In [2]:
```python
print('hello world i am here again')
```
```
hello world i am here again
```

In [3]:
```python
import pandas as pd
import matplotlib.pyplot as plt
```

In [5]:
```python
df=pd.read_csv(r'C:\Users\DELL\Downloads\Call-Center-Dataset.csv')
df.head()
```

Out[5]:

	Call Id	Agent	Date	Time	Topic	Answered (Y/N)	Resolved	Speed of answer in seconds	AvgTalkDuration	Satisfaction rating	Unnamed: 10	Unnamed: 11	Unnamed: 12	Unnamed: 13
0	ID0001	Diane	2021-01-01	9:12:58	Contract related	Y	Y	109.0	0:02:23	3.0	NaN	NaN	NaN	NaN
1	ID0002	Becky	2021-01-01	9:12:58	Technical Support	Y	N	70.0	0:04:02	3.0	NaN	NaN	NaN	NaN
2	ID0003	Stewart	2021-01-01	9:47:31	Contract related	Y	Y	10.0	0:02:11	3.0	NaN	NaN	NaN	NaN
3	ID0004	Greg	2021-01-01	9:47:31	Contract related	Y	Y	53.0	0:00:37	2.0	NaN	NaN	NaN	NaN
4	ID0005	Becky	2021-01-01	10:00:29	Payment related	Y	Y	95.0	0:01:00	3.0	NaN	NaN	NaN	NaN

In [6]:
```python
#subset data
df1=df[['Call Id','Agent','Date','Time','Topic','Answered (Y/N)','Resolved','Speed of answer in seconds','AvgTalkDuration','Satisfaction rating']]
```

In [7]:
```python
#time to check data type
df1.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 10 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Call Id                     5000 non-null   object
 1   Agent                       5000 non-null   object
 2   Date                        5000 non-null   object
 3   Time                        5000 non-null   object
 4   Topic                       5000 non-null   object
 5   Answered (Y/N)              5000 non-null   object
 6   Resolved                    5000 non-null   object
 7   Speed of answer in seconds  4054 non-null   float64
 8   AvgTalkDuration             4054 non-null   object
 9   Satisfaction rating         4054 non-null   float64
dtypes: float64(2), object(8)
memory usage: 390.8+ KB
```

In [8]:
```python
df1['Date']=df1['Date'].astype('datetime64[ns]')
```
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_12860\150799835.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1['Date']=df1['Date'].astype('datetime64[ns]')
```

In [9]:
```python
df1.describe()
```

Out[9]:

	Date	Speed of answer in seconds	Satisfaction rating
count	5000	4054.000000	4054.000000
mean	2021-02-13 05:32:38.400000256	67.520720	3.403552
min	2021-01-01 00:00:00	10.000000	1.000000
25%	2021-01-23 00:00:00	39.000000	3.000000
50%	2021-02-13 00:00:00	68.000000	3.000000
75%	2021-03-06 00:00:00	97.000000	4.000000
max	2021-03-31 00:00:00	125.000000	5.000000
std	NaN	33.592872	1.212220

In [10]:
```python
#looking for null values
df1.isnull().any().sum()
```
Out[10]: 3

In [12]:
```python
#i have a ghost call in call ID5000 ,i will drop it
df1.dropna().tail()
```

Out[12]:

	Call Id	Agent	Date	Time	Topic	Answered (Y/N)	Resolved	Speed of answer in seconds	AvgTalkDuration	Satisfaction rating
4990	ID4991	Stewart	2021-03-31	13:46:34	Streaming	Y	Y	73.0	0:06:26	1.0
4995	ID4996	Jim	2021-03-31	16:37:55	Payment related	Y	Y	22.0	0:05:40	1.0
4996	ID4997	Diane	2021-03-31	16:45:07	Payment related	Y	Y	100.0	0:03:16	3.0
4997	ID4998	Diane	2021-03-31	16:53:46	Payment related	Y	Y	84.0	0:01:49	4.0
4998	ID4999	Jim	2021-03-31	17:02:24	Streaming	Y	Y	98.0	0:00:58	5.0

In [13]:
```python
# I want to add a new column as Speed of answer in munites
df1['Speed of answer in munites']=df['Speed of answer in seconds']/60
```
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_12860\2123537462.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1['Speed of answer in munites']=df['Speed of answer in seconds']/60
```

In [14]:
```python
df1=df[['Call Id','Agent','Date','Time','Topic','Answered (Y/N)','Resolved','Speed of answer in seconds','Speed of answer in munites','AvgTalkDuration','Satisfaction rating']]
```
```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[14], line 1
----> 1 df1=df[['Call Id','Agent','Date','Time','Topic','Answered (Y/N)','Resolved','Speed of answer in seconds','Speed of answer in munites','AvgTalkDuration','Satisfaction rating']]

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:3899, in DataFrame.__getitem__(self, key)
   3897     if is_iterator(key):
   3898         key = list(key)
-> 3899     indexer = self.columns._get_indexer_strict(key, "columns")[1]
   3901 # take() does not accept boolean indexers
   3902 if getattr(indexer, "dtype", None) == bool:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6115, in Index._get_indexer_strict(self, key, axis_name)
   6112 else:
   6113     keyarr, indexer, new_indexer = self._reindex_non_unique(keyarr)
-> 6115 self._raise_if_missing(keyarr, indexer, axis_name)
   6117 keyarr = self.take(indexer)
   6118 if isinstance(key, Index):
   6119     # GH 42790 - Preserve name from an Index

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6179, in Index._raise_if_missing(self, key, indexer, axis_name)
   6176     raise KeyError(f"None of [{key}] are in the [{axis_name}]")
   6178 not_found = list(ensure_index(key)[missing_mask.nonzero()[0]].unique())
-> 6179 raise KeyError(f"{not_found} not in index")

KeyError: "['Speed of answer in munites'] not in index"
```

In [15]:
```python
df1
```

Out[15]:

	Call Id	Agent	Date	Time	Topic	Answered (Y/N)	Resolved	Speed of answer in seconds	AvgTalkDuration	Satisfaction rating	Speed of answer in munites
0	ID0001	Diane	2021-01-01	9:12:58	Contract related	Y	Y	109.0	0:02:23	3.0	1.816667
1	ID0002	Becky	2021-01-01	9:12:58	Technical Support	Y	N	70.0	0:04:02	3.0	1.166667
2	ID0003	Stewart	2021-01-01	9:47:31	Contract related	Y	Y	10.0	0:02:11	3.0	0.166667
3	ID0004	Greg	2021-01-01	9:47:31	Contract related	Y	Y	53.0	0:00:37	2.0	0.883333
4	ID0005	Becky	2021-01-01	10:00:29	Payment related	Y	Y	95.0	0:01:00	3.0	1.583333
...	...	...	...	...	...	...	...	...	...	...	...
4995	ID4996	Jim	2021-03-31	16:37:55	Payment related	Y	Y	22.0	0:05:40	1.0	0.366667
4996	ID4997	Diane	2021-03-31	16:45:07	Payment related	Y	Y	100.0	0:03:16	3.0	1.666667
4997	ID4998	Diane	2021-03-31	16:53:46	Payment related	Y	Y	84.0	0:01:49	4.0	1.400000
4998	ID4999	Jim	2021-03-31	17:02:24	Streaming	Y	Y	98.0	0:00:58	5.0	1.633333
4999	ID5000	Diane	2021-03-31	17:39:50	Contract related	N	N	NaN	NaN	NaN	NaN

5000 rows × 11 columns

In [16]:
```python
df1.dropna().tail()
```

Out[16]:

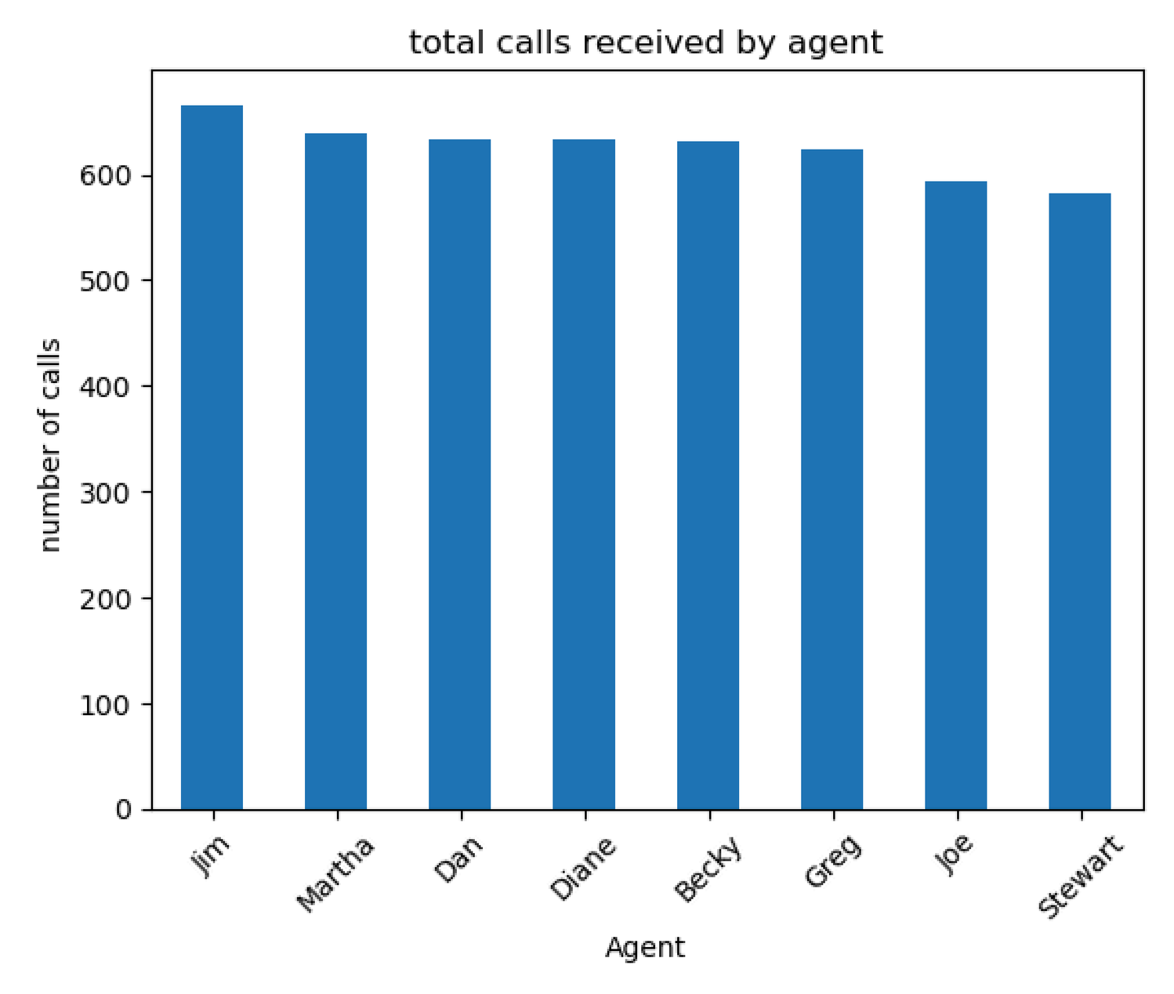
	Call Id	Agent	Date	Time	Topic	Answered (Y/N)	Resolved	Speed of answer in seconds	AvgTalkDuration	Satisfaction rating	Speed of answer in munites
4990	ID4991	Stewart	2021-03-31	13:46:34	Streaming	Y	Y	73.0	0:06:26	1.0	1.216667
4995	ID4996	Jim	2021-03-31	16:37:55	Payment related	Y	Y	22.0	0:05:40	1.0	0.366667
4996	ID4997	Diane	2021-03-31	16:45:07	Payment related	Y	Y	100.0	0:03:16	3.0	1.666667
4997	ID4998	Diane	2021-03-31	16:53:46	Payment related	Y	Y	84.0	0:01:49	4.0	1.400000
4998	ID4999	Jim	2021-03-31	17:02:24	Streaming	Y	Y	98.0	0:00:58	5.0	1.633333

In [17]:
```python
# number of agent
df1.nunique()
```
Out[17]:
```
Call Id                       5000
Agent                            8
Date                            90
Time                           375
Topic                            5
Answered (Y/N)                   2
Resolved                         2
Speed of answer in seconds     116
AvgTalkDuration                391
Satisfaction rating              5
Speed of answer in munites     116
dtype: int64
```

In [18]:
```python
#total calls received by agent
df1_group=df1.groupby('Agent') ['Call Id'].count().sort_values(ascending=False)
df1_group
```
Out[18]:
```
Agent
Jim        666
Martha     638
Dan        633
Diane      633
Becky      631
Greg       624
Joe        593
Stewart    582
Name: Call Id, dtype: int64
```

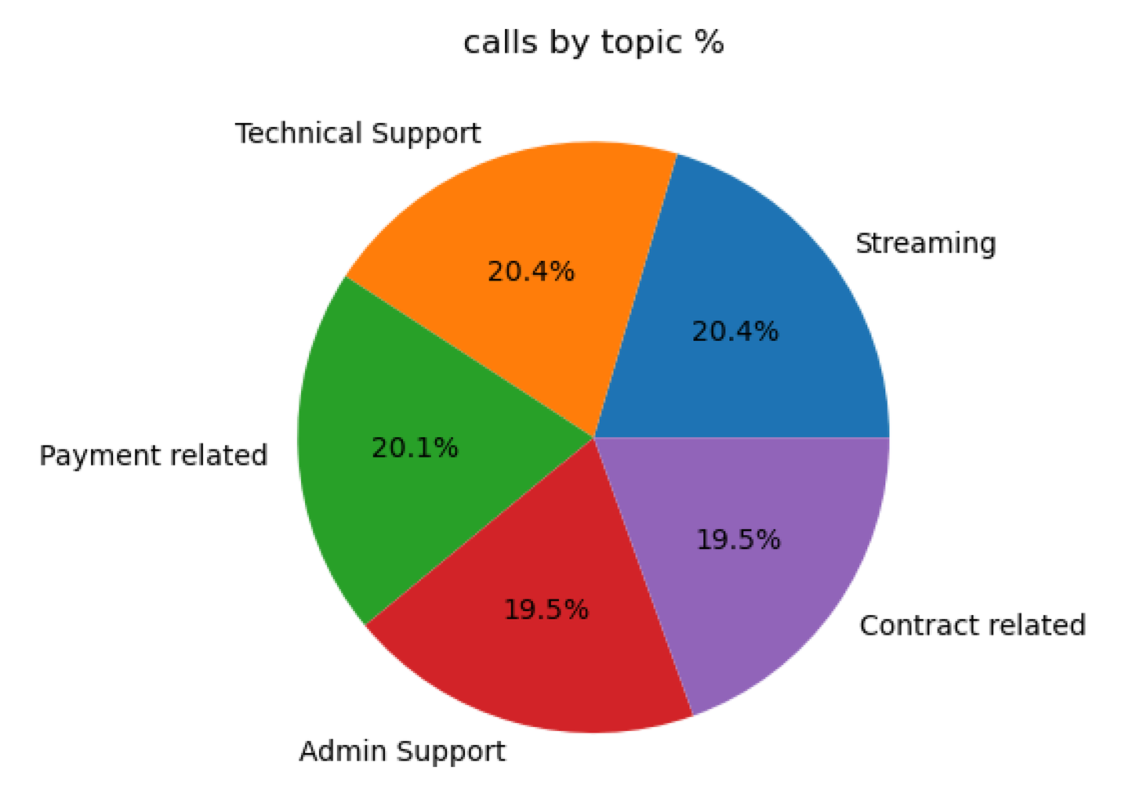
In [19]:
```python
# now we create bar chart

df1_group.plot(kind='bar',rot=45)
plt.title('total calls received by agent')
plt.ylabel('number of calls ')
plt.show()
```



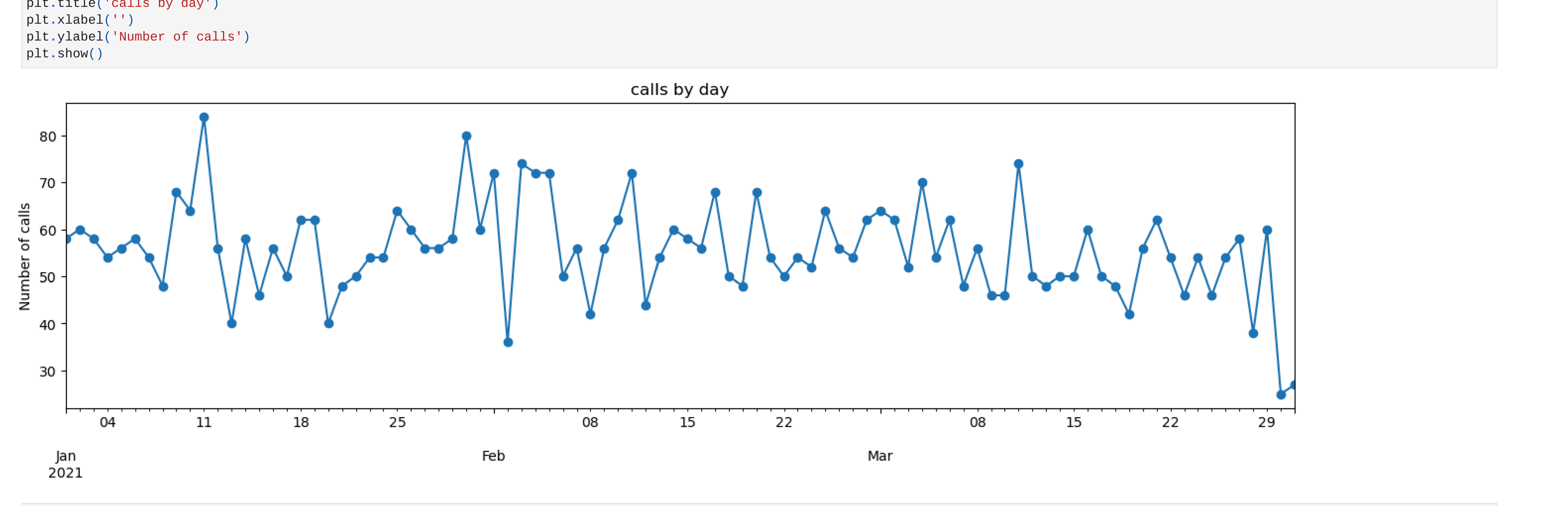
In [20]:
```python
#trendy topic in this time cycle
df1_topic_group= df1.groupby('Topic') ['Call Id'].count().sort_values(ascending=False)
df1_topic_group
```
Out[20]:
```
Topic
Streaming            1022
Technical Support    1019
Payment related      1007
Admin Support         976
Contract related      976
Name: Call Id, dtype: int64
```

In [21]:
```python
#now we create pie chart
df1_topic_group.plot(kind='pie',autopct='%1.1f%%')
plt.title('calls by topic %')
plt.ylabel('')
```
Out[21]: Text(0, 0.5, '')



In [22]:
```python
# Let's creat some trend over time
df_date_group=df1.groupby('Date') ['Call Id'].count()
df_date_group
```
Out[22]:
```
Date
2021-01-01    58
2021-01-02    60
2021-01-03    58
2021-01-04    54
2021-01-05    56
              ..
2021-03-27    58
2021-03-28    38
2021-03-29    60
2021-03-30    25
2021-03-31    27
Name: Call Id, Length: 90, dtype: int64
```

In [23]:
```python
#line chart
plt.figure(figsize=(16, 4))
df_date_group.plot(kind='line',marker='o')
plt.title('calls by day')
plt.xlabel('')
plt.ylabel('Number of calls')
plt.show()
```



In [24]:
```python
#top agent by Satisfaction rating Csat
df_Csat=df1.groupby('Agent') ['Satisfaction rating'].mean().sort_values(ascending=False).round(1)
df_Csat
```
Out[24]:
```
Agent
Martha     3.5
Dan        3.4
Diane      3.4
Greg       3.4
Stewart    3.4
Jim        3.4
Becky      3.4
Joe        3.3
Name: Satisfaction rating, dtype: float64
```

In [25]:
```python
#let's creat hist chart
df_Csat.plot(kind='barh')
plt.title('Avg Csat by agent')
plt.xlabel('Csat by 5 starts')
plt.show()
```