In [2]:
```
#IMPORTING REQUIRED LIBRARY

import os
import pandas as pd
import numpy as np
import matplotlib as plt
import datetime as dt
import seaborn as sns
```

In [3]:
```
import import_ipynb
import matplotlib.pyplot as plt1
```

In [4]:
```
%matplotlib inline
import sklearn
```

In [5]:
```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
```

In [6]:
```
from sklearn.ensemble import RandomForestRegressor
```

In [7]:
```
import statsmodels.api as sm
```

In [472]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neighbors import KNeighborsRegressor
```

In [10]:
```
#setting working directory
os.chdir("E:/data science and machine learning/BIKE RENTAL PREDICTIONS/Python"
)
```

In [11]:
```
os.getcwd()
```

Out[11]:
```
'E:\\data science and machine learning\\BIKE RENTAL PREDICTIONS\\Python'
```

In [12]:
```
#GETTING THE FILE FROM HDD
bdf=pd.read_csv("day.csv",sep=',')
```

In [13]:
```
bdf.shape
```

Out[13]:
```
(731, 16)
```

In [14]:
```
type(bdf)
```

Out[14]:
```
pandas.core.frame.DataFrame
```

In [16]:
```
bdf.columns
```

Out[16]:
```
Index(['instant', 'dteday', 'season', 'yr', 'mnth', 'holiday', 'weekday',
       'workingday', 'weathersit', 'temp', 'atemp', 'hum', 'windspeed',
       'casual', 'registered', 'cnt'],
      dtype='object')
```

In [146]:
```
bdf.dtypes
bdf
```

Out[146]:

	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.
5	6	2011-01-06	1	0	1	0	4	1	1	0.204348	0.
6	7	2011-01-07	1	0	1	0	5	1	2	0.196522	0.
7	8	2011-01-08	1	0	1	0	6	0	2	0.165000	0.
8	9	2011-01-09	1	0	1	0	0	0	1	0.138333	0.
9	10	2011-01-10	1	0	1	0	1	1	1	0.150833	0.
10	11	2011-01-11	1	0	1	0	2	1	2	0.169091	0.
11	12	2011-01-12	1	0	1	0	3	1	1	0.172727	0.
12	13	2011-01-13	1	0	1	0	4	1	1	0.165000	0.
13	14	2011-01-14	1	0	1	0	5	1	1	0.160870	0.
14	15	2011-01-15	1	0	1	0	6	0	2	0.233333	0.
15	16	2011-01-16	1	0	1	0	0	0	1	0.231667	0.
16	17	2011-01-17	1	0	1	1	1	0	2	0.175833	0.
17	18	2011-01-18	1	0	1	0	2	1	2	0.216667	0.
18	19	2011-01-19	1	0	1	0	3	1	2	0.292174	0.
19	20	2011-01-20	1	0	1	0	4	1	2	0.261667	0.
20	21	2011-01-21	1	0	1	0	5	1	1	0.177500	0.
21	22	2011-01-22	1	0	1	0	6	0	1	0.059130	0.
22	23	2011-01-23	1	0	1	0	0	0	1	0.096522	0.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	
23	24	2011-01-24	1	0	1	0	1	1	1	0.097391	0.
24	25	2011-01-25	1	0	1	0	2	1	2	0.223478	0.
25	26	2011-01-26	1	0	1	0	3	1	3	0.217500	0.
26	27	2011-01-27	1	0	1	0	4	1	1	0.195000	0.
27	28	2011-01-28	1	0	1	0	5	1	2	0.203478	0.
28	29	2011-01-29	1	0	1	0	6	0	1	0.196522	0.
29	30	2011-01-30	1	0	1	0	0	0	1	0.216522	0.
...	...	...	...	...	...	...	...	...	...	...	
701	702	2012-12-02	4	1	12	0	0	0	2	0.347500	0.
702	703	2012-12-03	4	1	12	0	1	1	1	0.452500	0.
703	704	2012-12-04	4	1	12	0	2	1	1	0.475833	0.
704	705	2012-12-05	4	1	12	0	3	1	1	0.438333	0.
705	706	2012-12-06	4	1	12	0	4	1	1	0.255833	0.
706	707	2012-12-07	4	1	12	0	5	1	2	0.320833	0.
707	708	2012-12-08	4	1	12	0	6	0	2	0.381667	0.
708	709	2012-12-09	4	1	12	0	0	0	2	0.384167	0.
709	710	2012-12-10	4	1	12	0	1	1	2	0.435833	0.
710	711	2012-12-11	4	1	12	0	2	1	2	0.353333	0.
711	712	2012-12-12	4	1	12	0	3	1	2	0.297500	0.
712	713	2012-12-13	4	1	12	0	4	1	1	0.295833	0.
713	714	2012-12-14	4	1	12	0	5	1	1	0.281667	0.
714	715	2012-12-15	4	1	12	0	6	0	1	0.324167	0.
715	716	2012-12-16	4	1	12	0	0	0	2	0.362500	0.
716	717	2012-12-17	4	1	12	0	1	1	2	0.393333	0.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	
717	718	2012-12-18	4	1	12	0	2	1	1	0.410833	0.
718	719	2012-12-19	4	1	12	0	3	1	1	0.332500	0.
719	720	2012-12-20	4	1	12	0	4	1	2	0.330000	0.
720	721	2012-12-21	1	1	12	0	5	1	2	0.326667	0.
721	722	2012-12-22	1	1	12	0	6	0	1	0.265833	0.
722	723	2012-12-23	1	1	12	0	0	0	1	0.245833	0.
723	724	2012-12-24	1	1	12	0	1	1	2	0.231304	0.
724	725	2012-12-25	1	1	12	1	2	0	2	0.291304	0.
725	726	2012-12-26	1	1	12	0	3	1	3	0.243333	0.
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.

731 rows × 18 columns

In [28]:
```
#extracting day from datetime
from datetime import date
bdf['date']=pd.to_datetime(bdf['dteday'],errors='coerce')
bdf['day']=bdf['date'].apply(lambda x:x.day)
```

In [173]:
```
bdf1=bdf
```

In [174]:
```
#MISSING VALUE ANALYSIS
missing_val=pd.DataFrame(bdf.isnull().sum())
```

In [175]:
```
missing_val
```

Out[175]:

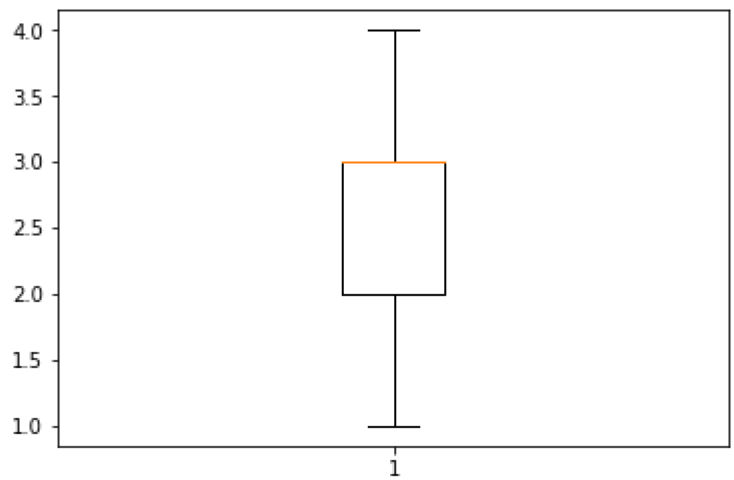
	0
instant	0
dteday	0
season	0
yr	0
mnth	0
holiday	0
weekday	0
workingday	0
weathersit	0
temp	0
atemp	0
hum	0
windspeed	0
casual	0
registered	0
cnt	0
date	0
day	0

In [176]:
```
cnames=['season','yr','mnth','holiday','weekday','workingday','weathersit','te
mp','atemp','hum','windspeed','casual','registered','cnt','day']
for i in cnames:
    q75,q25=np.percentile(bdf1.loc[:,i],[75,25])
    iqr=q75-q25
    min=q25-(iqr*1.5)
    max=q75+(iqr*1.5)

   # bdf1=bdf1.drop(bdf1[bdf1.loc[:,i]<min].index)
   # bdf1=bdf1.drop(bdf1[bdf1.loc[:,i]>max].index)
```

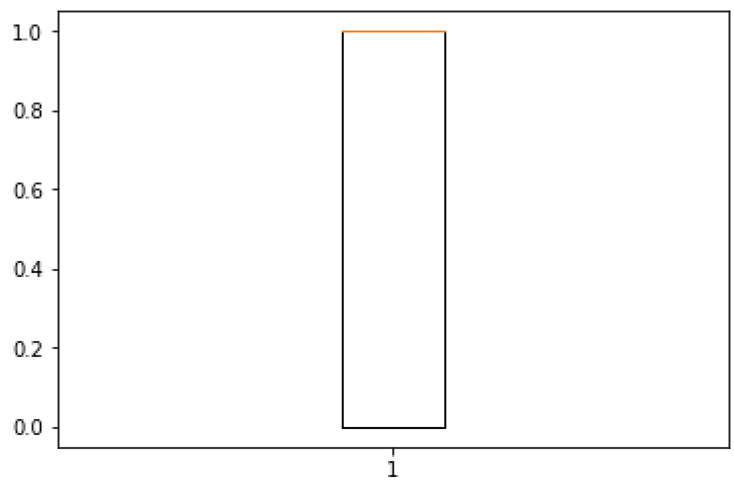
In [177]:
```
plt1.boxplot(bdf1['season'])
```

Out[177]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fc5014a8>,
  <matplotlib.lines.Line2D at 0x1e4fc5017f0>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fc501b38>,
  <matplotlib.lines.Line2D at 0x1e4fc501e80>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fc5010b8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fc501f60>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fc52e550>],
 'means': []}
```



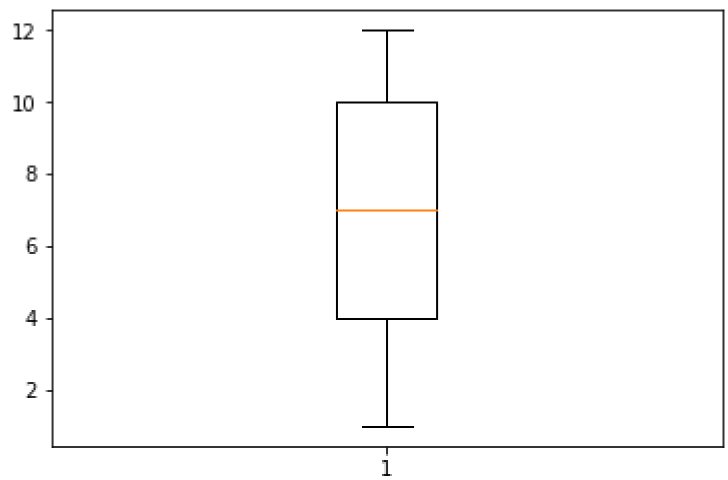
In [178]:
```
plt1.boxplot(bdf1['yr'])
```

Out[178]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fda3fe80>,
  <matplotlib.lines.Line2D at 0x1e4fda484a8>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fda487f0>,
  <matplotlib.lines.Line2D at 0x1e4fda48b38>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fda3fd30>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fda48e80>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fda48f60>],
 'means': []}
```



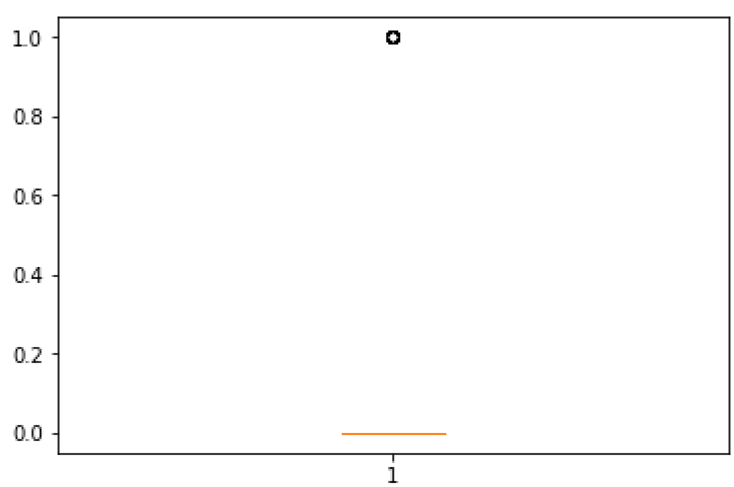
In [179]:
```
plt1.boxplot(bdf1['mnth'])
```

Out[179]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fda9a8d0>,
  <matplotlib.lines.Line2D at 0x1e4fda9ac18>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fda9af60>,
  <matplotlib.lines.Line2D at 0x1e4fdaa52e8>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fda9a4e0>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdaa5630>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdaa5978>],
 'means': []}
```



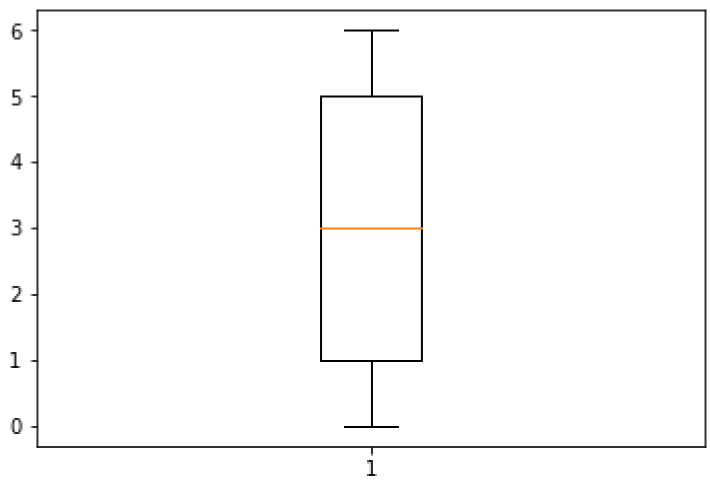
In [180]:
```
plt1.boxplot(bdf1['holiday'])
```

Out[180]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdaebf98>,
  <matplotlib.lines.Line2D at 0x1e4fdaf6320>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdaf6668>,
  <matplotlib.lines.Line2D at 0x1e4fdaf69b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdaebba8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdaf6cf8>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdaf6dd8>],
 'means': []}
```



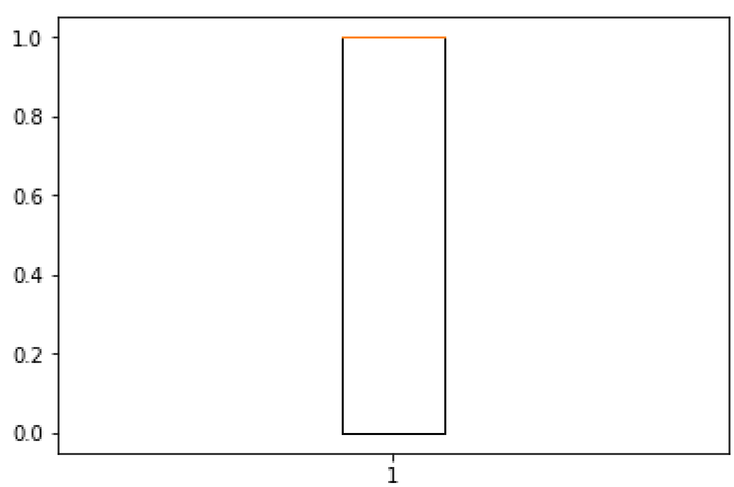
In [181]:
```
plt1.boxplot(bdf1['weekday'])
```

Out[181]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdb486d8>,
  <matplotlib.lines.Line2D at 0x1e4fdb48a20>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdb48d68>,
  <matplotlib.lines.Line2D at 0x1e4fdb48e48>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdb482e8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdb52438>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdb52780>],
 'means': []}
```



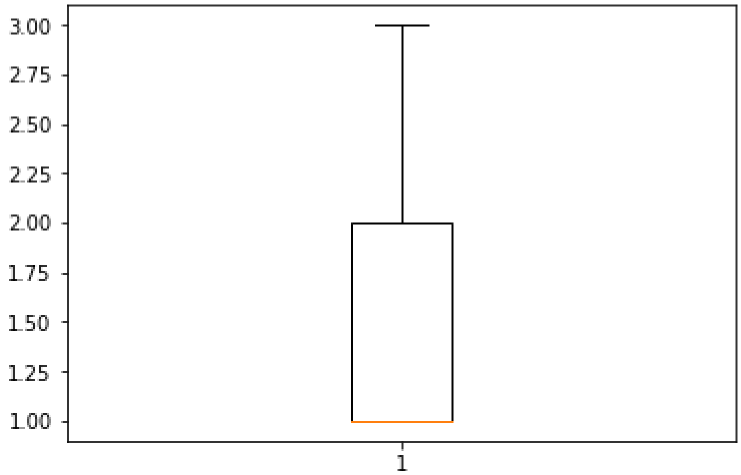
In [187]:
```
plt1.boxplot(bdf1['workingday'])
```

Out[187]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdd63e10>,
  <matplotlib.lines.Line2D at 0x1e4fdd6e438>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdd6e780>,
  <matplotlib.lines.Line2D at 0x1e4fdd6eac8>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdd63cc0>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdd6ee10>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdd6eef0>],
 'means': []}
```



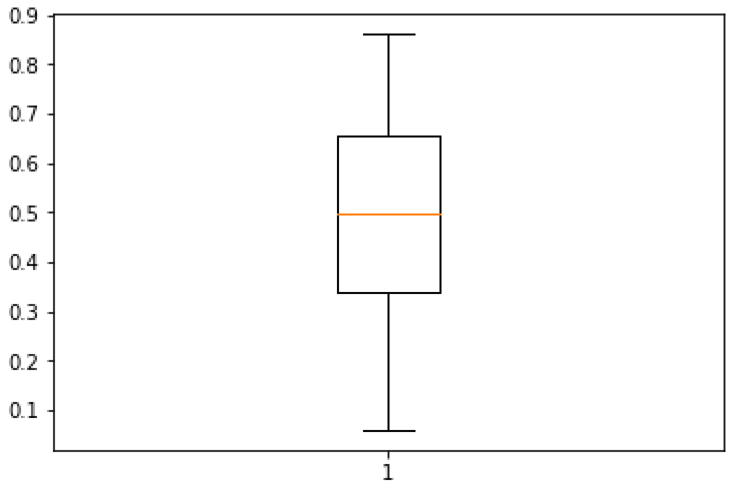
In [188]:
```
plt1.boxplot(bdf1['weathersit'])
```

Out[188]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fddbe940>,
  <matplotlib.lines.Line2D at 0x1e4fddbec88>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fddbefd0>,
  <matplotlib.lines.Line2D at 0x1e4fddc8358>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fddbe550>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fddc86a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fddc89e8>],
 'means': []}
```



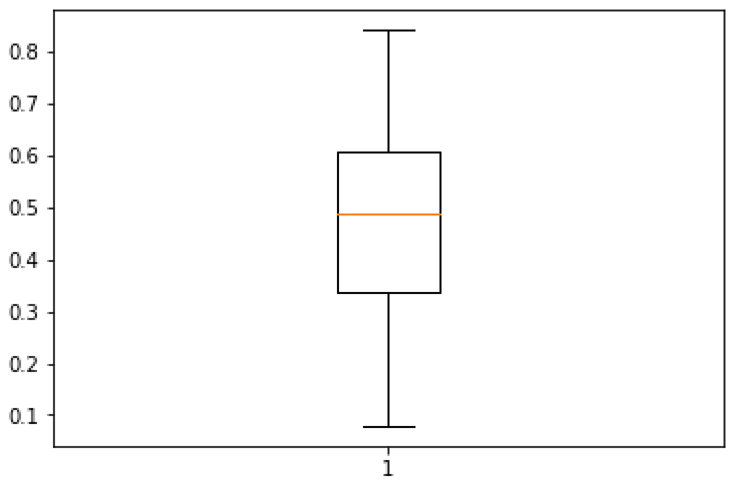
In [189]:
```
plt1.boxplot(bdf1['temp'])
```

Out[189]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fde18f98>,
  <matplotlib.lines.Line2D at 0x1e4fde24320>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fde24668>,
  <matplotlib.lines.Line2D at 0x1e4fde249b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fde18ba8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fde24cf8>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fde24dd8>],
 'means': []}
```



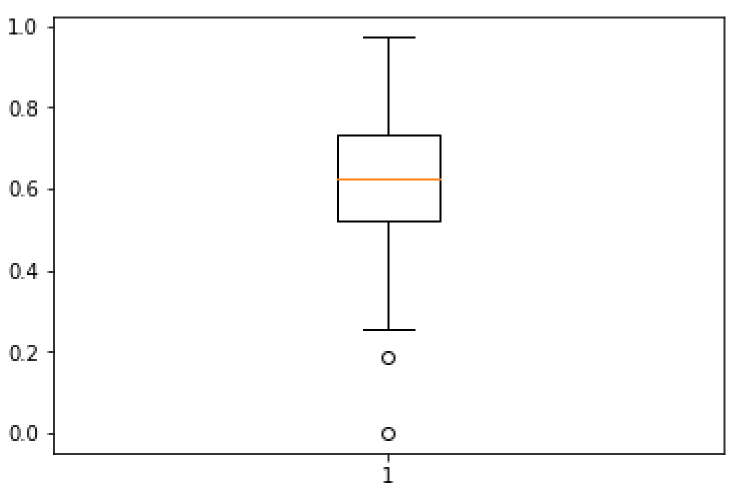
In [190]:
```
plt1.boxplot(bdf1['atemp'])
```

Out[190]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fde805f8>,
  <matplotlib.lines.Line2D at 0x1e4fde80940>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fde80c88>,
  <matplotlib.lines.Line2D at 0x1e4fde80fd0>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fde80208>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fde86358>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fde866a0>],
 'means': []}
```



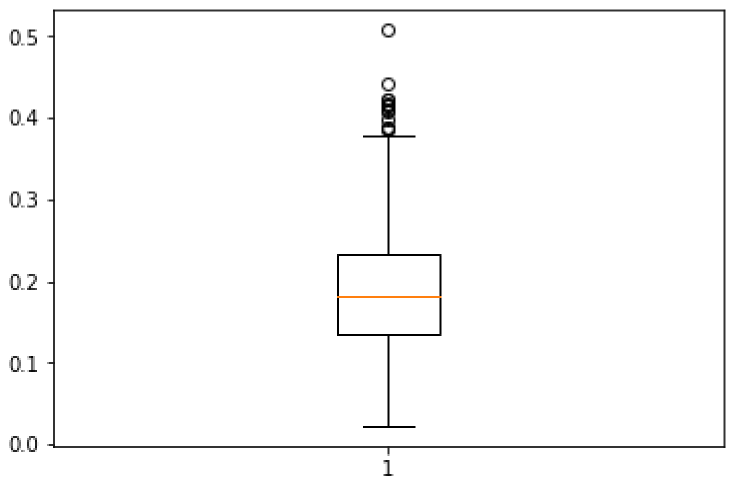
In [191]:
```
plt1.boxplot(bdf1['hum'])
```

Out[191]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fded8748>,
  <matplotlib.lines.Line2D at 0x1e4fded8a90>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fded8dd8>,
  <matplotlib.lines.Line2D at 0x1e4fded8eb8>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fded8358>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdee34a8>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdee37f0>],
 'means': []}
```



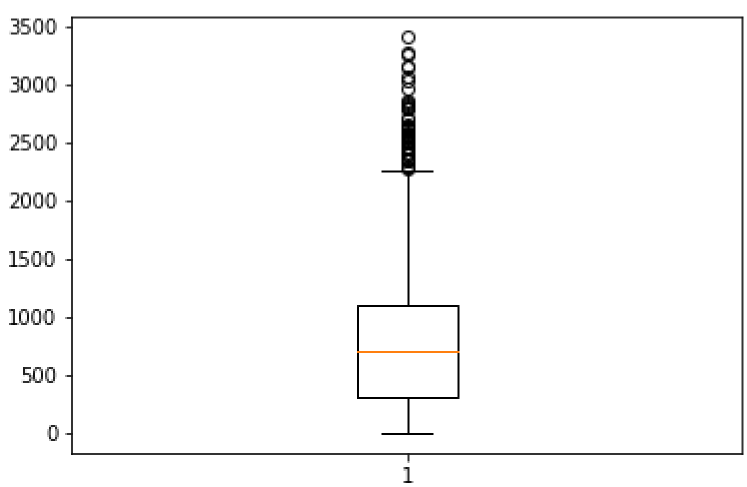
In [192]:
```
plt1.boxplot(bdf1['windspeed'])
```

Out[192]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdf2bf28>,
  <matplotlib.lines.Line2D at 0x1e4fdf342b0>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdf345f8>,
  <matplotlib.lines.Line2D at 0x1e4fdf34940>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdf2bb38>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdf34c88>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdf34fd0>],
 'means': []}
```



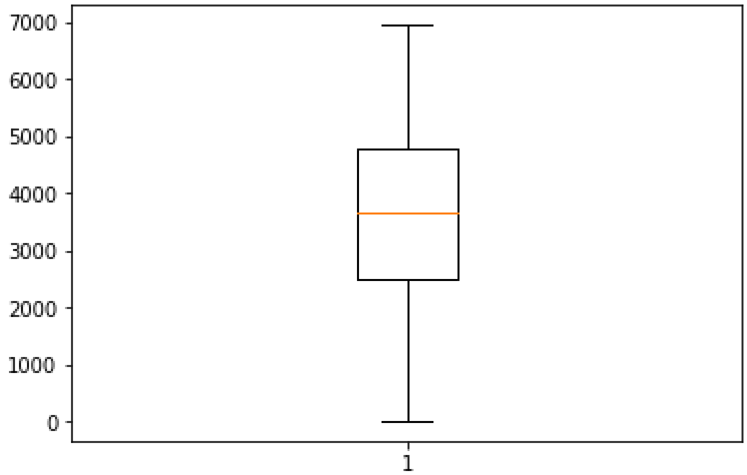
In [193]:
```
plt1.boxplot(bdf1['casual'])
```

Out[193]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdf866d8>,
  <matplotlib.lines.Line2D at 0x1e4fdf86a20>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdf86d68>,
  <matplotlib.lines.Line2D at 0x1e4fdf86e48>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdf862e8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdf92438>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdf92780>],
 'means': []}
```



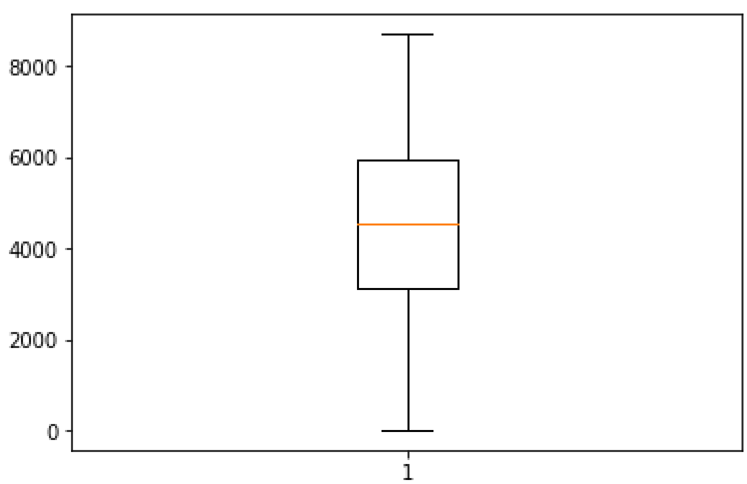
In [194]:
```
plt1.boxplot(bdf1['registered'])
```

Out[194]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fdfe27b8>,
  <matplotlib.lines.Line2D at 0x1e4fdfe2b00>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fdfe2e48>,
  <matplotlib.lines.Line2D at 0x1e4fdfe2f28>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fdfe23c8>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fdfee518>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fdfee860>],
 'means': []}
```



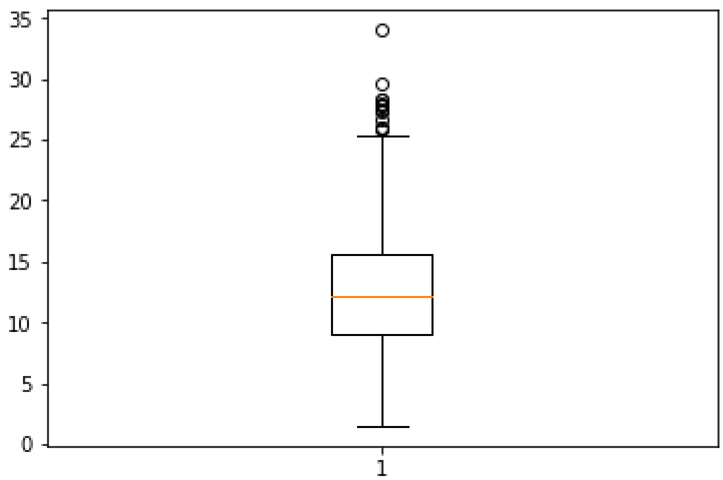
In [195]:
```
plt1.boxplot(bdf1['cnt'])
```

Out[195]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e4fe03a940>,
  <matplotlib.lines.Line2D at 0x1e4fe03ac88>],
 'caps': [<matplotlib.lines.Line2D at 0x1e4fe03afd0>,
  <matplotlib.lines.Line2D at 0x1e4fe044358>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e4fe03a550>],
 'medians': [<matplotlib.lines.Line2D at 0x1e4fe0446a0>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e4fe0449e8>],
 'means': []}
```



In [488]:
```
plt1.boxplot(bdf1['windspeed'])
```

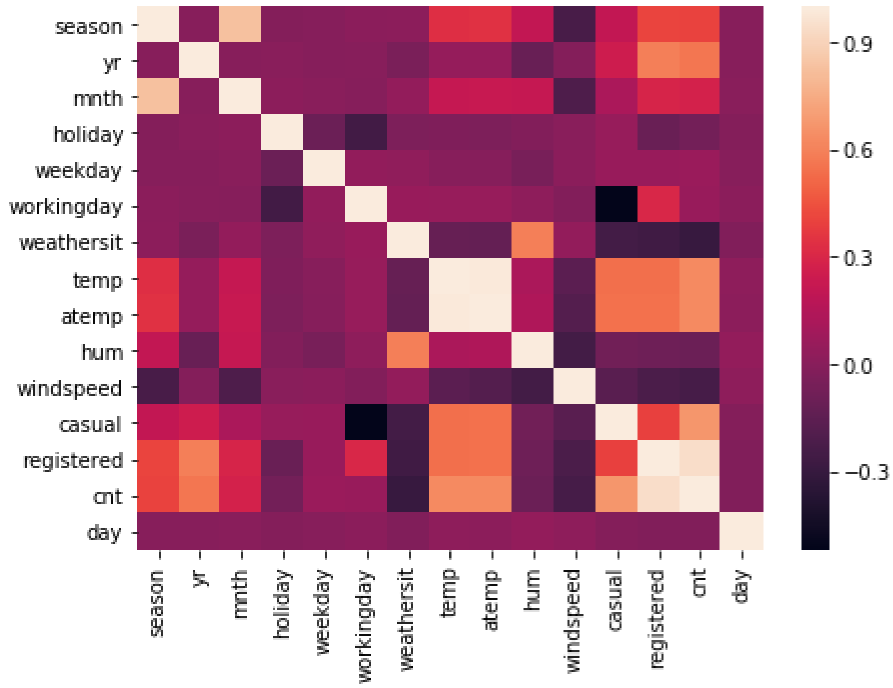
Out[488]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x1e48b605908>,
  <matplotlib.lines.Line2D at 0x1e48b605c50>],
 'caps': [<matplotlib.lines.Line2D at 0x1e48b605f98>,
  <matplotlib.lines.Line2D at 0x1e48b61f320>],
 'boxes': [<matplotlib.lines.Line2D at 0x1e48b605518>],
 'medians': [<matplotlib.lines.Line2D at 0x1e48b61f668>],
 'fliers': [<matplotlib.lines.Line2D at 0x1e48b61f9b0>],
 'means': []}
```



In [ ]:
```
bdf1  = bdf1.drop(["dteday"],axis=1)
bdf1  = bdf1.drop(["date"],axis=1)
```

In [201]:
```
#FEATURE SELECTION
bdf1_corr=bdf1.loc[:,cnames]

f,ax=plt1.subplots(figsize=(7,5))
corr=bdf1_corr.corr()
ax = sns.heatmap(corr)
```



In [202]:
```
#converting season
bdf1["season"] = bdf1.season.map({1: "Spring", 2 : "Summer", 3 : "Fall", 4 :"W
inter" })
```

In [203]:
```
#converting weather situation
bdf1["weathersit"] = bdf1.weathersit.map({1: "clear", 2 : "Cloudy", 3 : "Light
Rain", 4 :"Heavy Rain" })
```

In [204]:
```
#converting working day
bdf1["workingday"] = bdf1.workingday.map({0: "Holiday", 1 : "working" })
```

In [ ]:

In [205]:
```
def cnvert(x):
    ct=x*(39-(-8))+(-8)
    return ct
```

In [206]:
```
bdf1["temp"]=bdf1.temp.apply(cnvert)
```

In [207]:
```
def cnvert1(x):
    ct=x*(50-(-16))+(-16)
    return ct
```

In [208]:
```
bdf1["atemp"]=bdf1.atemp.apply(cnvert1)
```

In [209]:
```
def cnvert2(x):
    ct=x*67
    return ct
```

In [210]:
```
bdf1["windspeed"]=bdf1.windspeed.apply(cnvert2)
```

In [211]:
```
def cnvert3(x):
    ct=x*100
    return ct
```

In [212]:
```
bdf1["hum"]=bdf1.hum.apply(cnvert3)
```

In [213]:
```
bdf1
```

Out[213]:

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atem
0	1	Spring	0	1	0	6	Holiday	Cloudy	8.175849	7.99925
1	2	Spring	0	1	0	0	Holiday	Cloudy	9.083466	7.34677
2	3	Spring	0	1	0	1	working	clear	1.229108	-3.49927
3	4	Spring	0	1	0	2	working	clear	1.400000	-1.99994
4	5	Spring	0	1	0	3	working	clear	2.666979	-0.86818
5	6	Spring	0	1	0	4	working	clear	1.604356	-0.60820
6	7	Spring	0	1	0	5	working	Cloudy	1.236534	-2.21662
7	8	Spring	0	1	0	6	Holiday	Cloudy	-0.245000	-5.29123
8	9	Spring	0	1	0	0	Holiday	clear	-1.498349	-8.33245
9	10	Spring	0	1	0	1	working	clear	-0.910849	-6.04139
10	11	Spring	0	1	0	2	working	Cloudy	-0.052723	-3.36337
11	12	Spring	0	1	0	3	working	clear	0.118169	-5.40878
12	13	Spring	0	1	0	4	working	clear	-0.245000	-6.04172
13	14	Spring	0	1	0	5	working	clear	-0.439110	-3.56474
14	15	Spring	0	1	0	6	Holiday	Cloudy	2.966651	0.37539
15	16	Spring	0	1	0	0	Holiday	clear	2.888349	-0.54167
16	17	Spring	0	1	1	1	Holiday	Cloudy	0.264151	-4.33311
17	18	Spring	0	1	0	2	working	Cloudy	2.183349	-0.66602
18	19	Spring	0	1	0	3	working	Cloudy	5.732178	3.69585
19	20	Spring	0	1	0	4	working	Cloudy	4.298349	0.83330
20	21	Spring	0	1	0	5	working	clear	0.342500	-5.58302
21	22	Spring	0	1	0	6	Holiday	clear	-5.220871	-10.78140
22	23	Spring	0	1	0	0	Holiday	clear	-3.463480	-9.47661
23	24	Spring	0	1	0	1	working	clear	-3.422609	-8.21662
24	25	Spring	0	1	0	2	working	Cloudy	2.503466	-0.52128
25	26	Spring	0	1	0	3	working	Light Rain	2.222500	-2.56240
26	27	Spring	0	1	0	4	working	clear	1.165000	-1.49980
27	28	Spring	0	1	0	5	working	Cloudy	1.563466	-1.26107
28	29	Spring	0	1	0	6	Holiday	clear	1.236534	-1.99968
29	30	Spring	0	1	0	0	Holiday	clear	2.176534	0.52125
...	...	...	...	...	...	...	...	...	...	.
701	702	Winter	1	12	0	0	Holiday	Cloudy	8.332500	7.70772
702	703	Winter	1	12	0	1	working	clear	13.267500	14.08253
703	704	Winter	1	12	0	2	working	clear	14.364151	14.95756
704	705	Winter	1	12	0	3	working	clear	12.601651	12.24879

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atem
705	706	Winter	1	12	0	4	working	clear	4.024151	1.04146
706	707	Winter	1	12	0	5	working	Cloudy	7.079151	5.24922
707	708	Winter	1	12	0	6	Holiday	Cloudy	9.938349	9.70752
708	709	Winter	1	12	0	0	Holiday	Cloudy	10.055849	9.74963
709	710	Winter	1	12	0	1	working	Cloudy	12.484151	12.74795
710	711	Winter	1	12	0	2	working	Cloudy	8.606651	6.33195
711	712	Winter	1	12	0	3	working	Cloudy	5.982500	3.62430
712	713	Winter	1	12	0	4	working	clear	5.904151	3.41640
713	714	Winter	1	12	0	5	working	clear	5.238349	3.41667
714	715	Winter	1	12	0	6	Holiday	clear	7.235849	6.33327
715	716	Winter	1	12	0	0	Holiday	Cloudy	9.037500	8.41590
716	717	Winter	1	12	0	1	working	Cloudy	10.486651	10.49900
717	718	Winter	1	12	0	2	working	clear	11.309151	11.04072
718	719	Winter	1	12	0	3	working	clear	7.627500	6.58269
719	720	Winter	1	12	0	4	working	Cloudy	7.510000	6.12432
720	721	Spring	1	12	0	5	working	Cloudy	7.353349	3.91662
721	722	Spring	1	12	0	6	Holiday	clear	4.494151	-0.41654
722	723	Spring	1	12	0	0	Holiday	clear	3.554151	1.12508
723	724	Spring	1	12	0	1	working	Cloudy	2.871288	1.08740
724	725	Spring	1	12	1	2	Holiday	Cloudy	5.691288	3.43469
725	726	Spring	1	12	0	3	working	Light Rain	3.436651	-1.45802
726	727	Spring	1	12	0	4	working	Cloudy	3.945849	-1.04162
727	728	Spring	1	12	0	5	working	Cloudy	3.906651	0.83303
728	729	Spring	1	12	0	6	Holiday	Cloudy	3.906651	-0.00160
729	730	Spring	1	12	0	0	Holiday	clear	4.024151	-0.70780
730	731	Spring	1	12	0	1	working	Cloudy	2.144151	-1.24985

731 rows × 16 columns

In [215]:
```
sub_spring=bdf1[bdf1.season=="Spring"]
```

In [217]:
```
mean_spring=sub_spring["temp"].mean()
```

In [218]:
```
mean_spring
```

Out[218]:
```
5.994134811049725
```

In [219]:
```
median_spring=sub_spring["temp"].median()
```

In [220]:
```
median_spring
```

Out[220]: 5.434151

In [223]:
```
import statistics as st
```

In [225]:
```
sd_spring=st.pstdev(sub_spring["temp"])
```

In [226]:
```
sd_spring
```

Out[226]: 4.81518884088463

In [227]:
```
sub_winter=bdf1[bdf1.season=="Winter"]
mean_winter=sub_winter["temp"].mean()
median_winter=sub_winter["temp"].median()
sd_winter=st.pstdev(sub_winter["temp"])
```

In [232]:
```
mean_winter
```

Out[232]: 11.876583848314608

In [233]:
```
median_winter
```

Out[233]: 11.2308255

In [234]:
```
sd_winter
```

Out[234]: 5.0539245784284175

In [235]:
```
sub_summer=bdf1[bdf1.season=="Summer"]
mean_summer=sub_summer["temp"].mean()
median_summer=sub_summer["temp"].median()
sd_summer=st.pstdev(sub_summer["temp"])
```

In [236]:
```
mean_summer
```

Out[236]: 17.587042407608696

In [237]:
```
median_summer
```

Out[237]: 18.417924499999998

In [238]:
```
sd_summer
```

Out[238]: 5.748862573816254

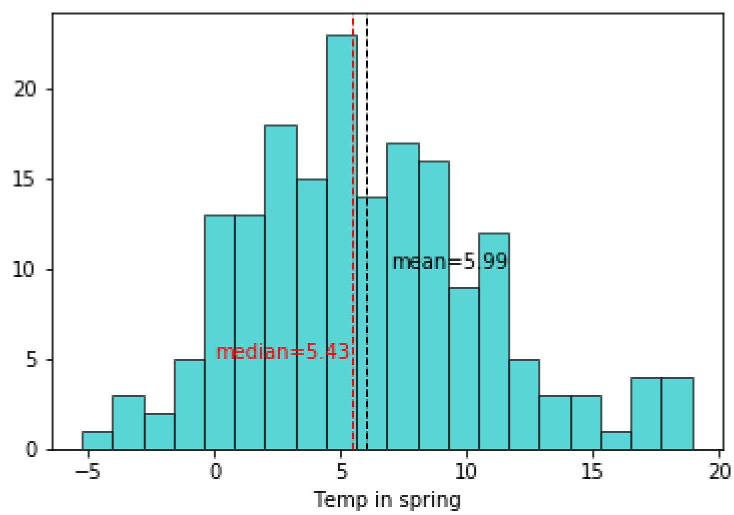
In [239]:
```
sub_fall=bdf1[bdf1.season=="Fall"]
mean_fall=sub_fall["temp"].mean()
median_fall=sub_fall["temp"].median()
sd_fall=st.pstdev(sub_fall["temp"])
```

In [240]:
```
mean_fall
```
Out[240]: 25.196537499999994

In [241]:
```
median_fall
```
Out[241]: 25.585400999999997

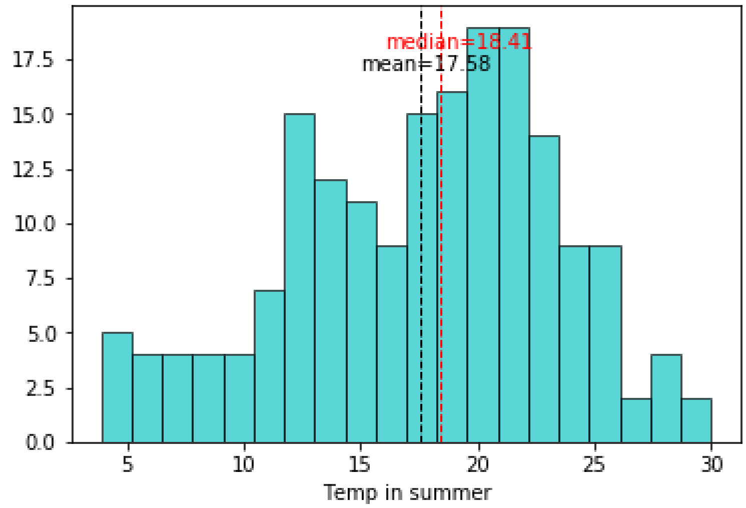
In [242]:
```
sd_fall
```
Out[242]: 3.3209664525970823

In [257]:
```
his_spring=plt1.hist(sub_spring["temp"],bins=20,color='c',edgecolor='k',alpha=
0.65)
plt1.xlabel('Temp in spring')
plt1.axvline(sub_spring["temp"].mean(),color='k',linestyle='dashed',linewidth=
1)
plt1.text(0,5,r'median=5.43',color='red')
plt1.text(7,10,r'mean=5.99',color='black')
plt1.axvline(sub_spring["temp"].median(),color='red',linestyle='dashed',linewi
dth=1)
```
Out[257]: <matplotlib.lines.Line2D at 0x1e4ff8e1208>



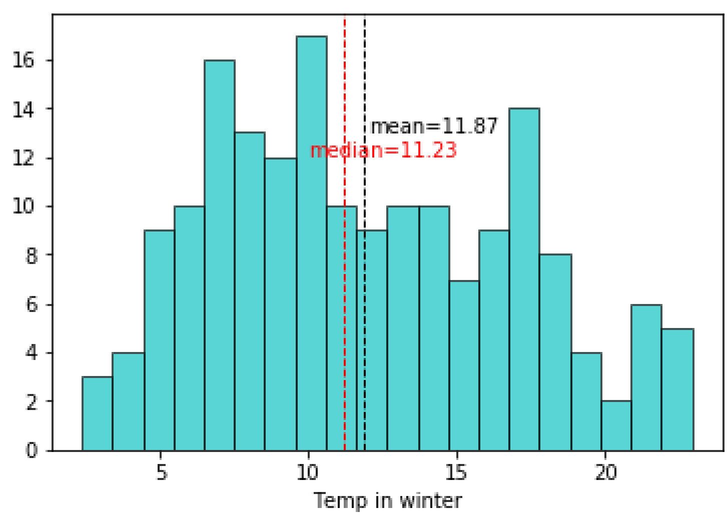
In [260]:
```
his_summer=plt1.hist(sub_summer["temp"],bins=20,color='c',edgecolor='k',alpha=
0.65)
plt1.xlabel('Temp in summer')
plt1.axvline(sub_summer["temp"].mean(),color='k',linestyle='dashed',linewidth=
1)
plt1.text(16,18,r'median=18.41',color='red')
plt1.text(15,17,r'mean=17.58',color='black')
plt1.axvline(sub_summer["temp"].median(),color='red',linestyle='dashed',linewi
dth=1)
```

Out[260]: <matplotlib.lines.Line2D at 0x1e4fc5aa080>



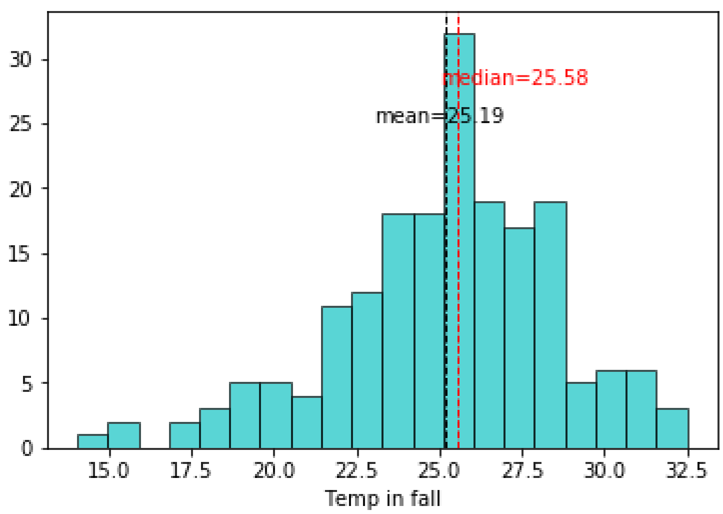
In [261]:
```
his_winter=plt1.hist(sub_winter["temp"],bins=20,color='c',edgecolor='k',alpha=
0.65)
plt1.xlabel('Temp in winter')
plt1.axvline(sub_winter["temp"].mean(),color='k',linestyle='dashed',linewidth=
1)
plt1.text(10,12,r'median=11.23',color='red')
plt1.text(12,13,r'mean=11.87',color='black')
plt1.axvline(sub_winter["temp"].median(),color='red',linestyle='dashed',linewi
dth=1)
```

Out[261]: <matplotlib.lines.Line2D at 0x1e4fe174eb8>



In [262]:
```
his_fall=plt1.hist(sub_fall["temp"],bins=20,color='c',edgecolor='k',alpha=0.65
)
plt1.xlabel('Temp in fall')
plt1.axvline(sub_fall["temp"].mean(),color='k',linestyle='dashed',linewidth=1)
plt1.text(25,28,r'median=25.58',color='red')
plt1.text(23,25,r'mean=25.19',color='black')
plt1.axvline(sub_fall["temp"].median(),color='red',linestyle='dashed',linewidt
h=1)
```

Out[262]: <matplotlib.lines.Line2D at 0x1e4ffa10e48>



In [268]:
```
mean_sprcnt=sub_spring["cnt"].mean()
median_sprcnt=sub_spring["cnt"].median()
sd_sprcnt=st.pstdev(sub_spring["cnt"])
```

In [264]:
```
mean_sprcnt
```

Out[264]: 2604.1325966850827

In [266]:
```
median_sprcnt
```

Out[266]: 2209.0

In [269]:
```
sd_sprcnt
```

Out[269]: 1396.0695197464488

In [270]:
```
mean_sumcnt=sub_summer["cnt"].mean()
median_sumcnt=sub_summer["cnt"].median()
sd_sumcnt=st.pstdev(sub_summer["cnt"])
```

In [271]:
```
mean_sumcnt
```

Out[271]: 4992.33152173913

In [272]:
```
median_sumcnt
```

Out[272]: 4941.5

In [273]:
```
sd_sumcnt
```
Out[273]: 1691.3623219830135

In [275]:
```
mean_wincnt=sub_winter["cnt"].mean()
median_wincnt=sub_winter["cnt"].median()
sd_wincnt=st.pstdev(sub_winter["cnt"])
```

In [276]:
```
mean_wincnt
```
Out[276]: 4728.162921348315

In [277]:
```
median_wincnt
```
Out[277]: 4634.5

In [278]:
```
sd_wincnt
```
Out[278]: 1694.8343365983633

In [279]:
```
mean_falcnt=sub_fall["cnt"].mean()
median_falcnt=sub_fall["cnt"].median()
sd_falcnt=st.pstdev(sub_fall["cnt"])
```

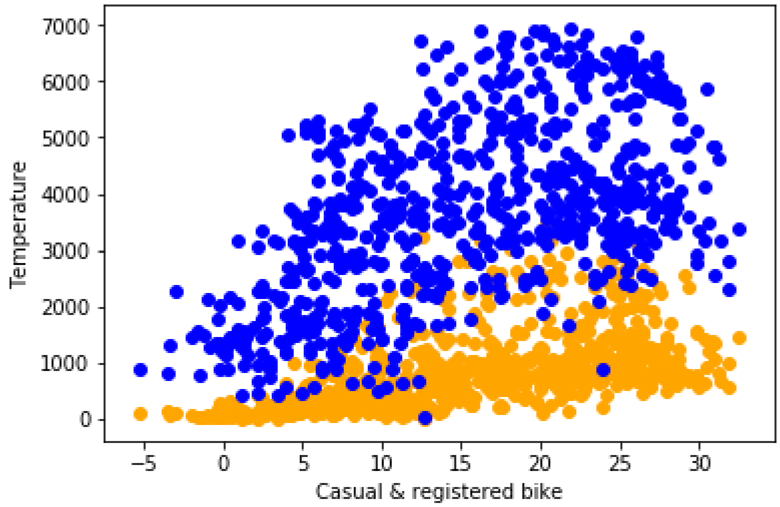
In [280]:
```
mean_falcnt
```
Out[280]: 5644.303191489362

In [281]:
```
median_falcnt
```
Out[281]: 5353.5

In [282]:
```
sd_falcnt
```
Out[282]: 1455.9127569834882

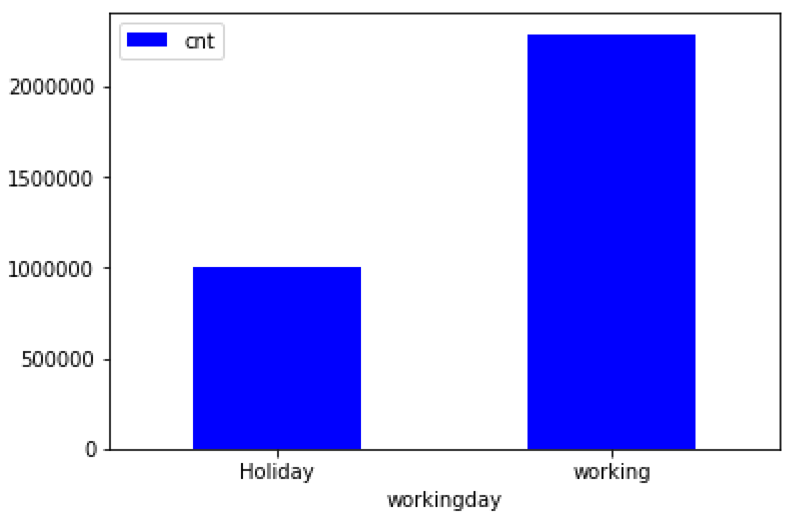
In [302]:
```
# plot of temperature
plt1.scatter(bdf1["temp"],bdf1["casual"],color="orange")
plt1.scatter(bdf1["temp"],bdf1["registered"],color="blue")
plt1.xlabel('Casual & registered bike')
plt1.ylabel('Temperature')
#plt1.text(6,7,"Blue=Registered")
#plt1.text(-5,0,"Red=Casual")
```

Out[302]: Text(0, 0.5, 'Temperature')



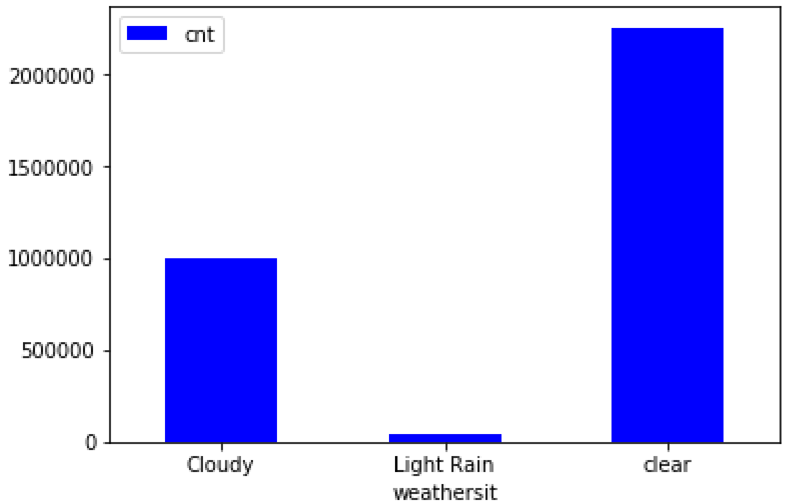
In [328]:
```
#Bar Graph for working day
p=bdf1.groupby('workingday')['cnt'].sum()
p=pd.DataFrame(p)
p.plot.bar(rot=0,color='blue')
```

Out[328]: <matplotlib.axes._subplots.AxesSubplot at 0x1e489f0eb00>



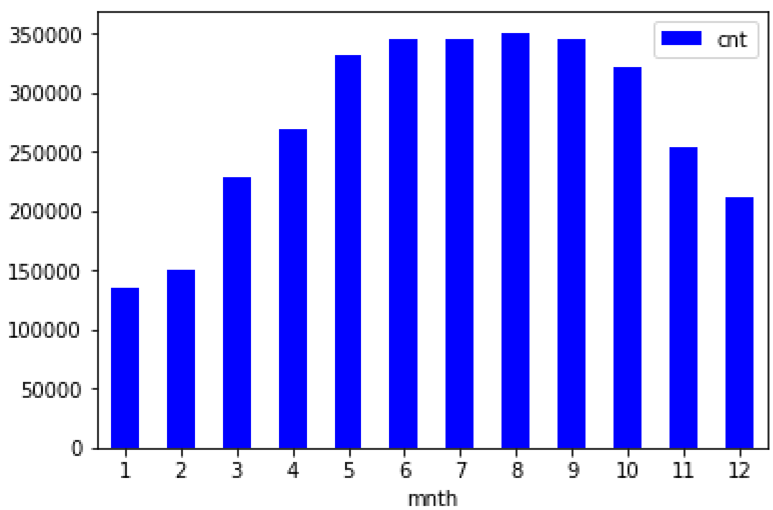
In [327]:
```
#Bar Graph for weather situation
p1=bdf1.groupby('weathersit')['cnt'].sum()
p1=pd.DataFrame(p1)
p1.plot.bar(rot=0,color='blue')
```

Out[327]: <matplotlib.axes._subplots.AxesSubplot at 0x1e489ee6e48>



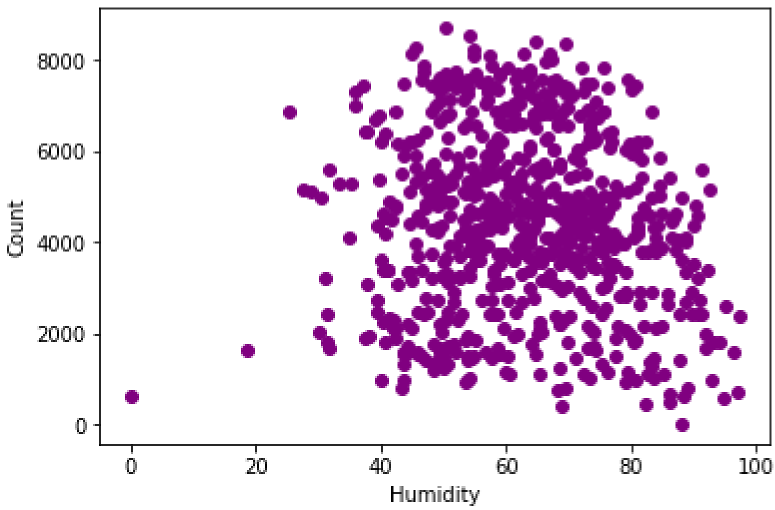
In [326]:
```
#Bar Graph for month
p2=bdf1.groupby('mnth')['cnt'].sum()
p2=pd.DataFrame(p2)
p2.plot.bar(rot=0,color='blue')
```

Out[326]: <matplotlib.axes._subplots.AxesSubplot at 0x1e488cb3cf8>



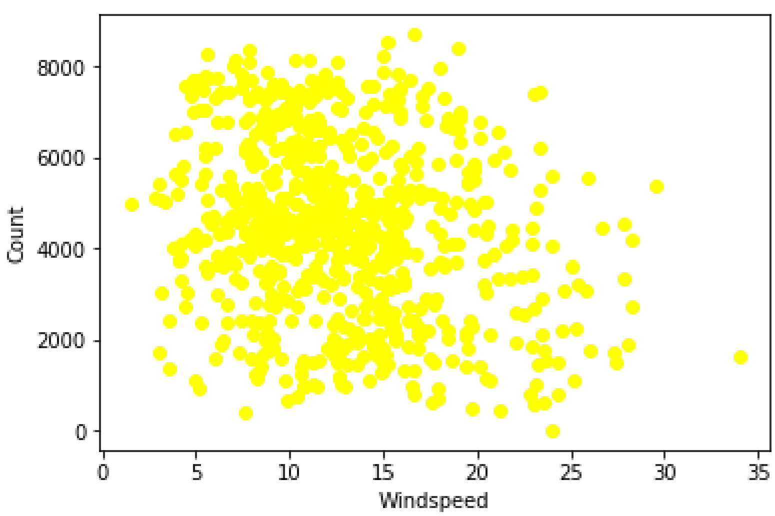
In [330]:
```
#scatter plot of humidity
plt1.scatter(bdf1["hum"],bdf1["cnt"],color="purple")
plt1.ylabel('Count')
plt1.xlabel('Humidity')
```

Out[330]: Text(0.5, 0, 'Humidity')



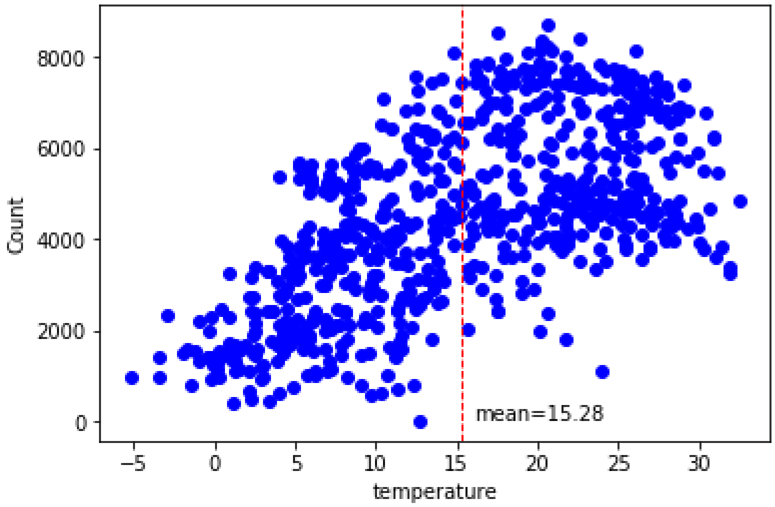
In [333]:
```
#scatter plot of windspeed
plt1.scatter(bdf1["windspeed"],bdf1["cnt"],color="yellow")
plt1.ylabel('Count')
plt1.xlabel('Windspeed')
```

Out[333]: Text(0.5, 0, 'Windspeed')



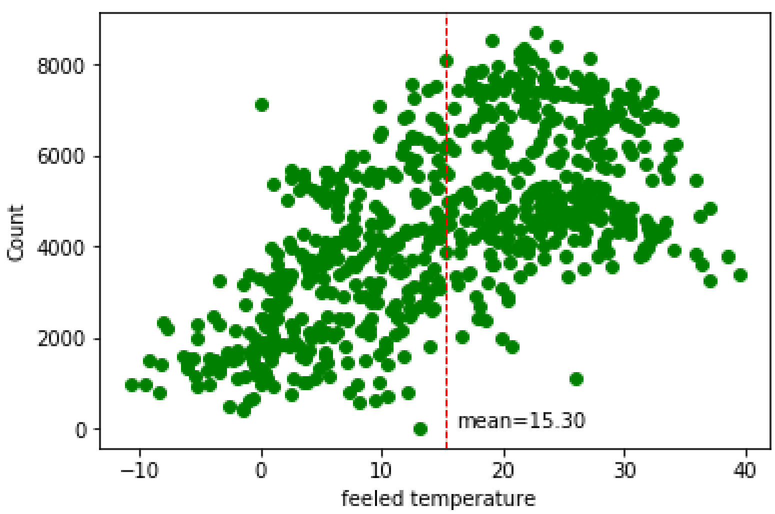
In [343]:
```
#scatter plot of tempature
plt1.scatter(bdf1["temp"],bdf1["cnt"],color="blue")
plt1.ylabel('Count')
plt1.xlabel('temperature')
plt1.axvline(bdf1["temp"].mean(),color='red',linestyle="dashed",linewidth=1)
mean_temp=bdf1["temp"].mean()
plt1.text(16,20,r'mean=15.28',color='black')
```

Out[343]: Text(16, 20, 'mean=15.28')



In [344]:
```
#scatter plot of feeled tempature
plt1.scatter(bdf1["atemp"],bdf1["cnt"],color="Green")
plt1.ylabel('Count')
plt1.xlabel(' feeled temperature')
plt1.axvline(bdf1["atemp"].mean(),color='red',linestyle="dashed",linewidth=1)
mean_atemp=bdf1["atemp"].mean()
plt1.text(16,20,r'mean=15.30',color='black')
```

Out[344]: Text(16, 20, 'mean=15.30')



In [489]:
```
bdf2=bdf
```

In [490]:
```
dropFeatures = ['casual',"dteday","date","registered","holiday","instant","ate
mp"]
bdf2  = bdf2.drop(dropFeatures,axis=1)
```

In [491]:
```
bdf2=bdf2[['day','season','yr','mnth','weekday','workingday','weathersit','tem
p','hum','windspeed','cnt']]

bdf2
```

Out[491]:

	day	season	yr	mnth	weekday	workingday	weathersit	temp	hum	windspeed
0	1	1	0	1	6	0	2	0.344167	0.805833	0.160446
1	2	1	0	1	0	0	2	0.363478	0.696087	0.248539
2	3	1	0	1	1	1	1	0.196364	0.437273	0.248309
3	4	1	0	1	2	1	1	0.200000	0.590435	0.160296
4	5	1	0	1	3	1	1	0.226957	0.436957	0.186900
5	6	1	0	1	4	1	1	0.204348	0.518261	0.089565
6	7	1	0	1	5	1	2	0.196522	0.498696	0.168726
7	8	1	0	1	6	0	2	0.165000	0.535833	0.266804
8	9	1	0	1	0	0	1	0.138333	0.434167	0.361950
9	10	1	0	1	1	1	1	0.150833	0.482917	0.223267
10	11	1	0	1	2	1	2	0.169091	0.686364	0.122132
11	12	1	0	1	3	1	1	0.172727	0.599545	0.304627
12	13	1	0	1	4	1	1	0.165000	0.470417	0.301000
13	14	1	0	1	5	1	1	0.160870	0.537826	0.126548
14	15	1	0	1	6	0	2	0.233333	0.498750	0.157963
15	16	1	0	1	0	0	1	0.231667	0.483750	0.188433
16	17	1	0	1	1	0	2	0.175833	0.537500	0.194017
17	18	1	0	1	2	1	2	0.216667	0.861667	0.146775
18	19	1	0	1	3	1	2	0.292174	0.741739	0.208317
19	20	1	0	1	4	1	2	0.261667	0.538333	0.195904
20	21	1	0	1	5	1	1	0.177500	0.457083	0.353242
21	22	1	0	1	6	0	1	0.059130	0.400000	0.171970
22	23	1	0	1	0	0	1	0.096522	0.436522	0.246600
23	24	1	0	1	1	1	1	0.097391	0.491739	0.158330
24	25	1	0	1	2	1	2	0.223478	0.616957	0.129796
25	26	1	0	1	3	1	3	0.217500	0.862500	0.293850
26	27	1	0	1	4	1	1	0.195000	0.687500	0.113837
27	28	1	0	1	5	1	2	0.203478	0.793043	0.123300
28	29	1	0	1	6	0	1	0.196522	0.651739	0.145365
29	30	1	0	1	0	0	1	0.216522	0.722174	0.073983
...	...	...	...	...	...	...	...	...	...	...
701	2	4	1	12	0	0	2	0.347500	0.823333	0.124379
702	3	4	1	12	1	1	1	0.452500	0.767500	0.082721
703	4	4	1	12	2	1	1	0.475833	0.733750	0.174129
704	5	4	1	12	3	1	1	0.438333	0.485000	0.324021

	day	season	yr	mnth	weekday	workingday	weathersit	temp	hum	windspeed
705	6	4	1	12	4	1	1	0.255833	0.508750	0.174754
706	7	4	1	12	5	1	2	0.320833	0.764167	0.130600
707	8	4	1	12	6	0	2	0.381667	0.911250	0.101379
708	9	4	1	12	0	0	2	0.384167	0.905417	0.157975
709	10	4	1	12	1	1	2	0.435833	0.925000	0.190308
710	11	4	1	12	2	1	2	0.353333	0.596667	0.296037
711	12	4	1	12	3	1	2	0.297500	0.538333	0.162937
712	13	4	1	12	4	1	1	0.295833	0.485833	0.174129
713	14	4	1	12	5	1	1	0.281667	0.642917	0.131229
714	15	4	1	12	6	0	1	0.324167	0.650417	0.106350
715	16	4	1	12	0	0	2	0.362500	0.838750	0.100742
716	17	4	1	12	1	1	2	0.393333	0.907083	0.098258
717	18	4	1	12	2	1	1	0.410833	0.666250	0.221404
718	19	4	1	12	3	1	1	0.332500	0.625417	0.184092
719	20	4	1	12	4	1	2	0.330000	0.667917	0.132463
720	21	1	1	12	5	1	2	0.326667	0.556667	0.374383
721	22	1	1	12	6	0	1	0.265833	0.441250	0.407346
722	23	1	1	12	0	0	1	0.245833	0.515417	0.133083
723	24	1	1	12	1	1	2	0.231304	0.791304	0.077230
724	25	1	1	12	2	0	2	0.291304	0.734783	0.168726
725	26	1	1	12	3	1	3	0.243333	0.823333	0.316546
726	27	1	1	12	4	1	2	0.254167	0.652917	0.350133
727	28	1	1	12	5	1	2	0.253333	0.590000	0.155471
728	29	1	1	12	6	0	2	0.253333	0.752917	0.124383
729	30	1	1	12	0	0	1	0.255833	0.483333	0.350754
730	31	1	1	12	1	1	2	0.215833	0.577500	0.154846

731 rows × 11 columns

In [492]:
```
#Sampling
train, test = train_test_split(bdf2, test_size=0.2)
```

In [493]:
```
test
```

Out[493]:

	day	season	yr	mnth	weekday	workingday	weathersit	temp	hum	windspeed
325	22	4	0	11	2	1	3	0.416667	0.962500	0.118792
474	19	2	1	4	4	1	1	0.498333	0.612500	0.065929
502	17	2	1	5	4	1	1	0.593333	0.520000	0.229475
487	2	2	1	5	3	1	1	0.564167	0.797083	0.138058
141	22	2	0	5	0	0	1	0.604167	0.749583	0.148008
443	19	1	1	3	1	1	1	0.545000	0.728750	0.162317
482	27	2	1	4	5	1	1	0.457500	0.400833	0.347633
125	6	2	0	5	5	1	1	0.479167	0.590000	0.228246
490	5	2	1	5	6	0	2	0.621667	0.756667	0.152992
294	22	4	0	10	6	0	1	0.422500	0.629167	0.092667
26	27	1	0	1	4	1	1	0.195000	0.687500	0.113837
78	20	1	0	3	0	0	1	0.332500	0.473750	0.207721
100	11	2	0	4	1	1	2	0.595652	0.716956	0.324474
274	2	4	0	10	0	0	2	0.356667	0.791667	0.222013
639	1	4	1	10	1	1	2	0.520833	0.649167	0.090804
300	28	4	0	10	5	1	2	0.330833	0.585833	0.229479
101	12	2	0	4	2	1	2	0.502500	0.739167	0.274879
329	26	4	0	11	6	0	1	0.375833	0.681667	0.068421
241	30	3	0	8	2	1	1	0.639167	0.548333	0.125008
415	20	1	1	2	1	0	1	0.280000	0.507826	0.229083
522	6	2	1	6	3	1	1	0.554167	0.611250	0.077125
269	27	4	0	9	2	1	2	0.636667	0.885417	0.118171
149	30	2	0	5	1	0	1	0.733333	0.685000	0.131225
130	11	2	0	5	3	1	1	0.542500	0.632917	0.120642
117	28	2	0	4	4	1	2	0.617500	0.700833	0.320908
427	3	1	1	3	6	0	2	0.414167	0.621250	0.161079
425	1	1	1	3	4	1	1	0.485833	0.615417	0.226987
723	24	1	1	12	1	1	2	0.231304	0.791304	0.077230
478	23	2	1	4	1	1	2	0.321667	0.766667	0.303496
6	7	1	0	1	5	1	2	0.196522	0.498696	0.168726
...	...	...	...	...	...	...	...	...	...	...
528	12	2	1	6	2	1	2	0.653333	0.833333	0.214546
605	28	3	1	8	2	1	1	0.728333	0.620000	0.190925
264	22	3	0	9	4	1	2	0.628333	0.902083	0.128125
40	10	1	0	2	4	1	1	0.144348	0.437391	0.221935

	day	season	yr	mnth	weekday	workingday	weathersit	temp	hum	windspeed
54	24	1	0	2	4	1	2	0.295652	0.697391	0.250496
289	17	4	0	10	1	1	1	0.534167	0.579583	0.175379
28	29	1	0	1	6	0	1	0.196522	0.651739	0.145365
588	11	3	1	8	6	0	2	0.692500	0.732917	0.206479
557	11	3	1	7	3	1	1	0.716667	0.633333	0.151733
305	2	4	0	11	3	1	1	0.377500	0.718750	0.082092
447	23	2	1	3	5	1	2	0.601667	0.694167	0.116300
373	9	1	1	1	1	1	2	0.224167	0.701667	0.098900
653	15	4	1	10	1	1	2	0.561667	0.707500	0.296037
61	3	1	0	3	4	1	1	0.198333	0.318333	0.225754
509	24	2	1	5	4	1	1	0.655000	0.716667	0.172896
225	14	3	0	8	0	0	2	0.676667	0.817500	0.222633
472	17	2	1	4	2	1	1	0.608333	0.390417	0.273629
355	22	1	0	12	4	1	2	0.423333	0.757500	0.047275
98	9	2	0	4	6	0	2	0.342500	0.877500	0.133083
92	3	2	0	4	0	0	1	0.378333	0.480000	0.182213
675	6	4	1	11	2	1	1	0.280833	0.567083	0.173513
44	14	1	0	2	1	1	1	0.415000	0.375833	0.417908
215	4	3	0	8	4	1	2	0.710000	0.757500	0.197150
372	8	1	1	1	0	0	1	0.337500	0.465000	0.191542
292	20	4	0	10	4	1	1	0.475833	0.636250	0.422275
657	19	4	1	10	5	1	2	0.563333	0.815000	0.134954
235	24	3	0	8	3	1	1	0.673333	0.605000	0.253108
680	11	4	1	11	0	0	1	0.420833	0.659167	0.127500
483	28	2	1	4	6	0	2	0.376667	0.489583	0.129975
259	17	3	0	9	6	0	2	0.491667	0.718333	0.189675

147 rows × 11 columns

In [495]:
```
#LINEAR REGRESSION
model = sm.OLS(train.iloc[:,10], train.iloc[:,0:9]).fit()
```

In [496]:
```
predictions_LR = model.predict(test.iloc[:,0:9])
```

In [497]:
```
predictions_LR
```

Out[497]:
```
325    2603.676931
474    5744.737667
502    6187.111501
487    6102.085667
141    3567.825312
443    5214.418517
482    5514.316319
125    3654.835614
490    5668.781131
294    4068.301814
26     1568.899696
78     1555.954239
100    3266.834589
274    2632.803715
639    5917.140938
300    3019.878605
101    2864.699428
329    3775.024870
241    4563.281794
415    3512.900080
522    5931.510737
269    4507.219094
149    4279.940169
130    3800.069364
117    3558.360161
427    4038.011407
425    5223.958898
723    2465.607383
478    3866.660172
6      1005.237920
          ...
528    5699.099248
605    7159.447509
264    4082.559302
40     1269.419973
54     1372.530016
289    4497.500417
28     1461.608329
588    6467.351769
557    7317.779584
305    3921.085557
447    5722.015517
373    2940.702651
653    6080.290932
61     1512.096735
509    6536.727727
225    3768.492005
472    6092.633499
355    1653.198167
98     2157.894022
92     2430.955735
675    5349.549223
44     2395.743942
215    4603.484940
372    3817.732863
292    4457.007144
657    6455.790189
```

```
235    4879.509382
680    5642.759496
483    4221.002654
259    3235.676387
Length: 147, dtype: float64
```

In [498]:
```
from sklearn.metrics import mean_squared_log_error
from math import sqrt
```

In [499]:
```
Rmsle_LR=sqrt(mean_squared_log_error(test['cnt'],predictions_LR))
Rmsle_LR
```

Out[499]: 0.265612846391153

In [500]:
```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms_LR = sqrt(mean_squared_error(test['cnt'],predictions_LR))
rms_LR
```

Out[500]: 831.7785571438695

In [501]:
```
#DECISIONS TREE
fit_DT = DecisionTreeRegressor(max_depth=2).fit(train.iloc[:,0:9], train.iloc
[:,10])
```

In [502]:
```
predictions_DT = fit_DT.predict(test.iloc[:,0:9])
```

In [503]:
```
Rmsle_DT=sqrt(mean_squared_log_error(test['cnt'],predictions_DT))
Rmsle_DT
```

Out[503]: 0.32019966436458364

In [504]:
```
rms_DT = sqrt(mean_squared_error(test['cnt'],predictions_DT))
rms_DT
```

Out[504]: 1000.0559446049203

In [506]:
```
#RANDOM FOREST
RF_model = RandomForestRegressor(n_estimators = 100).fit(train.iloc[:,0:9], tr
ain.iloc[:,10])
```

In [508]:
```
predictions_RF = RF_model.predict(test.iloc[:,0:9])
```

In [509]:
```
Rmsle_RF=sqrt(mean_squared_log_error(test['cnt'],predictions_RF))
Rmsle_RF
```

Out[509]: 0.22436479244688715

In [510]:
```
rms_RF = sqrt(mean_squared_error(test['cnt'],predictions_RF))
rms_RF
```

Out[510]: 690.2795730800418

In [511]:
```
#Random fOrest with 300
RF_model = RandomForestRegressor(n_estimators = 300).fit(train.iloc[:,0:9], tr
ain.iloc[:,10])
predictions_RF = RF_model.predict(test.iloc[:,0:9])
Rmsle_RF=sqrt(mean_squared_log_error(test['cnt'],predictions_RF))
rms_RF = sqrt(mean_squared_error(test['cnt'],predictions_RF))
```

In [512]:
```
rms_RF
```

Out[512]: 671.8935463572394

In [513]:
```
Rmsle_RF
```

Out[513]: 0.22018108122078306

In [514]:
```
#Random fOrest with 500
RF_model = RandomForestRegressor(n_estimators = 500).fit(train.iloc[:,0:9], tr
ain.iloc[:,10])
predictions_RF = RF_model.predict(test.iloc[:,0:9])
Rmsle_RF=sqrt(mean_squared_log_error(test['cnt'],predictions_RF))
rms_RF = sqrt(mean_squared_error(test['cnt'],predictions_RF))
```

In [515]:
```
rms_RF
```

Out[515]: 667.8353528588938

In [534]:
```
Rmsle_RF
```

Out[534]: 0.21813592693400397

In [ ]:
```
#Random fOrest with 700
RF_model = RandomForestRegressor(n_estimators = 700).fit(train.iloc[:,0:10], t
rain.iloc[:,11])
predictions_RF = RF_model.predict(test.iloc[:,0:10])
Rmsle_RF=sqrt(mean_squared_log_error(test['cnt'],predictions_RF))
rms_RF = sqrt(mean_squared_error(test['cnt'],predictions_RF))
```

In [469]:
```
rms_RF
```

Out[469]: 667.4561329775676

In [535]:
```
Rmsle_RF
```

Out[535]: 0.21813592693400397

In [518]:
```
#KNN IMPUTATION
KNN_model=KNeighborsRegressor(n_neighbors=3).fit(train.iloc[:,0:9], train.iloc
[:,10])
```

In [519]:
```
KNN_Predictions=KNN_model.predict(test.iloc[:,0:9])
```

In [520]:
```
KNN_Predictions
```

Out[520]:
```
array([3983.33333333, 5269.66666667, 4155.33333333, 5826.66666667,
       4274.66666667, 5138.        , 4794.        , 5202.33333333,
       6032.66666667, 5434.66666667, 2899.33333333, 1957.66666667,
       4538.66666667, 4676.33333333, 4092.33333333, 5975.33333333,
       5335.66666667, 6029.33333333, 7339.33333333, 2524.33333333,
       6442.66666667, 5423.33333333, 5873.        , 4843.        ,
       4382.        , 2759.33333333, 3844.66666667, 1184.66666667,
       5326.66666667, 2221.        , 2221.        , 5861.66666667,
       4687.33333333, 6255.33333333, 6798.33333333, 6725.66666667,
       4792.66666667, 2431.        , 2064.        , 1759.66666667,
       4913.33333333, 5895.        , 5238.66666667, 5374.66666667,
       5427.33333333, 4301.33333333, 2320.        , 2781.33333333,
       4371.        , 5746.66666667, 5526.66666667, 1930.66666667,
       5536.33333333, 5253.33333333, 5516.33333333, 3950.        ,
       5401.33333333, 6348.33333333, 5378.33333333, 6347.66666667,
       6012.        , 5117.33333333, 4307.66666667, 3606.        ,
       6085.66666667, 3996.        , 1927.        , 5043.66666667,
       4949.66666667, 3878.        , 4883.66666667, 2691.        ,
       2822.        , 4069.        , 1957.66666667, 4508.        ,
       6645.33333333, 6702.66666667, 6088.66666667, 4747.        ,
       4944.66666667, 6242.        , 5738.33333333, 5117.        ,
       5006.33333333, 5687.33333333, 4212.33333333, 4532.33333333,
       4652.66666667, 5884.        , 2374.        , 5287.33333333,
       5250.66666667, 3893.33333333, 2073.66666667, 4712.66666667,
       4783.        , 7542.        , 5074.66666667, 4353.        ,
       5680.66666667, 6030.        , 5278.66666667, 4312.66666667,
       1894.33333333, 5669.        , 6489.        , 4283.33333333,
       4568.        , 6137.33333333, 3579.66666667, 4901.66666667,
       2805.33333333, 3956.        , 4611.        , 4514.        ,
       5065.33333333, 6114.33333333, 6423.33333333, 4707.33333333,
       2394.        , 2356.33333333, 6481.33333333, 2882.        ,
       5093.66666667, 5998.        , 4593.66666667, 4576.66666667,
       1983.33333333, 6219.66666667, 2611.66666667, 5478.        ,
       5750.66666667, 3836.66666667, 2830.66666667, 3386.        ,
       4158.        , 4776.33333333, 2275.33333333, 6094.        ,
       1385.66666667, 5723.33333333, 6400.66666667, 7084.        ,
       4709.66666667, 4888.33333333, 6978.66666667])
```

In [521]:
```
rms_KNN = sqrt(mean_squared_error(test['cnt'],KNN_Predictions))
rms_KNN
```

Out[521]:
```
1519.6985685421384
```

In [522]:
```
Rmsle_KNN=sqrt(mean_squared_log_error(test['cnt'],KNN_Predictions))
Rmsle_KNN
```

Out[522]: 0.4043876169424441

In [523]:
```
#KNN with 5 neighbour
KNN_model=KNeighborsRegressor(n_neighbors=5).fit(train.iloc[:,0:9], train.iloc
[:,10])
KNN_Predictions=KNN_model.predict(test.iloc[:,0:9])
rms_KNN = sqrt(mean_squared_error(test['cnt'],KNN_Predictions))
Rmsle_KNN=sqrt(mean_squared_log_error(test['cnt'],KNN_Predictions))
```

In [524]:
```
rms_KNN
```

Out[524]: 1662.1508252822289

In [525]:
```
Rmsle_KNN
```

Out[525]: 0.43275639640883745

In [526]:
```
#KNN with 7 neighbour
KNN_model=KNeighborsRegressor(n_neighbors=7).fit(train.iloc[:,0:9], train.iloc
[:,10])
KNN_Predictions=KNN_model.predict(test.iloc[:,0:9])
rms_KNN = sqrt(mean_squared_error(test['cnt'],KNN_Predictions))
Rmsle_KNN=sqrt(mean_squared_log_error(test['cnt'],KNN_Predictions))
```

In [527]:
```
rms_KNN
```

Out[527]: 1729.6704805430352

In [528]:
```
Rmsle_KNN
```

Out[528]: 0.45175974165637767

In [529]:
```
#KNN with 9 neighbour
KNN_model=KNeighborsRegressor(n_neighbors=9).fit(train.iloc[:,0:9], train.iloc
[:,10])
KNN_Predictions=KNN_model.predict(test.iloc[:,0:9])
rms_KNN = sqrt(mean_squared_error(test['cnt'],KNN_Predictions))
Rmsle_KNN=sqrt(mean_squared_log_error(test['cnt'],KNN_Predictions))
```

In [530]:
```
rms_KNN
```

Out[530]: 1708.9750265792075

In [531]:
```
Rmsle_KNN
```

Out[531]: 0.4539029050198128

In [ ]: