Istanbul Medipol University
School of Engineering and
Natural Sciences
Computer Engineering

Spring 2022 – Assignment #3
Introduction to Programming
Assignment Date: 14.05.2022
Due Date: 30.05.2022

Programming Assignment #3

(Use a jupyter notebook to solve the following Pandas exercises)

Question I (20 pts): Perform the following tasks with pandas Series:

a. Create a Series from the list [7, 11, 13, 17].

b. Create a Series with five elements that are all 100.0.

c. Create a Series with 20 elements that are all random numbers in the range 0 to 100. Use method describe to produce the Series' basic descriptive statistics.

d. Create a Series called temperatures of the floating-point values 98.6, 98.9, 100.2 and 97.9. Using the index keyword argument, specify the custom indices 'Julie', 'Charlie', 'Sam' and 'Andrea'.

e. Form a dictionary from the names and values in Part (d), then use it to initialize a Series.

Question II (40 pts): Perform the following tasks with pandas DataFrames:

a. Create a DataFrame named temperatures from a dictionary of three temperature readings each for 'Maxine', 'James' and 'Amanda'.

b. Recreate the DataFrame temperatures in Part (a) with custom indices using the index keyword argument and a list containing 'Morning', 'Afternoon' and 'Evening'.

c. Select from temperatures the column of temperature readings for 'Maxine'.

d. Select from temperatures the row of 'Morning' temperature readings.

e. Select from temperatures the rows for 'Morning' and 'Evening' temperature readings.

f. Select from temperatures the columns of temperature readings for 'Amanda' and 'Maxine'.

g. Select from temperatures the elements for 'Amanda' and 'Maxine' in the 'Morning' and 'Afternoon'.

h. Use the describe method to produce temperatures' descriptive statistics.

i. Transpose temperatures.

j. Sort temperatures so that its column names are in alphabetical order.

Question III (40 pts): These questions are based on Human Resources (HR) database given in site https://www.w3resource.com/python-exercises/pandas/index.php. This site includes Pandas exercises, practice facilities and solutions of some exercises. You can look at these exercises before solving the following questions. CSV files in HR database can be found in assignment's attachments (**HR-**

Database.rar). First, generate a data frame for each of tables in HR Database as follows:

```
import pandas as pd
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
employees = pd.read_csv(r"EMPLOYEES.csv")
departments = pd.read_csv(r"DEPARTMENTS.csv")
job_history = pd.read_csv(r"JOB_HISTORY.csv")
jobs = pd.read_csv(r"JOBS.csv")
countries = pd.read_csv(r"COUNTRIES.csv")
regions = pd.read_csv(r"REGIONS.csv")
locations = pd.read_csv(r"LOCATIONS.csv")
```

a. Write a Pandas program to display all the records of DEPARTMENTS file.
b. Display the number of records in each of data frames (records or file).
c. Display employees who has salary > 10000.
d. In employees data frame, the column commission_pct has some none values (NaN). Fill these none values with 0.
e. Display the first name, last name, salary, and department number for those employees # who work in departments with ids 30, 50 or 80.
f. Merge/Join data frames employees and departments using their common column department_id. Store the result in a new data frame called emp_dept.
g. Find the minimum, maximum and mean salaries of employees in each department (use empt_dept).

	salary		
	min	max	mean
department_name			
Accounting	8300	12000	10150.000000
Administration	4400	4400	4400.000000
Executive	17000	24000	19333.333333
Finance	6900	12000	8600.000000
Human Resources	6500	6500	6500.000000

h. Find mean salaries of employees grouped by country_id, city, # in ranges (0, 5000] (5000, 10000] (10000, 15000] (15000, 25000]. (First, merge/join locations and empt_dept.)

		salary			
	salary	(0, 5000]	(5000, 10000]	(10000, 15000]	(15000, 25000]
country_id	city				
CA	Toronto	0	6000.000000	13000.000000	0.000000
DE	Munich	0	10000.000000	0.000000	0.000000
UK	London	0	6500.000000	0.000000	0.000000
	Oxford	0	8096.153846	11750.000000	0.000000
US	Seattle	3050	7983.333333	11666.666667	19333.333333
	South San Francisco	3000	7280.000000	0.000000	0.000000
	Southlake	4600	7500.000000	0.000000	0.000000

Question IV (40 pts): A data repository is maintained by Johns Hopkins University CSSE research center (https://github.com/CSSEGISandData/COVID-19/) about corona virus incidents. The site https://www.w3resource.com/python-exercises/project/covid-19/index.php includes some exercises on COVID-19 data set. You can look at these exercises before solving the following questions. First, get the latest covid data from github as follows:

Import data

```
import pandas as pd

covid_data= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_daily_reports/05-10-2022.csv')

covid_series= pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
```

a. (5 pts) Display first 5 rows from COVID-19 daily summary (covid_data) and time series (covid_series) datasets.

b. (5 pts) Write a Python program to get the latest number of confirmed, deaths, recovered and active cases of Novel Coronavirus (COVID-19) Country wise. The result should be sorted by Active cases.

	Country_Region	Confirmed	Deaths	Recovered	Active
171	US	461437	16478	25410	419549
84	Italy	143626	18279	28470	96877
156	Spain	153222	15447	52165	85610
61	France	118781	12228	23413	83140
65	Germany	118181	2607	52407	63167
175	United Kingdom	65872	7993	359	57520
170	Turkey	42282	908	2142	39232
80	Iran	66220	4110	32309	29801
120	Netherlands	21903	2403	278	19222
16	Belgium	24983	2523	5164	17296

c. (10 pts) Write a Python program to get the countries data (Country/Region, Last Update, Confirmed, Deaths, # Recovered, Active, and death_confirmed ratio) of Novel Coronavirus (COVID-19). The death_confimed_ratio is # Deaths / Confirmed * 100. The result should only contain countries where confimed cases > 1000 and be sorted by death_confirmed ratio.

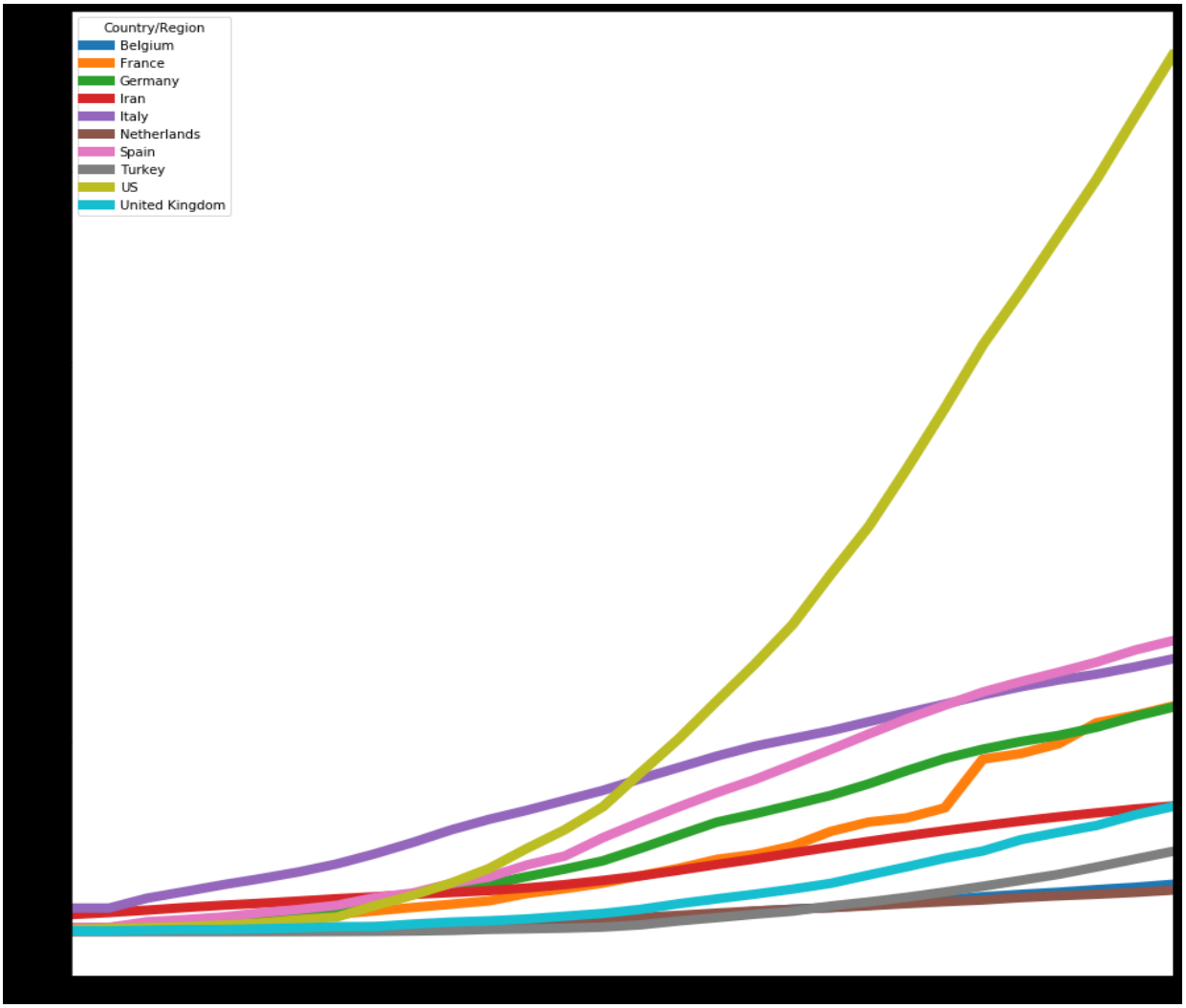
Country_Region	Last_Update	Confirmed	Deaths	Recovered	Active	Death_Confirmed_Ratio
Algeria	2020-04-09 23:02:19	1666	235	347	1084	14.105642
Italy	2020-04-09 23:02:19	143626	18279	28470	96877	12.726804
United Kingdom	2020-04-09 23:02:19	65077	7978	135	56964	12.259324
Netherlands	2020-04-09 23:02:19	21762	2396	250	19116	11.010017
France	2020-04-09 23:02:19	117749	12210	23206	82333	10.369515
Belgium	2020-04-09 23:02:19	24983	2523	5164	17296	10.098867
Spain	2020-04-09 23:02:19	153222	15447	52165	85610	10.081450
Sweden	2020-04-09 23:02:19	9141	793	205	8143	8.675200
Indonesia	2020-04-09 23:02:19	3293	280	252	2761	8.502885
Morocco	2020-04-09 23:02:19	1374	97	109	1168	7.059680

d. (20 pts) From question 4.b, get the list of top 10 countries according to confirmed cases. Then select data belong to the top 10 countries from convid_series data frame. By grouping with 'Country/Region', get the sum of cases in countries. Only select the columns after the date ‘3/11/21’ that is the first date in which a death case recorded in Turkey. The following figure shows a part of data you will obtain:

Country/Region	3/11/20	3/12/20	3/13/20	3/14/20	3/15/20	3/16/20	3/17/20	3/18/20	3/19/20	3/20/20	3/21/20	3/22/20	3/23/20	3/24/20	3/25/20	3/26/20	3/27/20
Belgium	314	314	559	689	886	1058	1243	1486	1795	2257	2815	3401	3743	4269	4937	6235	7284
France	2293	2293	3681	4496	4532	6683	7715	9124	10970	12758	14463	16243	20123	22622	25600	29551	33402
Germany	1908	2078	3675	4585	5795	7272	9257	12327	15320	19848	22213	24873	29056	32986	37323	43938	50871
Iran	9000	10075	11364	12729	13938	14991	16169	17361	18407	19644	20610	21638	23049	24811	27017	29406	32332
Italy	12462	12462	17660	21157	24747	27980	31506	35713	41035	47021	53578	59138	63927	69176	74386	80589	86498
Netherlands	503	503	806	962	1138	1416	1711	2058	2467	3003	3640	4217	4764	5580	6438	7468	8647
Spain	2277	2277	5232	6391	7798	9942	11748	13910	17963	20410	25374	28768	35136	39885	49515	57786	65719
Turkey	1	1	5	5	6	18	47	98	192	359	670	1236	1529	1872	2433	3629	5698
US	1281	1663	2179	2727	3499	4632	6421	7783	13747	19273	25600	33276	43847	53740	65778	83836	101657
United Kingdom	459	459	802	1144	1145	1551	1960	2642	2716	4014	5067	5745	6726	8164	9640	11812	14745

After getting these data, plot the graph of data by using the following code:

```
%matplotlib inline
import matplotlib.pyplot as plt
temp2.T.plot(figsize=(15,15),lw=8)
```



****

Submit your program code as .ipynb file (like COE-64160099-KAYA-A2.ipynb), python .py file (like COE-64160099-KAYA-A2.py) and html (like 64160099-KAYA-A2.html) file. Put all files inside of a ZIP file (64160099-KAYA-A2.zip)

To obtain .py file, select

- *File / (Export Notebook As.. / Export Notebook to Executacle Script* (if you are using Jupyter lab)
- *File / Download as / Python (.py)* (if you are using jupyter)

after completing your solution using Jupyter notebooks.

To obtain .html file, select

- *File / (Export Notebook As.. / Export Notebook to HTML* (if you are using Jupyter lab)
- *File / Download as / HTML (.html)* (if you are using jupyter)

after completing your solution using Jupyter notebooks.

The first cell of the notebook should contain information about you as follows:

[1]:
```
##############################################################################################
#  Name:         Ali Cokcalışır
#  Student ID:   6321211
#  Department:   Computer Engineering
#
#  Assignment ID: A2
##############################################################################################
```

[2]:
```
import numpy as np
```