

KONGU ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE
Mini Project
E.SRIMATHI – 21ALR044

CLASS	II yr AI-ML	PROJECT ID	PID03
SUBJECT	20ALC33 - DATA VISUALIZATION	DATE OF COMPLETION	11.11.2022

Question:

Create a cheat sheet for the python pandas function by applying the following operation on Anime Recommendations Database which can be downloaded from the Kaggle repository(<https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database>). This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings.

Solution:

Datasets used:

1. anime.csv
2. rating.csv

A. Convert a CSV file directly into a data frame Coding:

```
import pandas as pd
df = pd.read_csv('/content/anime.csv') df
```

Output:

anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37 200630

anime_id	name	genre	type	episodes	rating	members
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26 793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25 114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17 673572

4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266
...
12289	9316	Toushindai My Lover: Minami tai MechaMinami	Hentai	OVA	1	4.15	211
12290	5543	Under World	Hentai	OVA	1	4.28	183
12291	5621	Violence Gekiga David no Hoshi	Hentai	OVA	4	4.88	219
12292	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Hentai	OVA	1	4.98	175
12293	26081	Yasuji no Pornorama: Yacchimae!!	Hentai	Movie	1	5.46	142

12294 rows × 7 columns

Coding:

```
df1 = pd.read_csv('/content/rating.csv')
df1
```

Output:

	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1
...
	user_id	anime_id	rating
550456	5487	7	8

550457	5487	30	7
550458	5487	31	7
550459	5487	32	7
550460	5487	43	10

550461 rows × 3 columns

B. Concatenate 2 dataframes Coding:

```
vertical_concat = pd.concat([df, df1], axis=0) #Concatenating along row
horizontal_concat = pd.concat([df, df1], axis=1) #concatenating along column
display(vertical_concat, horizontal_concat)
```

Output:

#VERTICAL

anime_id	name	genre	type	episodes	rating	members	user_id	
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630.0	NaN
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665.0	NaN
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262.0	NaN
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572.0	NaN
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266.0	NaN
...
550456	7	NaN	NaN	NaN	NaN	8.00	NaN	5487.0

anime_id	name	genre	type	episodes	rating	members	user_id	
550457	30	NaN	NaN	NaN	NaN	7.00	NaN	5487.0
550458	31	NaN	NaN	NaN	NaN	7.00	NaN	5487.0
550459	32	NaN	NaN	NaN	NaN	7.00	NaN	5487.0
550460	43	NaN	NaN	NaN	NaN	10.00	NaN	5487.0

562755 rows × 8 columns

#HORIZONTAL

anime_id	name	genre	type	episodes	rating	members	user_id	anime_id	rating
0	32281.0	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630.0	1	20 -1
1	5114.0	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665.0	1	24 -1
2	28977.0	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262.0	1	79 -1
3	9253.0	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572.0	1	226 -1
4	9969.0	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266.0	1	241 -1

...
550456	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5487	7	8
550457	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5487	30	7
550458	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5487	31	7
550459	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5487	32	7
anime_id	name	genre	type	episodes	rating	members	user_id	anime_id	rating	
550460	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5487	43	10

550461 rows × 10 columns

C. Merge dataframes:

Coding:

```
newdf1 = df.merge(df1,how='right')
newdf1.head()
```

Output:

	anime_id	name	genre	type	episodes	rating	members	user_id
0	20	NaN	NaN	NaN	NaN	-1.0	NaN	1
1	24	NaN	NaN	NaN	NaN	-1.0	NaN	1
2	79	NaN	NaN	NaN	NaN	-1.0	NaN	1
3	226	NaN	NaN	NaN	NaN	-1.0	NaN	1
4	241	NaN	NaN	NaN	NaN	-1.0	NaN	1

Coding:

```
newdf4 = df.merge(df1,how='left') newdf4.head()
```

Output:

anime_id	name	genre	type	episodes	rating	members	user_id
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630

1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665	NaN
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262	NaN
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572	NaN
4	9969	Gintama'	Action, Comedy, Historical,	TV	51	9.16	151266	NaN
anime_id	name	genre	type	episodes	rating	members	user_id	
			Parody, Samurai, S...					

Coding:

```
newdf3 = df.merge(df1) newdf3.head()
```

Output:

anime_id	name	genre	type	episodes	rating	members	user_id	
0	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	1418
1	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	1798
2	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	2072
3	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	2533
4	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	4985

D. Retrieve rows with matching index values(explicit index)

Coding:

```
df2=newdf3.loc[:,["anime_id","rating"]] df2.head()
```

Output:

	anime_id	rating
0	1470	8.0
1	1470	8.0
2	1470	8.0
3	1470	8.0
4	1470	8.0

E. Retrieve rows by numbered index values(implicit index)

Coding:

```
df2=newdf3.iloc[:, [1,2,4]] df2.head()
```

Output:

	name	genre	episodes
0	City Hunter	Action, Comedy, Mystery, Shounen	51
1	City Hunter	Action, Comedy, Mystery, Shounen	51
2	City Hunter	Action, Comedy, Mystery, Shounen	51
3	City Hunter	Action, Comedy, Mystery, Shounen	51
4	City Hunter	Action, Comedy, Mystery, Shounen	51

F. Retrieve rows where a column's value is in a given list

Coding:

```
newdf3.iloc[[3,8,9], [1,3,4]]
```

Output:

	name	type	episode
3	City Hunter	TV	51
8	Detective Conan Movie 02: The Fourteenth Target	Movie	1
9	Detective Conan Movie 02: The Fourteenth Target	Movie	1

G. Slice a data frame to get all rows before/between/after specified indices Coding:

```
newdf3.iloc[:, :3].head()
```

Output:

	anime_id	name	genre
0	1470	City Hunter	Action, Comedy, Mystery, Shounen
1	1470	City Hunter	Action, Comedy, Mystery, Shounen
2	1470	City Hunter	Action, Comedy, Mystery, Shounen
3	1470	City Hunter	Action, Comedy, Mystery, Shounen
4	1470	City Hunter	Action, Comedy, Mystery, Shounen

Coding:

```
newdf3.iloc[:, 5:7].head()
```

Output:

	rating	members
0	8.0	19963
1	8.0	19963
2	8.0	19963
3	8.0	19963
4	8.0	19963

Coding:

```
newdf3.iloc[:, 7:].head()
```

Output:

	user_id
0	1418
1	1798
2	2072
3	2533
4	4985

H. Filter data frame for rows that meet a condition Coding:

```
newdf3[newdf3["name"] == 'City Hunter'].head()
```

Output:

anime_id	name	genre	type	episodes	rating	members	user_id
----------	------	-------	------	----------	--------	---------	---------

0	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	1418
1	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	1798
2	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	2072
3	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	2533
4	1470	City Hunter	Action, Comedy, Mystery, Shounen	TV	51	8.0	19963	4985

I.Sort data frame by values in a column.

Coding:

```
newdf3.sort_values(by="rating").head()
```

Output:

anime_id	name	genre	type	episodes	rating	members	user_id	
562	3287	Tenkuu Danzai Skelter+Heaven	Mecha, Sci-Fi	OVA	1	2.0	7680	3479
561	3287	Tenkuu Danzai Skelter+Heaven	Mecha, Sci-Fi	OVA	1	2.0	7680	17
560	3112	Papillon Rose	Comedy, Ecchi, Magic, Parody	TV	6	5.0	1817	1145
520	31318	Comet Lucifer	Action, Adventure, Fantasy, Mecha	TV	12	6.0	66659	861
521	31318	Comet Lucifer	Action, Adventure, Fantasy, Mecha	TV	12	6.0	66659	1013

J. Count number of records for each distinct value in a column.

Coding:

```
n = newdf3.nunique(axis=0)
n
```

Output:

```
anime_id 48  
name 48  
genre 46  
type 6  
episodes 15  
rating 5  
members 48  
user_id 483  
dtype: int64
```

Coding:

```
n = newdf3.nunique(axis=1)
n
```

Output:

```
0 8
1 8
2 8
3 8
4 8 ..
577 8
578 8
579 8
580 8
581 8
Length: 582, dtype: int64
```

K. Groupby and aggregate columns in different ways

Coding:

```
grouped_multiple = df.groupby(['anime_id','type']).agg({'rating': ['mean', 'min', 'max']})
grouped_multiple.columns = ['rating_mean', 'rating_min', 'rating_max']
grouped_multiple = grouped_multiple.reset_index()
print(grouped_multiple)
```

Output:

	anime_id	type	rating_mean	rating_min	rating_max
1	TV	8.82	8.82	8.82	
1	5	Movie	8.40	8.40	8.40
2	6	TV	8.32	8.32	8.32
3	7	TV	7.36	7.36	7.36
4	8	TV	7.06	7.06	7.06
	
	
12264	34514	ONA	7.21	7.21	7.21
12265	34519	Special	5.67	5.67	5.67
12266	34522	TV	NaN	NaN	NaN
12267	34525	TV	NaN	NaN	NaN
	12268	34527	Movie	NaN	NaN
	NaN				

```
[12269 rows x 5 columns]
```

L. Create a pivot table Coding:

```
table = pd.pivot_table(data=newdf3, index=['episodes'], aggfunc='sum')
table
```

Output:

anime_id	members	rating	user_id	episodes
1	4163765	27124232	2091.0	752189
12	1585734	2953656	388.0	149872
13	200137	336209	148.0	75583
14	198410	180720	70.0	24418
2	757303	1359912	382.0	139583
22	677775	2601340	245.0	94574
24	5091	523	7.0	2774
26	8960	32755	35.0	18363
3	291161	1057705	242.0	111100
39	7262	1862	14.0	1022
4	374897	224566	139.0	48286
5	10164	51468	39.0	21638
51	8820	119778	48.0	18102
6	1246597	2372057	365.0	122115
7	9284	49132	32.0	19776

M. Set NaN cells to some value

Coding:

```
newdf3.isnull()
```

Output:

anime_id	name	genre	type	episodes	rating	members	user_id	
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...
577	False	False	False	False	False	False	False	False

578	False	False	False	False	False	False	False	False
579	False	False	False	False	False	False	False	False
anime_id	name	genre	type	episodes	rating	members	user_id	
580	False	False	False	False	False	False	False	False
581	False	False	False	False	False	False	False	False

582 rows × 8 columns

Coding:

```
newdf1.head()
```

Output:

	anime_id	name	genre	type	episodes	rating	members	user_id
0	20	NaN	NaN	NaN	NaN	-1.0	NaN	1
1	24	NaN	NaN	NaN	NaN	-1.0	NaN	1
2	79	NaN	NaN	NaN	NaN	-1.0	NaN	1
3	226	NaN	NaN	NaN	NaN	-1.0	NaN	1
4	241	NaN	NaN	NaN	NaN	-1.0	NaN	1

Coding:

```
newdf1.fillna(0).head()
```

Output:

	anime_id	name	genre	type	episodes	rating	members	user_id
0	20	0	0	0	0	-1.0	0.0	1
1	24	0	0	0	0	-1.0	0.0	1
2	79	0	0	0	0	-1.0	0.0	1
3	226	0	0	0	0	-1.0	0.0	1
4	241	0	0	0	0	-1.0	0.0	1

Coding:

```
newdf4.head()
```

Output:

	anime_id	name	genre	type	episodes	rating	members	user_id
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	NaN

1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665	NaN
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262	NaN
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572	NaN
	anime_id	name	genre	type	episodes	rating	members	user_id
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266	NaN

Coding:

```
newdf4.fillna(method='ffill',axis=1).head()
```

Output:

	anime_id	name	genre	type	episodes	rating	members	user_id
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572	673572
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266	151266

