

basic-pandas-key-features

February 14, 2023

Name= Pushpraj Singh Rathore Topic= Pandas

1 How to create a series with help of Pandas

```
[1]: import pandas as pd  
s=pd.Series([1,2,3,4])  
print(s)
```

```
0    1  
1    2  
2    3  
3    4  
dtype: int64
```

```
[138]: # with index  
s1=pd.Series([2,5,6],index=["a","b","c"])  
print(s1)
```

```
a    2  
b    5  
c    6  
dtype: int64
```

```
[139]: #change the datatype of Series  
s2=pd.Series([2,5,4],index=["A","B","C"],dtype=float)  
print(s2)
```

```
A    2.0  
B    5.0  
C    4.0  
dtype: float64
```

```
[4]: # create a empty series  
s=pd.Series([])  
print(s)
```

```
Series([], dtype: float64)
```

```
C:\Users\LENOVO\AppData\Local\Temp\ipykernel_8120\2262464387.py:2:
FutureWarning: The default dtype for empty Series will be 'object' instead of
'float64' in a future version. Specify a dtype explicitly to silence this
warning.
s=pd.Series([])
```

```
[9]: s=pd.Series(0.5,index=[1,2,3]) # without use of Square Bracket create Series
      ↪with same value with deffrent index values
      print(s)
```

```
1    0.5
2    0.5
3    0.5
dtype: float64
```

2 Create A Series With Help of dictionary

```
[10]: s=pd.Series({"A":5,"B":6,"C":8})
      print(s)
```

```
A    5
B    6
C    8
dtype: int64
```

3 How to create Data Frame

```
[15]: l=[1,2,3,4]
      df=pd.DataFrame(l)
      print(df)
```

```
0
0  1
1  2
2  3
3  4
```

```
[14]: l=[1,2,3,4] #create a datafreame with column name A with help of Dictionary
      df=pd.DataFrame({"A":l})
      print(df)
```

```
   A
0  1
1  2
2  3
3  4
```

```
[22]: l=[[1,2,3,4],[4,5,6,9]] #with help of Dictionary
df=pd.DataFrame({"A":l,"B":l,"c":l,"D":l})
print(df)
```

```
      A      B      c      D
0  [1, 2, 3, 4]  [1, 2, 3, 4]  [1, 2, 3, 4]  [1, 2, 3, 4]
1  [4, 5, 6, 9]  [4, 5, 6, 9]  [4, 5, 6, 9]  [4, 5, 6, 9]
```

```
[34]: df=pd.DataFrame({"A":[1,2,2],"B":[4,5,9]})
df
```

```
[34]:   A  B
0  1  4
1  2  5
2  2  9
```

```
[72]: l=({"A":1,"B":2},{"A":5,"B":6})
df1=pd.DataFrame(l)
df1
```

```
[72]:   A  B
0  1  2
1  5  6
```

```
[78]: l=({"A":1,"B":2},{"A":4,"B":6})
df2=pd.DataFrame(l)
df2
```

```
[78]:   A  B
0  1  2
1  4  6
```

```
[79]: # How to merge to data Frame
pd.merge(df1,df2)
```

```
[79]:   A  B
0  1  2
```

```
[173]: # A Value will not matched paticualr columns and not satisfy a condication
↪system auotomatic suggest NAN
l=({"A":1,"B":2},{"B":6})
df2=pd.DataFrame(l)
df2
```

```
[173]:   A  B
0  1.0  2
1  NaN  6
```

4 How to Read a data In pandas

first you uplod a data set in jupyter notebook and copy a name of data set and pest

```
[95]: data=pd.read_csv("car data.csv")
data.head()
```

```
[95]: Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0      ritz    2014           3.35           5.59        27000    Petrol
1      sx4     2013           4.75           9.54        43000    Diesel
2      ciaz    2017           7.25           9.85         6900    Petrol
3  wagon r    2011           2.85           4.15         5200    Petrol
4    swift    2014           4.60           6.87        42450    Diesel

Seller_Type  Transmission  Owner
0      Dealer           Manual    0
1      Dealer           Manual    0
2      Dealer           Manual    0
3      Dealer           Manual    0
4      Dealer           Manual    0
```

```
[101]: data.info() # short Information about Data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Car_Name              301 non-null   object
1   Year                  301 non-null   int64
2   Selling_Price         301 non-null   float64
3   Present_Price         301 non-null   float64
4   Kms_Driven            301 non-null   int64
5   Fuel_Type             301 non-null   object
6   Seller_Type           301 non-null   object
7   Transmission          301 non-null   object
8   Owner                 301 non-null   int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

```
[144]: # how to find null values
data.isnull().head()
```

```
[144]: Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0      False  False           False           False        False    False
1      False  False           False           False        False    False
2      False  False           False           False        False    False
3      False  False           False           False        False    False
```

4	False	False	False	False	False	False
---	-------	-------	-------	-------	-------	-------

	Seller_Type	Transmission	Owner
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False

```
[148]: # count a null values
data.isnull().sum()
```

```
[148]: Car_Name      0
Year              0
Selling_Price     0
Present_Price     0
Kms_Driven        0
Fuel_Type         0
Seller_Type       0
Transmission      0
Owner             0
dtype: int64
```

```
[145]: # TO find a unique values on patricular columns
data.nunique()
```

```
[145]: Car_Name      98
Year              16
Selling_Price     156
Present_Price     147
Kms_Driven        206
Fuel_Type         3
Seller_Type       2
Transmission      2
Owner             3
dtype: int64
```

```
[147]: data["Car_Name"].nunique
```

```
[147]: 98
```

```
[151]: # count a unique values on
data["Car_Name"].value_counts().head(50)
```

```
[151]: city              26
corolla altis         16
verna                 14
```

fortuner	11
brio	10
ciaz	9
innova	9
i20	9
grand i10	8
jazz	7
amaze	7
Royal Enfield Classic 350	7
ertiga	6
eon	6
sx4	6
alto k10	5
i10	5
swift	5
Bajaj Pulsar 150	4
Royal Enfield Thunder 350	4
ritz	4
wagon r	4
etios liva	4
dzire	4
xcent	3
etios cross	3
Royal Enfield Thunder 500	3
TVS Apache RTR 160	3
Yamaha FZ S V 2.0	3
creta	3
Honda CB Hornet 160R	3
etios g	3
Bajaj Avenger 220	3
Bajaj Pulsar NS 200	3
Bajaj Discover 125	2
Bajaj Pulsar 220 F	2
TVS Apache RTR 180	2
Yamaha FZ v 2.0	2
Honda Aactive 4G	2
Hero Extreme	2
Honda CB twister	2
Hero Splender iSmart	2
Aactive 3g	2
Hero Passion Pro	2
Honda CB Shine	2
Bajaj Avenger 220 dtsi	2
Honda CBR 150	2
elantra	2
Royal Enfield Classic 500	2
KTM RC200	2

Name: Car_Name, dtype: int64

5 how to Create a new columns name

6 Rename a Columns

```
[84]: data=pd.read_csv("car data.csv")
data.head()
```

```
[84]:   Car_Name  Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014         3.35         5.59       27000    Petrol
1    sx4    2013         4.75         9.54       43000    Diesel
2    ciaz    2017         7.25         9.85        6900    Petrol
3  wagon r    2011         2.85         4.15        5200    Petrol
4   swift    2014         4.60         6.87       42450    Diesel

   Seller_Type  Transmission  Owner
0     Dealer         Manual      0
1     Dealer         Manual      0
2     Dealer         Manual      0
3     Dealer         Manual      0
4     Dealer         Manual      0
```

```
[86]: data.rename(columns={"Car_Name":"CarName","Selling_Price":"SP"},inplace=True).
      ↪head()
```

```
[86]:   CarName  Year   SP  Present_Price  Kms_Driven  Fuel_Type  Seller_Type  \
0    ritz    2014  3.35         5.59       27000    Petrol     Dealer
1    sx4    2013  4.75         9.54       43000    Diesel     Dealer
2    ciaz    2017  7.25         9.85        6900    Petrol     Dealer
3  wagon r    2011  2.85         4.15        5200    Petrol     Dealer
4   swift    2014  4.60         6.87       42450    Diesel     Dealer

   Transmission  Owner
0     Manual      0
1     Manual      0
2     Manual      0
3     Manual      0
4     Manual      0
```

7 find a mean,sum,min,max,count,std,var

```
[88]: data["Kms_Driven"].mean()
```

```
[88]: 36947.20598006644
```

```
[89]: data["Kms_Driven"].max()
```

```
[89]: 500000
```

```
[90]: data["Kms_Driven"].min()
```

```
[90]: 500
```

```
[91]: data["Kms_Driven"].sum()
```

```
[91]: 11121109
```

```
[92]: data["Kms_Driven"].count()
```

```
[92]: 301
```

```
[93]: data["Kms_Driven"].std()
```

```
[93]: 38886.88388206788
```

```
[94]: data["Kms_Driven"].var()
```

```
[94]: 1512189738.0574307
```

8 Some Filltering Opertaions

```
[100]: data.head() # head Show starting 5 rows
```

```
[100]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
0	ritz	2014	3.35	5.59	27000	Petrol	
1	sx4	2013	4.75	9.54	43000	Diesel	
2	ciaz	2017	7.25	9.85	6900	Petrol	
3	wagon r	2011	2.85	4.15	5200	Petrol	
4	swift	2014	4.60	6.87	42450	Diesel	

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
1	Dealer	Manual	0
2	Dealer	Manual	0
3	Dealer	Manual	0
4	Dealer	Manual	0

```
[99]: data.tail() # tail show ending 5 rows
```

```
[99]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
296	city	2016	9.50	11.6	33988	Diesel	

297	brio	2015	4.00	5.9	60000	Petrol
298	city	2009	3.35	11.0	87934	Petrol
299	city	2017	11.50	12.5	9000	Diesel
300	brio	2016	5.30	5.9	5464	Petrol

	Seller_Type	Transmission	Owner
296	Dealer	Manual	0
297	Dealer	Manual	0
298	Dealer	Manual	0
299	Dealer	Manual	0
300	Dealer	Manual	0

9 How to cahange a data type of columns

```
[109]: data=pd.read_csv("car data.csv",dtype={"Year":float})
data.head()
```

```
[109]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         3.35         5.59        27000    Petrol
1    sx4    2013.0         4.75         9.54        43000    Diesel
2    ciaz    2017.0         7.25         9.85         6900    Petrol
3  wagon r    2011.0         2.85         4.15         5200    Petrol
4    swift    2014.0         4.60         6.87        42450    Diesel
```

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
1	Dealer	Manual	0
2	Dealer	Manual	0
3	Dealer	Manual	0
4	Dealer	Manual	0

```
[110]: data.fillna # Headling A missing Value
```

```
[110]: <bound method DataFrame.fillna of      Car_Name    Year  Selling_Price
Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         3.35         5.59        27000    Petrol
1    sx4    2013.0         4.75         9.54        43000    Diesel
2    ciaz    2017.0         7.25         9.85         6900    Petrol
3  wagon r    2011.0         2.85         4.15         5200    Petrol
4    swift    2014.0         4.60         6.87        42450    Diesel
..    ...    ...         ...         ...         ...         ...
296   city    2016.0         9.50        11.60        33988    Diesel
297   brio    2015.0         4.00         5.90        60000    Petrol
298   city    2009.0         3.35        11.00        87934    Petrol
299   city    2017.0        11.50        12.50         9000    Diesel
300   brio    2016.0         5.30         5.90         5464    Petrol
```

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
1	Dealer	Manual	0
2	Dealer	Manual	0
3	Dealer	Manual	0
4	Dealer	Manual	0
..
296	Dealer	Manual	0
297	Dealer	Manual	0
298	Dealer	Manual	0
299	Dealer	Manual	0
300	Dealer	Manual	0

[301 rows x 9 columns]>

10 How to replace a row value

```
[111]: data.head(2)
```

```
[111]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         3.35         5.59         27000    Petrol
1    sx4    2013.0         4.75         9.54         43000    Diesel

      Seller_Type  Transmission  Owner
0      Dealer      Manual      0
1      Dealer      Manual      0
```

```
[114]: data.replace("Manual", "NotManual").head()
```

```
[114]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         3.35         5.59         27000    Petrol
1    sx4    2013.0         4.75         9.54         43000    Diesel
2    ciaz    2017.0         7.25         9.85          6900    Petrol
3  wagon r    2011.0         2.85         4.15          5200    Petrol
4    swift    2014.0         4.60         6.87         42450    Diesel

      Seller_Type  Transmission  Owner
0      Dealer    NotManual      0
1      Dealer    NotManual      0
2      Dealer    NotManual      0
3      Dealer    NotManual      0
4      Dealer    NotManual      0
```

```
[116]: data.replace(3.35,55).head()
```

```
[116]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         55.00         5.59         27000    Petrol
1    sx4    2013.0         4.75         9.54         43000    Diesel
2    ciaz    2017.0         7.25         9.85          6900    Petrol
3  wagon r    2011.0         2.85         4.15          5200    Petrol
4    swift    2014.0         4.60         6.87         42450    Diesel

Seller_Type  Transmission  Owner
0    Dealer      Manual      0
1    Dealer      Manual      0
2    Dealer      Manual      0
3    Dealer      Manual      0
4    Dealer      Manual      0
```

```
[118]: data.replace([2014.0,2017.0],8).head()
```

```
[118]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz      8.0         3.35         5.59         27000    Petrol
1    sx4    2013.0         4.75         9.54         43000    Diesel
2    ciaz      8.0         7.25         9.85          6900    Petrol
3  wagon r    2011.0         2.85         4.15          5200    Petrol
4    swift      8.0         4.60         6.87         42450    Diesel

Seller_Type  Transmission  Owner
0    Dealer      Manual      0
1    Dealer      Manual      0
2    Dealer      Manual      0
3    Dealer      Manual      0
4    Dealer      Manual      0
```

```
[123]: data.head(1)
```

```
[123]: Car_Name    Year  Selling_Price  Present_Price  Kms_Driven  Fuel_Type  \
0    ritz    2014.0         3.35         5.59         27000    Petrol

Seller_Type  Transmission  Owner
0    Dealer      Manual      0
```

11 how to check paticular rows

loc=with help of loc we find a full rows on patricular indexing iloc=with help of iloc we find paticular value on paticular rows with help of paticular indexing

```
[127]: data.loc[[0,5]]
```

```
[127]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
0	ritz	2014.0	3.35	5.59	27000	Petrol	
5	vitara brezza	2018.0	9.25	9.83	2071	Diesel	

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
5	Dealer	Manual	0

```
[132]: data.loc[[1,2]]
```

```
[132]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
1	sx4	2013.0	4.75	9.54	43000	Diesel	
2	ciaz	2017.0	7.25	9.85	6900	Petrol	

	Seller_Type	Transmission	Owner
1	Dealer	Manual	0
2	Dealer	Manual	0

```
[137]: data.iloc[1,3]
```

```
[137]: 9.54
```

```
[152]: # count a null values
data.isnull().sum()
```

```
[152]: Car_Name      0
Year              0
Selling_Price     0
Present_Price     0
Kms_Driven        0
Fuel_Type         0
Seller_Type       0
Transmission      0
Owner             0
dtype: int64
```

12 To Find a values on particular columns

```
[153]: # To Find a values on particular columns
data.groupby("Car_Name").get_group("ritz")
```

```
[153]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
0	ritz	2014.0	3.35	5.59	27000	Petrol	
30	ritz	2012.0	3.10	5.98	51439	Diesel	
31	ritz	2011.0	2.35	4.89	54200	Petrol	
46	ritz	2013.0	2.65	4.89	64532	Petrol	

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
30	Dealer	Manual	0
31	Dealer	Manual	0
46	Dealer	Manual	0

```
[155]: data.groupby("Fuel_Type").get_group("Petrol").head()
```

```
[155]:
```

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	\
0	ritz	2014.0	3.35	5.59	27000	Petrol	
2	ciaz	2017.0	7.25	9.85	6900	Petrol	
3	wagon r	2011.0	2.85	4.15	5200	Petrol	
6	ciaz	2015.0	6.75	8.12	18796	Petrol	
10	alto 800	2017.0	2.85	3.60	2135	Petrol	

	Seller_Type	Transmission	Owner
0	Dealer	Manual	0
2	Dealer	Manual	0
3	Dealer	Manual	0
6	Dealer	Manual	0
10	Dealer	Manual	0

13 Comparison

```
[ ]: data[data["Selling_Price"]==3.35 & (data["Kms_Driven"]==27000)]
```

```
[ ]: data[data["Selling_Price"]==3.35 | (data["Kms_Driven"]==27000)]
```