In [64]:

```
%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

DATASET DESCRIPTION

```
    - instant: record index
    - dteday : date
    - season : season (1:springer, 2:summer, 3:fall, 4:winter)
    - yr : year (0: 2011, 1:2012)
    - mnth : month ( 1 to 12)
    - hr : hour (0 to 23)
    - holiday : weather day is holiday or not (extracted from http://dchr.dc.
    gov/page/holiday-schedule)
    - weekday : day of the week
    - workingday : if day is neither weekend nor holiday is 1, otherwise is 0
    .
    + weathersit :
        - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
        - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
        - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light
    Rain + Scattered clouds
        - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
    - temp : Normalized temperature in Celsius. The values are divided to 41
    (max)
    - atemp: Normalized feeling temperature in Celsius. The values are divide
    d to 50 (max)
    - hum: Normalized humidity. The values are divided to 100 (max)
    - windspeed: Normalized wind speed. The values are divided to 67 (max)
    - cnt: count of total rental bikes including both casual and registered
```

In [65]:

```
df = pd.read_csv("day.csv")
df=df.drop(['casual', 'registered'], axis=1)
#convert date to numerical value
#df["date"]=df['dteday'].str.replace("-","").astype(int)
df.head()
```

Out[65]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0

Exploration

In [66]:

```
df.describe()
```

Out[66]:

	instant	season	yr	mnth	holiday	weekday	workingday
count	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000
mean	366.000000	2.496580	0.500684	6.519836	0.028728	2.997264	0.683995
std	211.165812	1.110807	0.500342	3.451913	0.167155	2.004787	0.465233
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	183.500000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000
50%	366.000000	3.000000	1.000000	7.000000	0.000000	3.000000	1.000000
75%	548.500000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000
max	731.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000

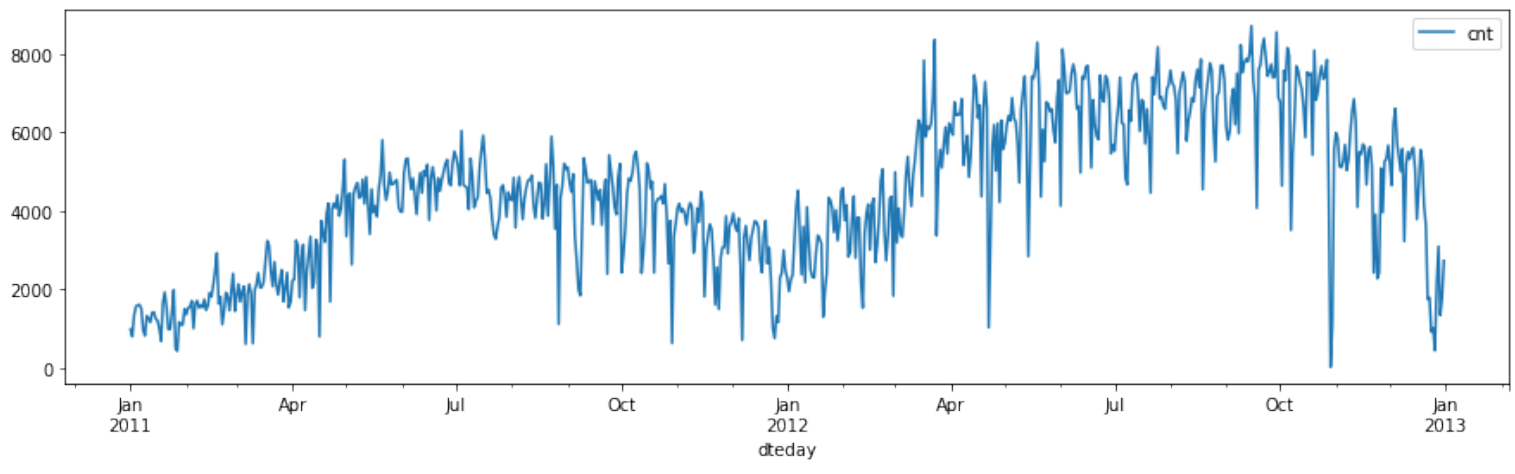
In [67]:

```
#distribution of number of rentals a day during the whole period

df['dteday'] = pd.to_datetime(df['dteday'],infer_datetime_format=True)
df.plot(kind='line',x='dteday',y='cnt',figsize=(15,4))
```

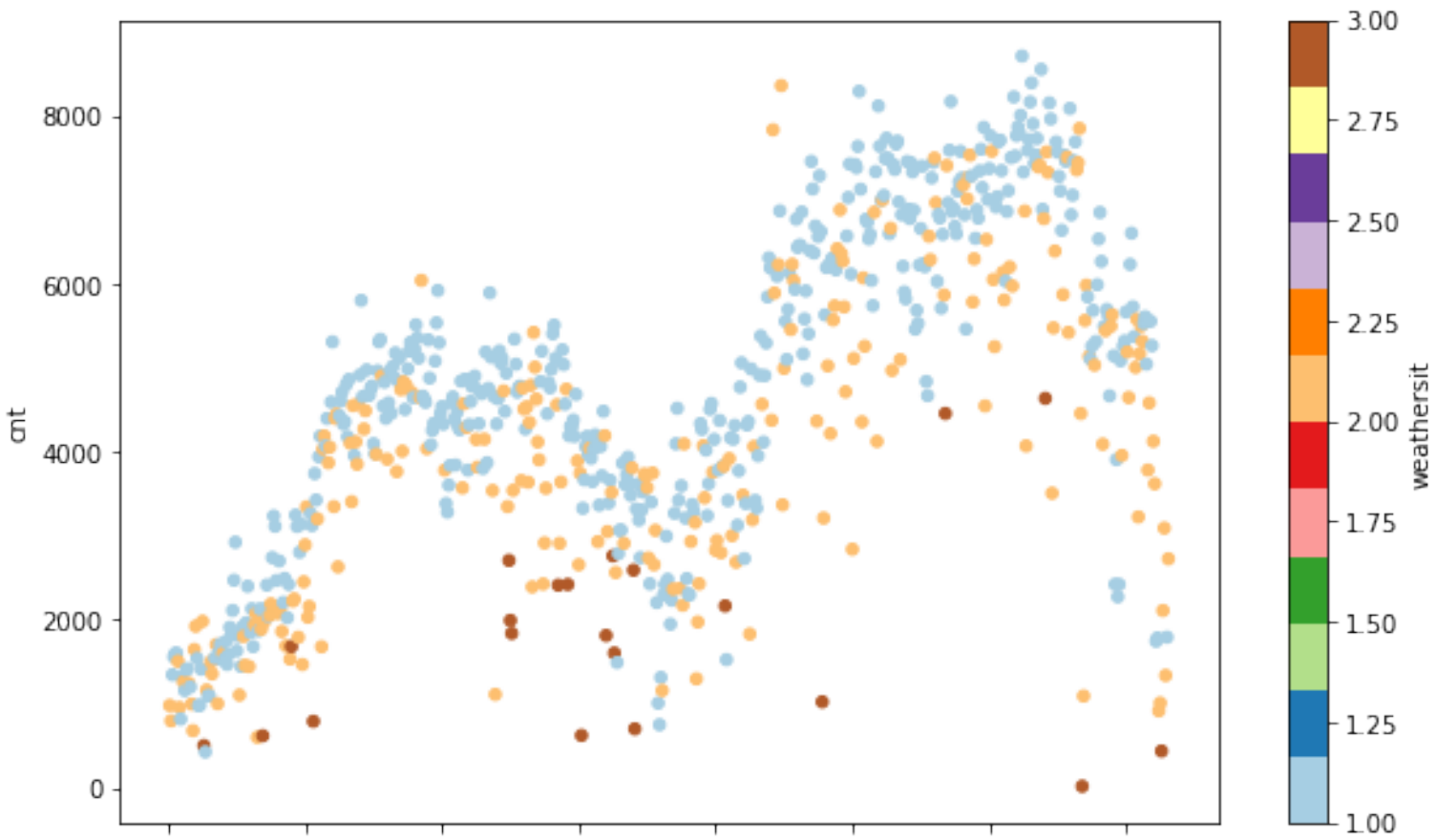
Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x124b17438>
```



In [68]:

```
#distribution of number of rentals a day during the whole period, colored by weat
df.plot(kind='scatter',x="instant",y='cnt',c='weathersit',colormap= "Paired",figs
plt.show()
```

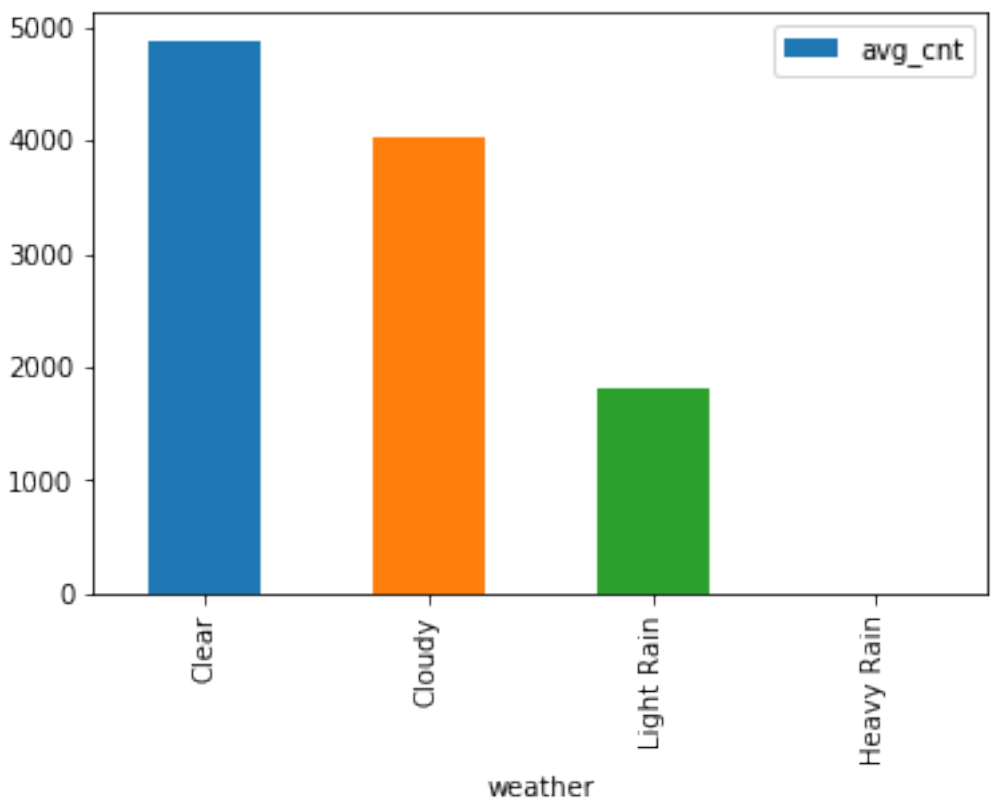


In [69]:

```
group1_df=df.loc[df['weathersit'] == 1]
print("Number of day with weather= Clear -->",group1_df.shape[0])
group2_df=df.loc[df['weathersit'] == 2]
print("Number of day with weather= Cloudy -->",group2_df.shape[0])
group3_df=df.loc[df['weathersit'] == 3]
print("Number of day with weather= Light Rain -->",group3_df.shape[0])
group4_df=df.loc[df['weathersit'] == 4]
print("Number of day with weather= Heavy Rain  -->",group4_df.shape[0])

#Avg number of rentals grouped by weather
d = {'weather': ["Clear", "Cloudy","Light Rain","Heavy Rain"], 'avg_cnt': [group1
df2 = pd.DataFrame(data=d)
df2.plot(kind='bar',x="weather",y='avg_cnt')
plt.show()
```

```
Number of day with weather= Clear --> 463
Number of day with weather= Cloudy --> 247
Number of day with weather= Light Rain --> 21
Number of day with weather= Heavy Rain  --> 0
```



In [ ]:

In [64]: