PANDAS IMPORTANT FUNCTIONS

follow for more @Lovee Kumar

TOP 15 IMPORTANT PANDAS FUNCTIONS

pd.read_csv( )

pandas.read_csv() is used to **read a CSV** (Comma Separated Values) file and convert it into a pandas DataFrame.

CODE:

```
import pandas as pd
df = pd.read_csv("sales_data.csv")
```

OUTPUT:

```
df
```

0.2s

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Categor
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada	British Columbia	Accessorie
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada	British Columbia	Accessorie
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia	New South Wales	Accessorie
3	2016-03-23	23	March	2016	49	Adults (35-64)	M	Australia	New South Wales	Accessorie
4	2014-05-15	15	May	2014	47	Adults (35-64)	F	Australia	New South Wales	Accessorie
...	...	...	...	...	...	...	...	...	...	.

SWIPE

df.info()

df.info() is used to display a **summary of a data frame**, including the data types and the number of non-null values in each column.

CODE:

```
df.info()
```

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113036 entries, 0 to 113035
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Date              113036 non-null  object
 1   Day               113036 non-null  int64 
 2   Month             113036 non-null  object
 3   Year              113036 non-null  int64 
 4   Customer_Age      113036 non-null  int64 
 5   Age_Group         113036 non-null  object
 6   Customer_Gender   113036 non-null  object
 7   Country           113036 non-null  object
 8   State             113036 non-null  object
 9   Product_Category  113036 non-null  object
 10  Sub_Category      113036 non-null  object
 11  Product           113036 non-null  object
 12  Order_Quantity    113036 non-null  int64 
 13  Unit_Cost         113036 non-null  int64 
 14  Unit_Price        113036 non-null  int64 
 15  Profit            113036 non-null  int64 
 16  Cost              113036 non-null  int64 
 17  Revenue           113036 non-null  int64 
dtypes: int64(9), object(9)
memory usage: 15.5+ MB
```

SWIPE

df.describe()

df.describe() method in Pandas is used to generate **descriptive statistics of the columns** of a DataFrame. useful for getting a quick overview of the distribution of the data(mean, median, mode) and measures of dispersion (standard deviation, range, interquartile range).

CODE:

```
df.describe()
```

OUTPUT:

	Day	Year	Customer_Age	Order_Quantity	Unit_Cost	Unit_Price	
count	113036.000000	113036.000000	113036.000000	113036.000000	113036.000000	113036.000000	1
mean	15.665753	2014.401739	35.919212	11.901660	267.296366	452.938427	
std	8.781567	1.272510	11.021936	9.561857	549.835483	922.071219	
min	1.000000	2011.000000	17.000000	1.000000	1.000000	2.000000	
25%	8.000000	2013.000000	28.000000	2.000000	2.000000	5.000000	
50%	16.000000	2014.000000	35.000000	10.000000	9.000000	24.000000	
75%	23.000000	2016.000000	43.000000	20.000000	42.000000	70.000000	
max	31.000000	2016.000000	87.000000	32.000000	2171.000000	3578.000000	

SWIPE

df.assign()

df.assign() method in Pandas is used to a**dd new columns** to a DataFrame.

CODE:

```
df = df.assign(New_column = df['Customer_Age']>30)
```

OUTPUT:

t_Category	Sub_Category	Product	Order_Quantity	Unit_Cost	Unit_Price	Profit	Cost	Revenue	New_column
Accessories	Bike Racks	Hitch Rack - 4-Bike	8	45	120	590	360	950	False
Accessories	Bike Racks	Hitch Rack - 4-Bike	8	45	120	590	360	950	False
Accessories	Bike Racks	Hitch Rack - 4-Bike	23	45	120	1366	1035	2401	True
Accessories	Bike Racks	Hitch Rack - 4-Bike	20	45	120	1188	900	2088	True
Accessories	Bike Racks	Hitch Rack - 4-Bike	4	45	120	238	180	418	True
...	...	...	...	...	...	...	...	...	...
Clothing	Vests	Classic Vest, S	3	24	64	112	72	184	True
Clothing	Vests	Classic Vest, M	22	24	64	655	528	1183	False

Activate Windows
Go to Settings to activate Windows.

SWIPE

df.sample()

df.sample() function returns a random sample of rows from a DataFrame.By default, it returns one random row, but the number of rows can be specified as an argument. For example, **df.sample(5)** returns 5 random rows from the DataFrame. This function can be useful for getting a quick understanding of the distribution of values in a large dataset.

CODE:

```
df.sample(5)
```

OUTPUT:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender
45713	2016-01-08	8	January	2016	32	Young Adults (25-34)	M
109678	2014-01-05	5	January	2014	46	Adults (35-64)	F
38732	2014-01-24	24	January	2014	39	Adults (35-64)	M
85601	2015-08-16	16	August	2015	42	Adults (35-64)	F
27074	2013-11-22	22	November	2013	24	Youth (<25)	M

SWIPE

df.head()

df.head() function returns the first n (default 5) rows of a DataFrame.The n number of rows can be specified as an argument, for example **df.head(10)** returns the first 10 rows of the DataFrame.

CODE:

```
df.head()
```

OUTPUT:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender
0	2013-11-26	26	November	2013	19	Youth (<25)	M
1	2015-11-26	26	November	2015	19	Youth (<25)	M
2	2014-03-23	23	March	2014	49	Adults (35-64)	M
3	2016-03-23	23	March	2016	49	Adults (35-64)	M
4	2014-05-15	15	May	2014	47	Adults (35-64)	F

SWIPE

df.tail()

The **df.tail()** function returns the last n (default 5) rows of a DataFrame. The n number of rows can be specified as an argument, for example **df.tail(10)** returns the last 10 rows of the DataFrame.

CODE:

```
df.tail()
```

OUTPUT:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender
113031	2016-04-12	12	April	2016	41	Adults (35-64)	M
113032	2014-04-02	2	April	2014	18	Youth (<25)	M
113033	2016-04-02	2	April	2016	18	Youth (<25)	M
113034	2014-03-04	4	March	2014	37	Adults (35-64)	F
113035	2016-03-04	4	March	2016	37	Adults (35-64)	F

SWIPE

df.drop( )

df.drop() method in pandas is used to **remove rows or columns** from a data frame.

CODE:

```
df = df.drop(columns=['Customer_Age','Age_Group'])
```

OUTPUT: Customer_Age and Age_Group column is dropped from the data frame

```
df
```
✓ 0.2s

	Date	Day	Month	Year	Customer_Gender	Country	State	Product_Category	Sub_Category	Product	Order_Quantity
0	2013-11-26	26	November	2013	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8
1	2015-11-26	26	November	2015	M	Canada	British Columbia	Accessories	Bike Racks	Hitch Rack - 4-Bike	8
2	2014-03-23	23	March	2014	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	23
3	2016-03-23	23	March	2016	M	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	20
4	2014-05-15	15	May	2014	F	Australia	New South Wales	Accessories	Bike Racks	Hitch Rack - 4-Bike	4
...	...	...	...	...	...	...	...	...	...	...	...
113031	2016-04-12	12	April	2016	M	United Kingdom	England	Clothing	Vests	Classic Vest, S	3
	2014-									Classic	

SWIPE

df.dropna( )

df.dropna() method in Python is used to **remove any rows that contain missing values** (i.e. NaN) from a DataFrame. This can be useful for cleaning data before analysis or modeling.

CODE:

```
df.dropna(subset='Year')
```

OUTPUT:

Before

```
df
```
✓ 0.3s

	Date	Day	Month	Year	Cus
0	2013-11-26	26	November	2013	
1	2015-11-26	26	November	2015	
2	2014-03-23	23	March	2014	
3	2016-03-23	23	March	NaN	
4	2014-05-15	15	May	2014	

After

	Date	Day	Month	Year
0	2013-11-26	26	November	2013
1	2015-11-26	26	November	2015
2	2014-03-23	23	March	2014
4	2014-05-15	15	May	2014
5	2016-05-15	15	May	2016

SWIPE

df.query()

df.query() is used to **filter rows of a DataFrame based on a condition.**

CODE:

```
df = df.query('Customer_Age > 30 and Country == "Canada"')
```

OUTPUT:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country	State	Product_Category	Sub_Category
24	2013-08-25	25	August	2013	49	Adults (35-64)	M	Canada	British Columbia	Accessories	Bike Racks
25	2015-08-25	25	August	2015	49	Adults (35-64)	M	Canada	British Columbia	Accessories	Bike Racks
26	2013-12-26	26	December	2013	49	Adults (35-64)	M	Canada	British Columbia	Accessories	Bike Racks
27	2015-12-26	26	December	2015	49	Adults (35-64)	M	Canada	British Columbia	Accessories	Bike Racks
28	2014-01-02	2	January	2014	48	Adults (35-64)	F	Canada	British Columbia	Accessories	Bike Racks
...	...	...	...	...	...	...	...	...	...	...	...
12885	2016-07-05	5	July	2016	38	Adults (35-64)	M	Canada	British Columbia	Clothing	Vests
12952	2013-08-18	18	August	2013	31	Young Adults (25-34)	F	Canada	British Columbia	Clothing	Vests
12953	2015-08-18	18	August	2015	31	Young Adults (25-	F	Canada	British Columbia	Clothing	Vests

SWIPE

df.sort_values()

df.sort_values() method in Pandas is used to **sort the rows** of a DataFrame based on the values of columns.

CODE:

```
df.sort_values(['Year'])
```

OUTPUT:

	Date	Day	Month	Year
56517	2011-12-23	23	December	2011
69025	2011-08-04	4	August	2011
69027	2011-08-06	6	August	2011
69029	2011-08-17	17	August	2011
69031	2011-10-22	22	October	2011
...	...	...	...	...

CODE:

```
df.sort_values(['Year'],ascending=False)
```

OUTPUT:

	Date	Day	Month	Year	Customer
113035	2016-03-04	4	March	2016	
60793	2016-05-23	23	May	2016	
25939	2016-02-09	9	February	2016	
97644	2016-02-04	4	February	2016	
25937	2016-01-16	16	January	2016	
...	...	...	...	...	
58893	2011-07-04	4	July	2011	
62575	2011-	2	September	2011	

SWIPE

df.groupby().sum()

df.groupby() is used to **group a data frame** by one or more columns. The result is a new data frame that has the **grouped columns** as the index, and the other columns are aggregated using a **specified aggregation method.**

CODE:

```
df.groupby('Year').sum()['Order_Quantity']
#df.groupby().mean()
#df.groupby().median()
#df.groupby().max()
#df.groupby().min()
```

OUTPUT:

```
Year
2011       5260
2012       5354
2013     294787
2014     379585
2015     289517
2016     370813
Name: Order_Quantity, dtype: int64
```

SWIPE

df.merge()

df.merge() is used to **combine two or more** DataFrames into a **single data frame**. The function works by joining the DataFrames on one or more common columns, similar to a SQL JOIN operation

CODE:

```
import pandas as pd

# Create two example DataFrames
df1 = pd.DataFrame({'key': ['A', 'B', 'C', 'D'],'value': [1, 2, 3, 4]})
df2 = pd.DataFrame({'key': ['B', 'D', 'E', 'F'],'value': [5, 6, 7, 8]})
# Merge the DataFrames on the 'key' column
merged_df = df1.merge(df2, on='key')
# Print the resulting DataFrame
print(merged_df)
```

OUTPUT:

```
  key  value_x  value_y
0   B        2        5
1   D        4        6
```

SWIPE

df.rename()

df.rename() is used to rename one or more columns in a data frame.

CODE:

```
df.rename(columns={'Customer_Age': 'Age'})
```

OUTPUT:

	Date	Day	Month	Year	Age	A
0	2013-11-26	26	November	2013	19	\
1	2015-11-26	26	November	2015	19	\
2	2014-03-23	23	March	2014	49	
3	2016-03-23	23	March	2016	49	
4	2014-05-15	15	May	2014	47	

SWIPE

df.to_csv()

df.to_csv() is used to save a data frame(export data) to a CSV (Comma Separated Values) file.

CODE:

```
df.to_csv('Myfile.csv')
```

OUTPUT:

Name	Date	Type	Size
Project	1/13/2023 7:52 AM	File folder	
MFDM	6/21/2022 7:29 PM	File folder	
Myfile	1/22/2023 10:56 PM	Microsoft Excel C...	15,548 KB
DummyData	1/20/2023 11:18 PM	Microsoft Excel C...	119 KB
Dummy	1/20/2023 11:12 PM	Microsoft Excel C...	121 KB
Book1	1/20/2023 11:11 PM	Microsoft Excel C...	1 KB
pandas.ipynb	1/20/2023 11:04 PM	IPYNB File	1 KB

SWIPE

follow for more @Lovee Kumar