In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
ds=pd.DataFrame(np.random.rand(10,4),index=pd.date_range('1/1/2000',periods=10),columns=list('ABCD'))
```

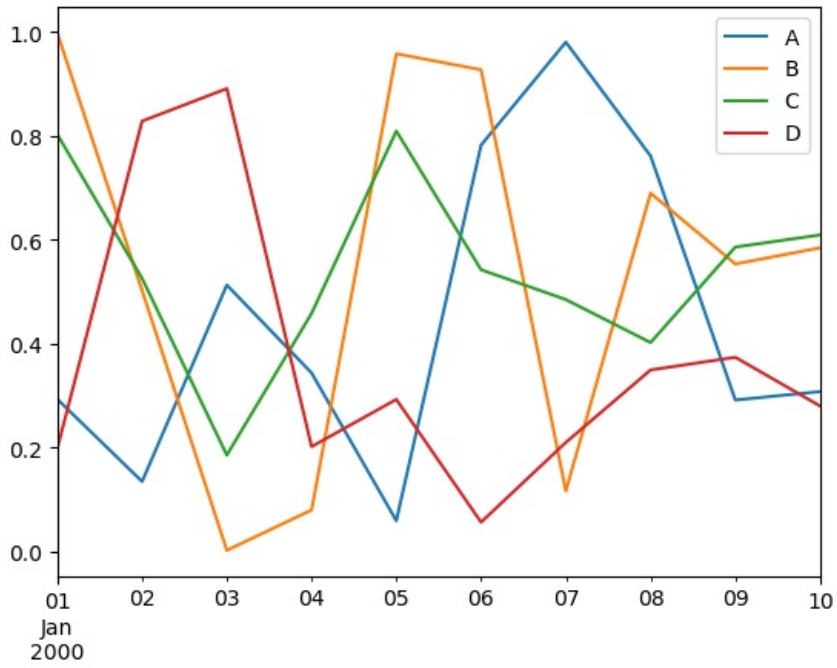
In [3]:
```
ds
```

Out[3]:

	A	B	C	D
2000-01-01	0.293128	0.997942	0.802750	0.198676
2000-01-02	0.134657	0.502403	0.525831	0.828573
2000-01-03	0.513041	0.002123	0.185465	0.891156
2000-01-04	0.343672	0.079846	0.458337	0.201819
2000-01-05	0.059177	0.958311	0.809499	0.292711
2000-01-06	0.782318	0.927415	0.542442	0.056322
2000-01-07	0.980649	0.116358	0.484993	0.209913
2000-01-08	0.761539	0.689791	0.402211	0.349444
2000-01-09	0.291555	0.553502	0.586065	0.373840
2000-01-10	0.307786	0.584556	0.609329	0.280115

In [4]:
```
ds.plot()
```

Out[4]:
```
<AxesSubplot:>
```



In [5]:
```
ds
```

Out[5]:

	A	B	C	D
2000-01-01	0.293128	0.997942	0.802750	0.198676
2000-01-02	0.134657	0.502403	0.525831	0.828573
2000-01-03	0.513041	0.002123	0.185465	0.891156
2000-01-04	0.343672	0.079846	0.458337	0.201819
2000-01-05	0.059177	0.958311	0.809499	0.292711
2000-01-06	0.782318	0.927415	0.542442	0.056322
2000-01-07	0.980649	0.116358	0.484993	0.209913
2000-01-08	0.761539	0.689791	0.402211	0.349444
2000-01-09	0.291555	0.553502	0.586065	0.373840
2000-01-10	0.307786	0.584556	0.609329	0.280115

In [6]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
```

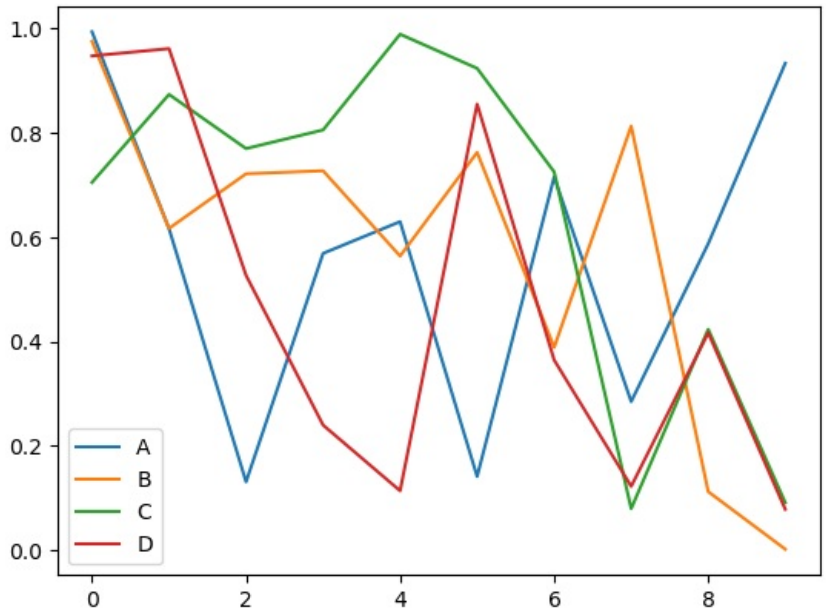
In [7]:
```
ds
```

Out[7]:

	A	B	C	D
0	0.993657	0.975484	0.704950	0.947386
1	0.616375	0.616804	0.873608	0.961373
2	0.131093	0.721497	0.769770	0.526827
3	0.568930	0.727212	0.805517	0.239888
4	0.629776	0.563814	0.988959	0.113951
5	0.141308	0.762552	0.923570	0.854960
6	0.715347	0.388782	0.725137	0.364549
7	0.284927	0.812854	0.079738	0.122458
8	0.587553	0.112268	0.423473	0.416761
9	0.933294	0.002026	0.091181	0.078523

In [8]:
```
ds.plot()
```

Out[8]:
```
<AxesSubplot:>
```



In [9]:
```
ds1=pd.DataFrame(np.random.rand(10,4),index=pd.date_range('1/1/2000',periods=10),columns=list('ABCD'))
```

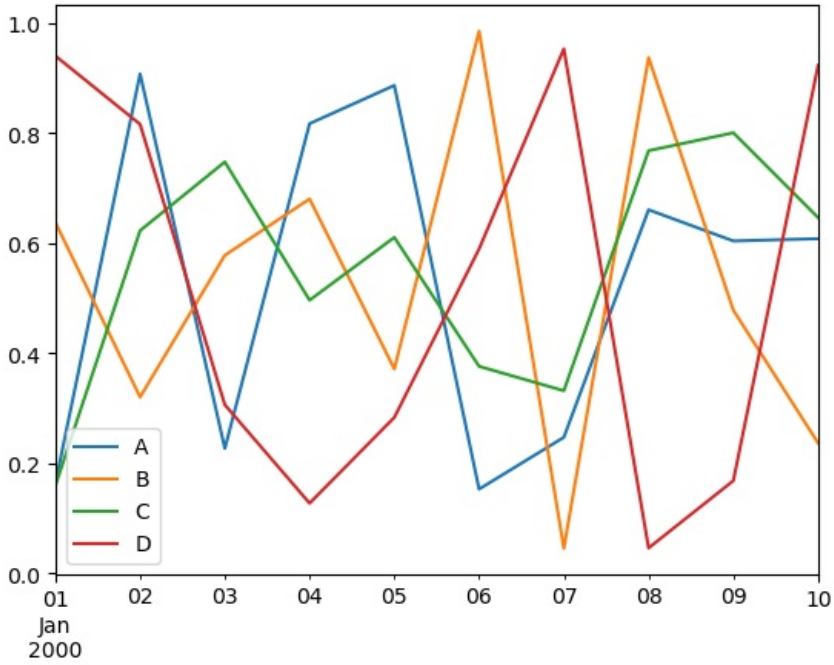
In [11]:
```
ds1
```

Out[11]:

	A	B	C	D
2000-01-01	0.163145	0.638107	0.159207	0.940753
2000-01-02	0.907934	0.320350	0.623062	0.816375
2000-01-03	0.226881	0.578176	0.747936	0.306374
2000-01-04	0.817495	0.680689	0.496557	0.127449
2000-01-05	0.887012	0.371525	0.610735	0.283353
2000-01-06	0.153050	0.985661	0.376268	0.589672
2000-01-07	0.247224	0.045397	0.331958	0.953454
2000-01-08	0.660937	0.937650	0.768299	0.046016
2000-01-09	0.604213	0.478440	0.800857	0.168312
2000-01-10	0.608330	0.236952	0.646737	0.923306

In [12]:
```
ds1.plot()
```

Out[12]:
```
<AxesSubplot:>
```



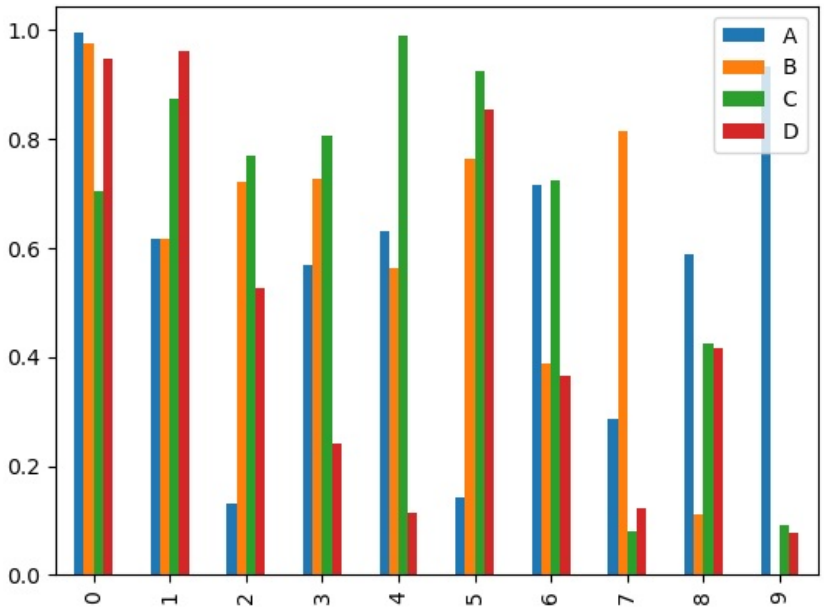
In [13]:
```
ds
```

Out[13]:

	A	B	C	D
0	0.993657	0.975484	0.704950	0.947386
1	0.616375	0.616804	0.873608	0.961373
2	0.131093	0.721497	0.769770	0.526827
3	0.568930	0.727212	0.805517	0.239888
4	0.629776	0.563814	0.988959	0.113951
5	0.141308	0.762552	0.923570	0.854960
6	0.715347	0.388782	0.725137	0.364549
7	0.284927	0.812854	0.079738	0.122458
8	0.587553	0.112268	0.423473	0.416761
9	0.933294	0.002026	0.091181	0.078523

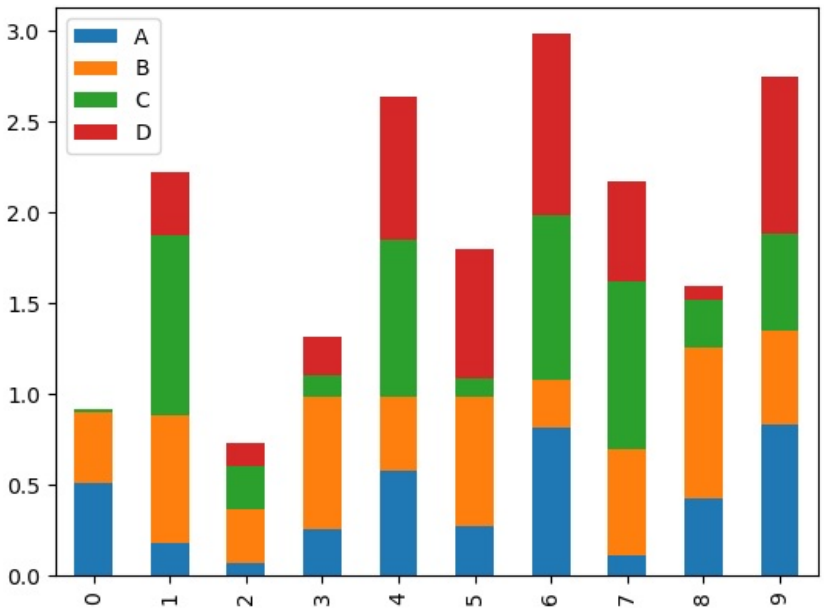
In [15]:
```
ds.plot.bar()
```

Out[15]:
```
<AxesSubplot:>
```



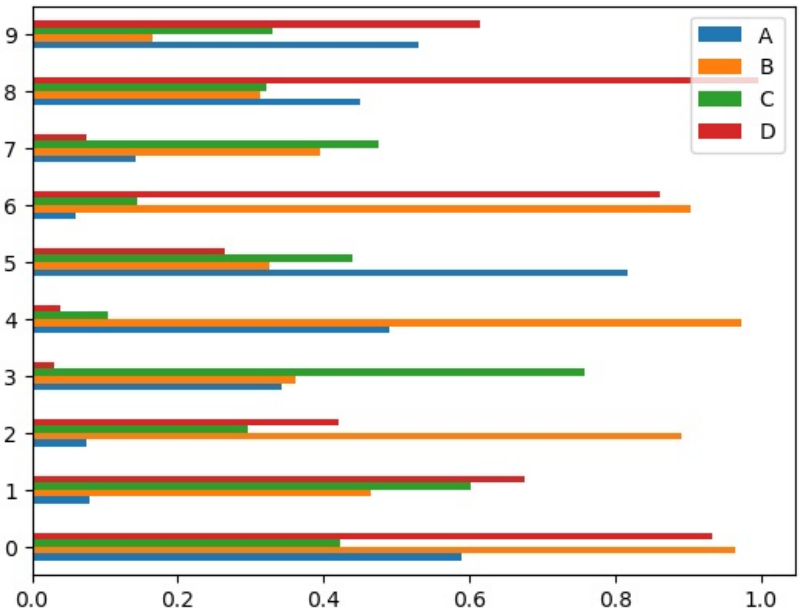
In [17]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
ds.plot.bar(stacked=True)
```

Out[17]:
```
<AxesSubplot:>
```



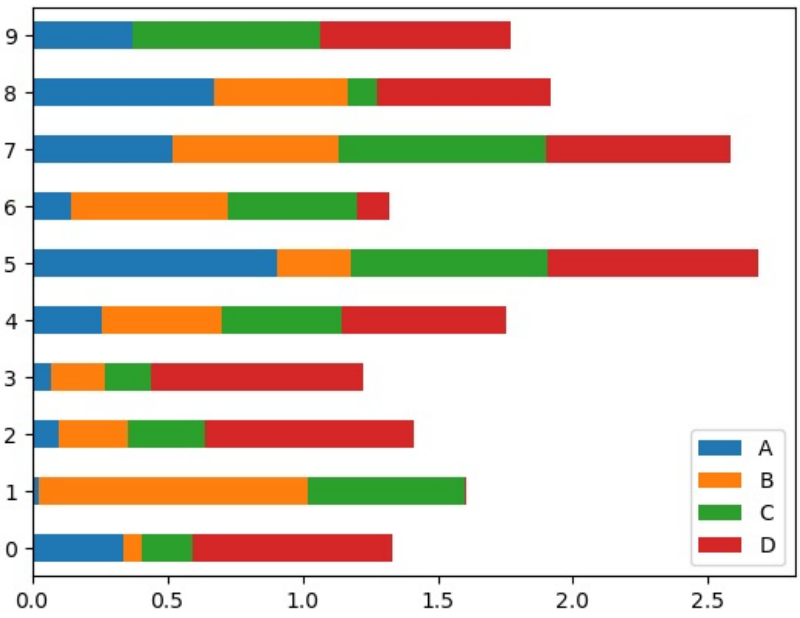
In [18]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
ds.plot.barh()
```

Out[18]:
```
<AxesSubplot:>
```



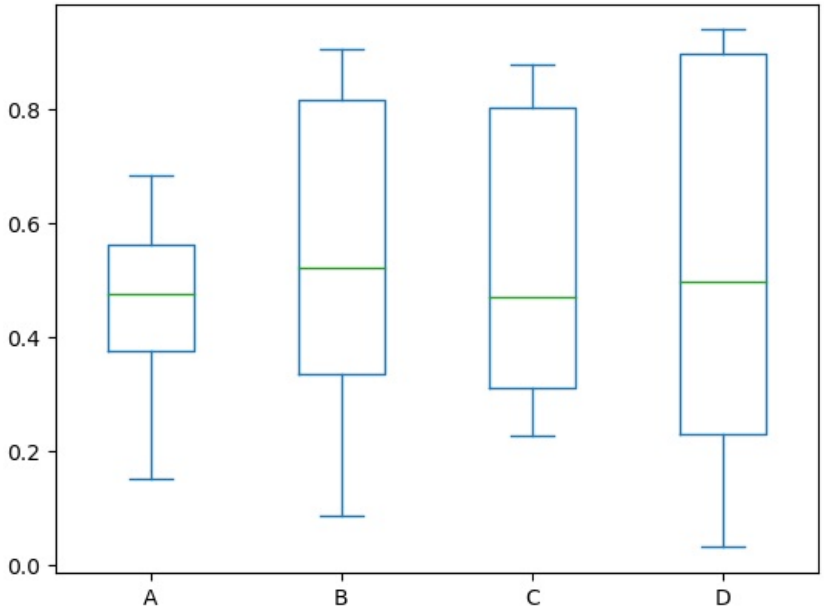
In [19]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
ds.plot.barh(stacked=True)
```

Out[19]:
```
<AxesSubplot:>
```



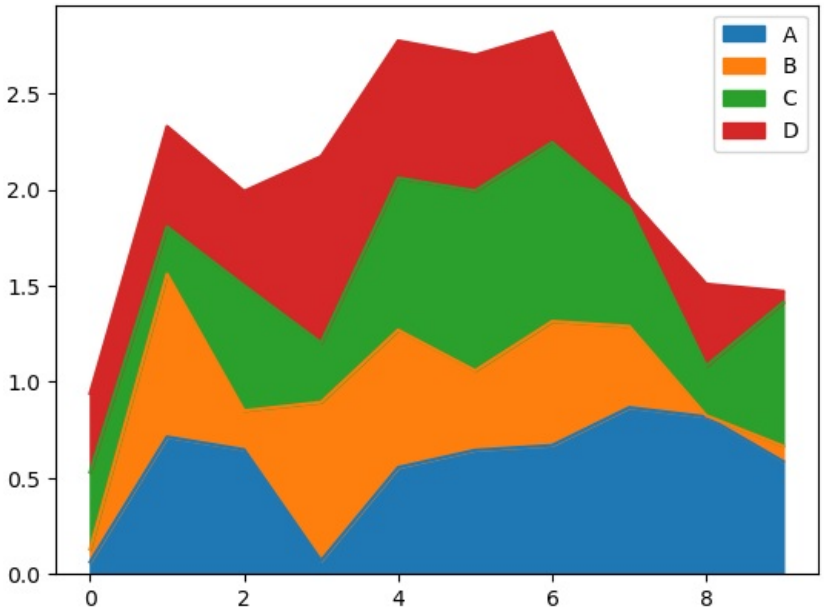
In [20]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
ds.plot.box()
```

Out[20]:
```
<AxesSubplot:>
```



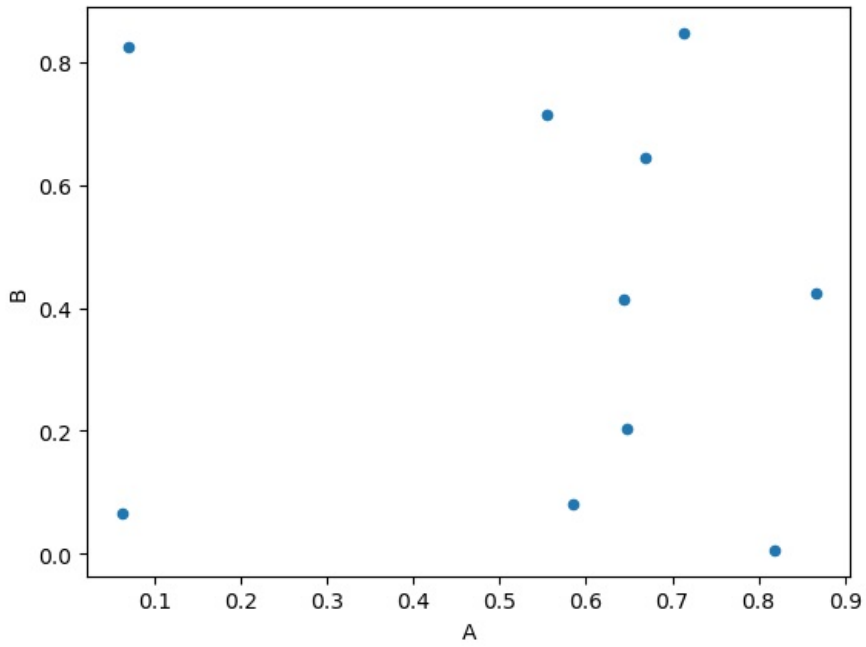
In [21]:
```
ds=pd.DataFrame(np.random.rand(10,4),columns=['A','B','C','D'])
ds.plot.area()
```

Out[21]:
```
<AxesSubplot:>
```



In [23]:
```
ds.plot.scatter(x='A',y='B')
```

Out[23]:
```
<AxesSubplot:xlabel='A', ylabel='B'>
```



In [25]:
```
ds.plot.(x='A',y='B')
```

```
  File "C:\Users\Nitesh\AppData\Local\Temp\ipykernel_14508\475949452.py", line 1
    ds.plot.3d(x='A',y='B')
            ^
SyntaxError: invalid syntax
```

In [28]:
```
ds2=pd.DataFrame(np.random.rand(6,3),columns=['Name','Age','Dept'])
```

In [29]:
```
ds2
```

Out[29]:

	Name	Age	Dept
0	0.337276	0.924176	0.603409
1	0.648923	0.877318	0.910074
2	0.130175	0.993290	0.805091
3	0.356688	0.137746	0.172263
4	0.987216	0.704289	0.184192
5	0.562687	0.142515	0.949205

In [35]:
```
ds=pd.DataFrame({'Name':['Jas','Loga','Sai','Chethan','Palla'],
                 'Age':[19,19,20,20,19],
                 'Dept':['CSE','CSE','CSE','CSE','CSE']})
```

In [36]:
```
ds
```

Out[36]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE

In [38]:
```
ds['Age']>20
```

Out[38]:
```
0    False
1    False
2    False
3    False
4    False
Name: Age, dtype: bool
```

In [44]:
```
ds[(ds['Age']>=19) & (ds['Age']<20)]
```

Out[44]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
4	Palla	19	CSE

In [46]:
```
ds.append('Name'='lokesh')
ds.append('Age'=21)
ds.append('Dept'='EEE')
```

```
  File "C:\Users\Nitesh\AppData\Local\Temp\ipykernel_14508\2503399675.py", line 1
    ds.append('Name'='lokesh')
                    ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

In [47]:
```
ds.insert(5,'Name'='lokesh','Age'=21,'Dept'='EEE')
```

```
  File "C:\Users\Nitesh\AppData\Local\Temp\ipykernel_14508\2709018093.py", line 1
    ds.insert(6,'Name'='lokesh','Age'=21,'Dept'='EEE')
                      ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

In [48]:
```
ds.insert(5,'lokesh',21,'EEE')
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14508\275546714.py in <module>
----> 1 ds.insert(5,'lokesh',21,'EEE')

~\anaconda3\lib\site-packages\pandas\core\frame.py in insert(self, loc, column, value, allow_duplicates)
   4446
   4447         value = self._sanitize_column(value)
-> 4448         self._mgr.insert(loc, column, value)
   4449
   4450     def assign(self, **kwargs) -> DataFrame:

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in insert(self, loc, item, value)
   1238         """
   1239         # insert to the axis; this could possibly raise a TypeError
-> 1240         new_axis = self.items.insert(loc, item)
   1241
   1242         if value.ndim == 2:

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in insert(self, loc, item)
   6614             # special-casing dt64/td64 https://github.com/numpy/numpy/issues/12550
   6615             casted = arr.dtype.type(item)
-> 6616             new_values = np.insert(arr, loc, casted)
   6617
   6618         else:

<__array_function__ internals> in insert(*args, **kwargs)

~\anaconda3\lib\site-packages\numpy\lib\function_base.py in insert(arr, obj, values, axis)
   4702         index = indices.item()
   4703         if index < -N or index > N:
-> 4704             raise IndexError(
   4705                 "index %i is out of bounds for axis %i with "
   4706                 "size %i" % (obj, axis, N))

IndexError: index 5 is out of bounds for axis 0 with size 3
```

In [49]:
```
ds.loc[5]=['lokesh',21,'EEE']
```

In [50]:
```
ds
```

Out[50]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE

In [53]:
```
ds.head()
```

```
  File "C:\Users\Nitesh\AppData\Local\Temp\ipykernel_14508\4249902344.py", line 1
    ds.head()\\to display first 5 rows\\
             ^
SyntaxError: unexpected character after line continuation character
```

In [54]:
```
ds.tail()
```

Out[54]:

	Name	Age	Dept
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE

In [57]:
```
ds.drop(5)
```

Out[57]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE

In [62]:
```
ds.loc[6]=['NaN',23,'NaN']
```

In [63]:
```
ds.loc[7]=['Nikki','NaN','NaN']
```

In [64]:
```
ds
```

Out[64]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN
7	Nikki	NaN	NaN

In [65]:
```
ds
```

Out[65]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN
7	Nikki	NaN	NaN

In [66]:
```
ds2=ds
```

In [67]:
```
ds
```

Out[67]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN
7	Nikki	NaN	NaN

In [68]:
```
ds2
```

Out[68]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN
7	Nikki	NaN	NaN

In [71]:
```
ds.fillna(method='ffill',axis=1)
```

Out[71]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN
7	Nikki	NaN	NaN

In [72]:
```
ds.drop(6)
```

Out[72]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
7	Nikki	NaN	NaN

In [73]:
```
ds.drop(7)
```

Out[73]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN

In [74]:
```
ds.drop(7,inplace=True)
```

In [75]:
```
ds
```

Out[75]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	NaN	23	NaN

In [76]:
```
ds.drop(6,inplace=True)
```

In [77]:
```
ds
```

Out[77]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE

In [78]:
```
ds.isnull()
```

Out[78]:

	Name	Age	Dept
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False

In [79]:
```
ds.loc[6]=['Nikki','NaN','EEE']
```

In [80]:
```
ds
```

Out[80]:

	Name	Age	Dept
0	Jas	19	CSE
1	Loga	19	CSE
2	Sai	20	CSE
3	Chethan	20	CSE
4	Palla	19	CSE
5	lokesh	21	EEE
6	Nikki	NaN	EEE

In [81]:
```
ds.isnull()
```

Out[81]:

	Name	Age	Dept
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False

In [84]:
```
ds.plot.bar()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14508\2181631866.py in <module>
----> 1 ds.plot.bar()

~\anaconda3\lib\site-packages\pandas\plotting\_core.py in bar(self, x, y, **kwargs)
   1129         other axis represents a measured value.
   1130         """
-> 1131         return self(kind="bar", x=x, y=y, **kwargs)
   1132
   1133     @Appender(

~\anaconda3\lib\site-packages\pandas\plotting\_core.py in __call__(self, *args, **kwargs)
    970                     data.columns = label_name
    971
--> 972         return plot_backend.plot(data, kind=kind, **kwargs)
    973
    974     __call__.__doc__ = __doc__

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\__init__.py in plot(data, kind, **kwargs)
     69             kwargs["ax"] = getattr(ax, "left_ax", ax)
     70     plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71     plot_obj.generate()
     72     plot_obj.draw()
     73     return plot_obj.result

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py in generate(self)
    325     def generate(self):
    326         self._args_adjust()
--> 327         self._compute_plot_data()
    328         self._setup_subplots()
    329         self._make_plot()

~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py in _compute_plot_data(self)
    504         # no non-numeric frames or series allowed
    505         if is_empty:
--> 506             raise TypeError("no numeric data to plot")
    507
    508         self.data = numeric_data.apply(self._convert_to_ndarray)

TypeError: no numeric data to plot
```

In [ ]: