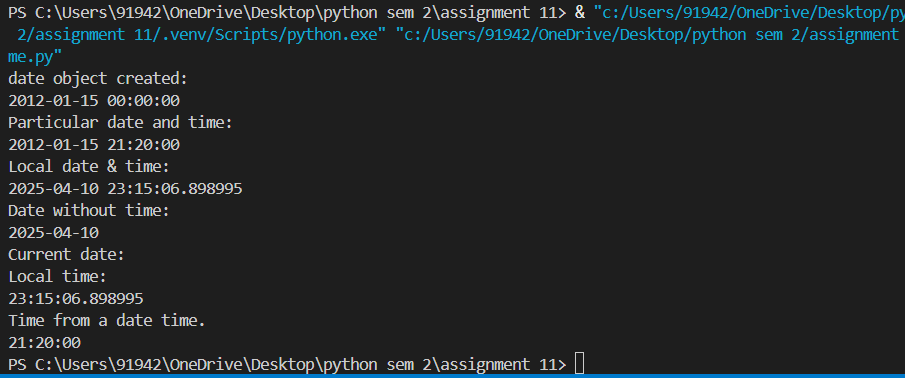
print(localTime)

t=pd.to\_datetime(dt)

print("Time from a date time.")

print(t.time())

OUTPUT:



Q2:

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:

import pandas as pd

n = int(input("Enter size of array\n"))

a = []

for i in range(n):

    k = input("Enter string: ")

    a.append(k)

ans = pd.Series(a)

upper\_strings = ans.str.upper()

lower\_strings = ans.str.lower()

lenghth = ans.str.len()

print("Uppercase Strings:")

print(upper\_strings.to\_list())

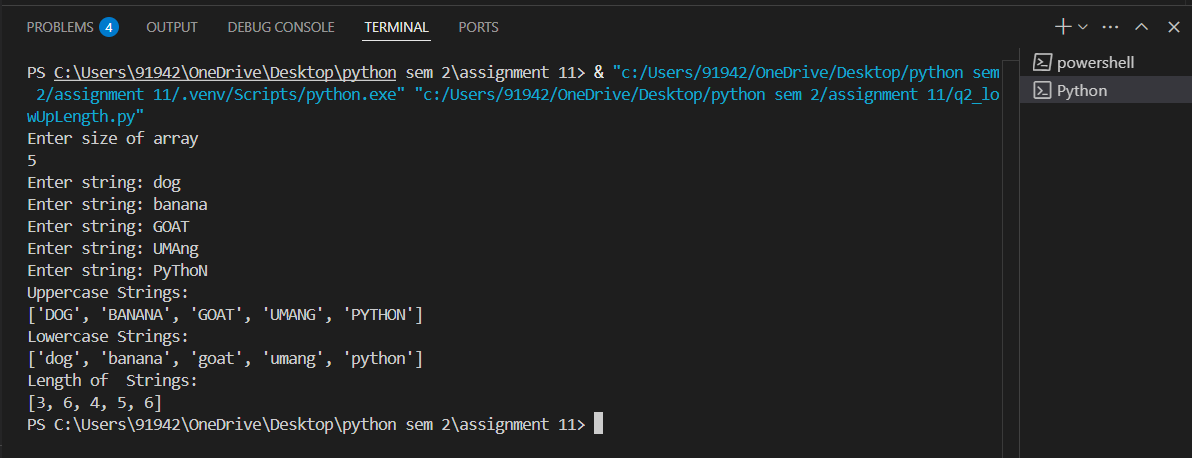
print("Lowercase Strings:")

print(lower\_strings.to\_list())

print("Length of  Strings:")

print(lenghth.to\_list())

OUTPUT:



Q3:

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:

import pandas as pd

n= int(input("Enter total no. of asking prices\n"))

asking\_prices = []

fair\_prices = []

for i in range(n):

    k = input("Enter asking price: ")

    asking\_prices.append(k)

    u = input("Enter fair price: ")

    fair\_prices.append(u)

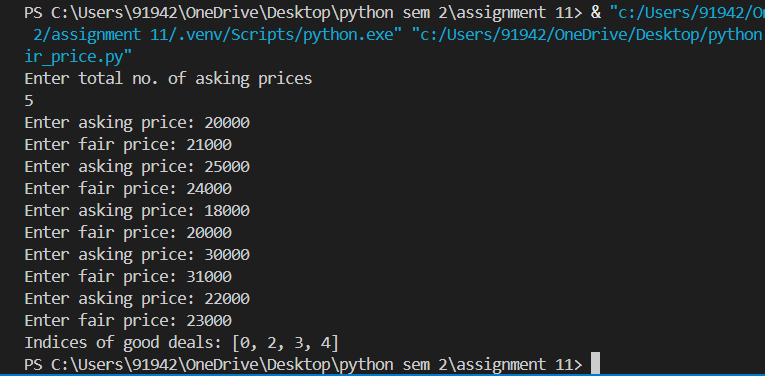
asking\_prices = pd.Series(asking\_prices)

fair\_prices = pd.Series(fair\_prices)

good\_deals = asking\_prices[asking\_prices < fair\_prices].index.tolist()

print("Indices of good deals:", good\_deals)

OUTPUT:



Q4:

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

# 1 indicates visits, 0 indicates he doesn't

data = {

    "John": [1, 0, 1, 1, 0, 0, 1, 0, 1, 1],

    "Judy": [1, 1, 0, 1, 0, 1, 1, 0, 1, 1]

}

schedule = pd.DataFrame(data)

schedule["party"] = (schedule["John"] & schedule["Judy"]).astype(int)

days\_til\_party = []

for i in range(len(schedule)):

    if schedule["party"].iloc[i] == 1:

        days\_til\_party.append(0)

    else:

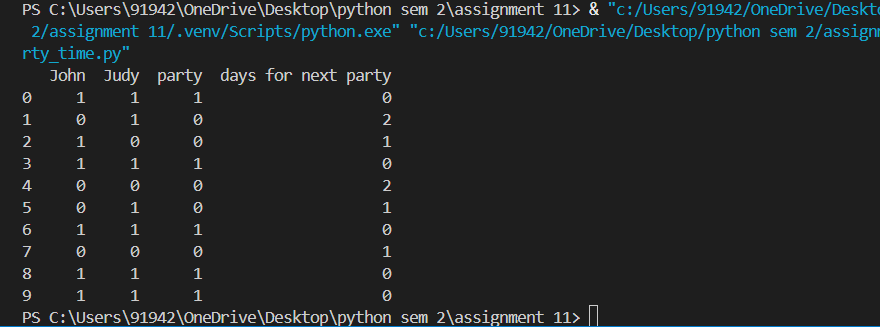
        next\_party = schedule["party"].iloc[i:].idxmax()

        days\_til\_party.append(next\_party - i if schedule["party"].iloc[next\_party] == 1 else len(schedule) - i)

schedule["days for next party"] = days\_til\_party

print(schedule)

OUTPUT:



Q5:

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

import itertools

data = {

    "artist": ["A", "B", "A", "C", "B", "A", "C", "B","A","C","B"],

    "venue": ["X", "Y", "X", "Z", "Y", "X", "Z", "Y","Z","Y","Z"],

   "date": [

        "2025-01-15", "2025-10-15", "2025-02-20", "2025-02-20",

        "2025-03-10", "2025-10-10", "2025-04-15", "2025-04-15", "2025-11-15", "2025-01-20", "2025-03-20"

    ]

}

concerts = pd.DataFrame(data)

concerts["year\_month"] = pd.to\_datetime(concerts["date"]).dt.to\_period("M")

concert\_counts = concerts.groupby(["year\_month", "artist", "venue"]).size().reset\_index(name="count")

artists = concerts["artist"].unique()

venues = concerts["venue"].unique()

artist\_venue\_pairs = pd.DataFrame(list(itertools.product(artists, venues)), columns=["artist", "venue"])

# Create table

result = (

    concert\_counts

    .merge(artist\_venue\_pairs, on=["artist", "venue"], how="right")

    .pivot(index="year\_month", columns=["artist", "venue"], values="count")

    .fillna(0)  # Fill missing values with 0

    .astype(int)

)

print(result)

OUTPUT:

