Regression_Assignment

March 19, 2019

Import all the required libraries

```
In [9]: import pandas as pd
        import numpy as np
        import matplotlib.pylab as plt
        %matplotlib inline
        from matplotlib.pylab import rcParams
        rcParams['figure.figsize'] = 15, 6
        from datetime import datetime
        from time import *
        from sklearn import metrics
        # from pandas.stats.api import ols
```

Read data into dataframe with dateparser

```
In [10]: dateparse = lambda dates: pd.datetime.strptime(dates, '%Y-%m-%d')
         df = pd.read_csv('day.csv', parse_dates=['dteday'], date_parser=dateparse)
```

```
In [11]: print(df)
```

```
     instant     dteday  season  yr  mnth  holiday  weekday  workingday  \
0          1 2011-01-01       1   0     1        0        6           0
1          2 2011-01-02       1   0     1        0        0           0
2          3 2011-01-03       1   0     1        0        1           1
3          4 2011-01-04       1   0     1        0        2           1
4          5 2011-01-05       1   0     1        0        3           1
5          6 2011-01-06       1   0     1        0        4           1
6          7 2011-01-07       1   0     1        0        5           1
7          8 2011-01-08       1   0     1        0        6           0
8          9 2011-01-09       1   0     1        0        0           0
9         10 2011-01-10       1   0     1        0        1           1
10        11 2011-01-11       1   0     1        0        2           1
11        12 2011-01-12       1   0     1        0        3           1
12        13 2011-01-13       1   0     1        0        4           1
13        14 2011-01-14       1   0     1        0        5           1
14        15 2011-01-15       1   0     1        0        6           0
15        16 2011-01-16       1   0     1        0        0           0
16        17 2011-01-17       1   0     1        1        1           0
```

17	18	2011-01-18	1	0	1	0	2	1
18	19	2011-01-19	1	0	1	0	3	1
19	20	2011-01-20	1	0	1	0	4	1
20	21	2011-01-21	1	0	1	0	5	1
21	22	2011-01-22	1	0	1	0	6	0
22	23	2011-01-23	1	0	1	0	0	0
23	24	2011-01-24	1	0	1	0	1	1
24	25	2011-01-25	1	0	1	0	2	1
25	26	2011-01-26	1	0	1	0	3	1
26	27	2011-01-27	1	0	1	0	4	1
27	28	2011-01-28	1	0	1	0	5	1
28	29	2011-01-29	1	0	1	0	6	0
29	30	2011-01-30	1	0	1	0	0	0
..	...	...	...	..	...	...	...	...
701	702	2012-12-02	4	1	12	0	0	0
702	703	2012-12-03	4	1	12	0	1	1
703	704	2012-12-04	4	1	12	0	2	1
704	705	2012-12-05	4	1	12	0	3	1
705	706	2012-12-06	4	1	12	0	4	1
706	707	2012-12-07	4	1	12	0	5	1
707	708	2012-12-08	4	1	12	0	6	0
708	709	2012-12-09	4	1	12	0	0	0
709	710	2012-12-10	4	1	12	0	1	1
710	711	2012-12-11	4	1	12	0	2	1
711	712	2012-12-12	4	1	12	0	3	1
712	713	2012-12-13	4	1	12	0	4	1
713	714	2012-12-14	4	1	12	0	5	1
714	715	2012-12-15	4	1	12	0	6	0
715	716	2012-12-16	4	1	12	0	0	0
716	717	2012-12-17	4	1	12	0	1	1
717	718	2012-12-18	4	1	12	0	2	1
718	719	2012-12-19	4	1	12	0	3	1
719	720	2012-12-20	4	1	12	0	4	1
720	721	2012-12-21	1	1	12	0	5	1
721	722	2012-12-22	1	1	12	0	6	0
722	723	2012-12-23	1	1	12	0	0	0
723	724	2012-12-24	1	1	12	0	1	1
724	725	2012-12-25	1	1	12	1	2	0
725	726	2012-12-26	1	1	12	0	3	1
726	727	2012-12-27	1	1	12	0	4	1
727	728	2012-12-28	1	1	12	0	5	1
728	729	2012-12-29	1	1	12	0	6	0
729	730	2012-12-30	1	1	12	0	0	0
730	731	2012-12-31	1	1	12	0	1	1

	weathersit	temp	atemp	hum	windspeed	casual	registered	\
0	2	0.344167	0.363625	0.805833	0.160446	331	654	
1	2	0.363478	0.353739	0.696087	0.248539	131	670	

2	1	0.196364	0.189405	0.437273	0.248309	120	1229
3	1	0.200000	0.212122	0.590435	0.160296	108	1454
4	1	0.226957	0.229270	0.436957	0.186900	82	1518
5	1	0.204348	0.233209	0.518261	0.089565	88	1518
6	2	0.196522	0.208839	0.498696	0.168726	148	1362
7	2	0.165000	0.162254	0.535833	0.266804	68	891
8	1	0.138333	0.116175	0.434167	0.361950	54	768
9	1	0.150833	0.150888	0.482917	0.223267	41	1280
10	2	0.169091	0.191464	0.686364	0.122132	43	1220
11	1	0.172727	0.160473	0.599545	0.304627	25	1137
12	1	0.165000	0.150883	0.470417	0.301000	38	1368
13	1	0.160870	0.188413	0.537826	0.126548	54	1367
14	2	0.233333	0.248112	0.498750	0.157963	222	1026
15	1	0.231667	0.234217	0.483750	0.188433	251	953
16	2	0.175833	0.176771	0.537500	0.194017	117	883
17	2	0.216667	0.232333	0.861667	0.146775	9	674
18	2	0.292174	0.298422	0.741739	0.208317	78	1572
19	2	0.261667	0.255050	0.538333	0.195904	83	1844
20	1	0.177500	0.157833	0.457083	0.353242	75	1468
21	1	0.059130	0.079070	0.400000	0.171970	93	888
22	1	0.096522	0.098839	0.436522	0.246600	150	836
23	1	0.097391	0.117930	0.491739	0.158330	86	1330
24	2	0.223478	0.234526	0.616957	0.129796	186	1799
25	3	0.217500	0.203600	0.862500	0.293850	34	472
26	1	0.195000	0.219700	0.687500	0.113837	15	416
27	2	0.203478	0.223317	0.793043	0.123300	38	1129
28	1	0.196522	0.212126	0.651739	0.145365	123	975
29	1	0.216522	0.250322	0.722174	0.073983	140	956
..	...	...	...	...	...	...	...
701	2	0.347500	0.359208	0.823333	0.124379	892	3757
702	1	0.452500	0.455796	0.767500	0.082721	555	5679
703	1	0.475833	0.469054	0.733750	0.174129	551	6055
704	1	0.438333	0.428012	0.485000	0.324021	331	5398
705	1	0.255833	0.258204	0.508750	0.174754	340	5035
706	2	0.320833	0.321958	0.764167	0.130600	349	4659
707	2	0.381667	0.389508	0.911250	0.101379	1153	4429
708	2	0.384167	0.390146	0.905417	0.157975	441	2787
709	2	0.435833	0.435575	0.925000	0.190308	329	4841
710	2	0.353333	0.338363	0.596667	0.296037	282	5219
711	2	0.297500	0.297338	0.538333	0.162937	310	5009
712	1	0.295833	0.294188	0.485833	0.174129	425	5107
713	1	0.281667	0.294192	0.642917	0.131229	429	5182
714	1	0.324167	0.338383	0.650417	0.106350	767	4280
715	2	0.362500	0.369938	0.838750	0.100742	538	3248
716	2	0.393333	0.401500	0.907083	0.098258	212	4373
717	1	0.410833	0.409708	0.666250	0.221404	433	5124
718	1	0.332500	0.342162	0.625417	0.184092	333	4934
719	2	0.330000	0.335217	0.667917	0.132463	314	3814

720	2	0.326667	0.301767	0.556667	0.374383	221	3402
721	1	0.265833	0.236113	0.441250	0.407346	205	1544
722	1	0.245833	0.259471	0.515417	0.133083	408	1379
723	2	0.231304	0.258900	0.791304	0.077230	174	746
724	2	0.291304	0.294465	0.734783	0.168726	440	573
725	3	0.243333	0.220333	0.823333	0.316546	9	432
726	2	0.254167	0.226642	0.652917	0.350133	247	1867
727	2	0.253333	0.255046	0.590000	0.155471	644	2451
728	2	0.253333	0.242400	0.752917	0.124383	159	1182
729	1	0.255833	0.231700	0.483333	0.350754	364	1432
730	2	0.215833	0.223487	0.577500	0.154846	439	2290

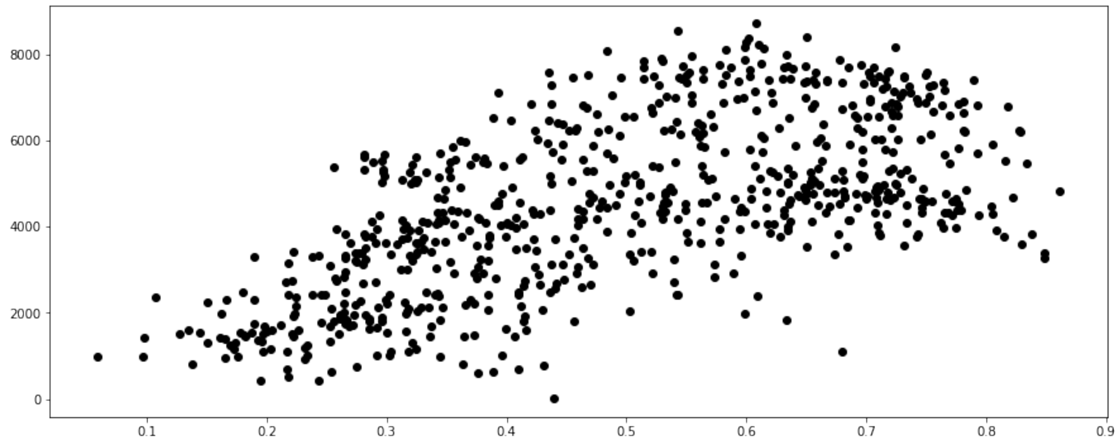
	cnt
0	985
1	801
2	1349
3	1562
4	1600
5	1606
6	1510
7	959
8	822
9	1321
10	1263
11	1162
12	1406
13	1421
14	1248
15	1204
16	1000
17	683
18	1650
19	1927
20	1543
21	981
22	986
23	1416
24	1985
25	506
26	431
27	1167
28	1098
29	1096
..	...
701	4649
702	6234
703	6606
704	5729

```
705    5375
706    5008
707    5582
708    3228
709    5170
710    5501
711    5319
712    5532
713    5611
714    5047
715    3786
716    4585
717    5557
718    5267
719    4128
720    3623
721    1749
722    1787
723     920
724    1013
725     441
726    2114
727    3095
728    1341
729    1796
730    2729

[731 rows x 16 columns]
```

Plotting cnt with respect to temperature

```
In [12]: plt.plot(df['temp'], df['cnt'], 'o', color='black');
```



cnt is increasing with temperature

Correlation between temp and cnt

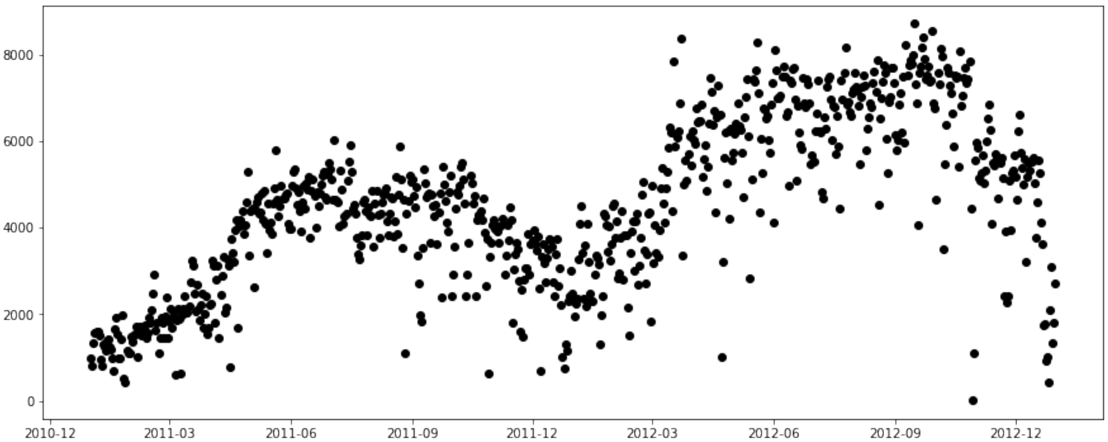
```
In [13]: df['temp'].corr(df['cnt'])
```

```
Out[13]: 0.6274940090334918
```

Plotting cnt with respect to dteday

```
In [14]: plt.plot(df['dteday'], df['cnt'], 'o', color='black')
```

```
Out[14]: [<matplotlib.lines.Line2D at 0x1a1cdbde48>]
```



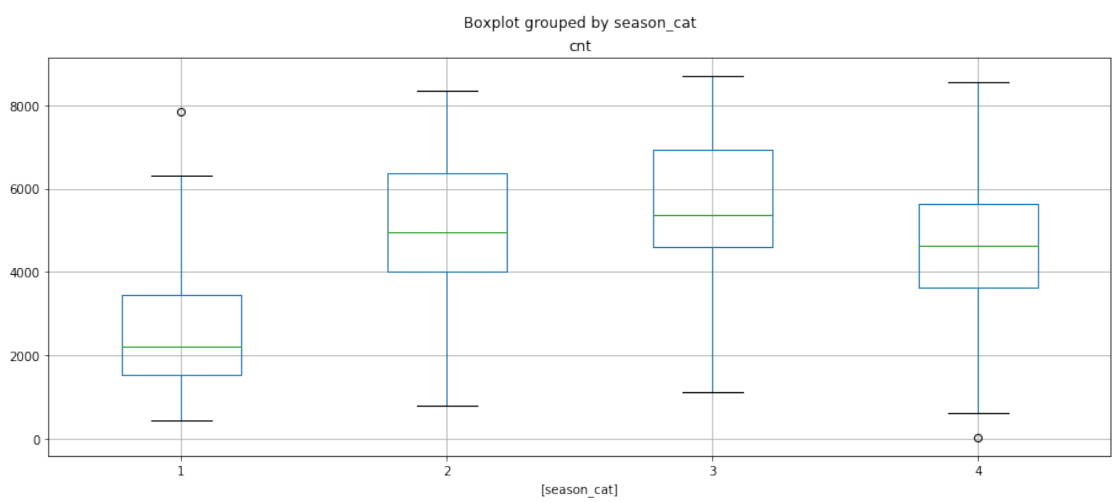
Changing season to categorical

```
In [15]: df['season_cat'] = pd.Categorical(df['season'])
```

Boxplot of cnt with respect to season

```
In [16]: df[['season_cat', 'cnt']].boxplot(by = 'season_cat')
```

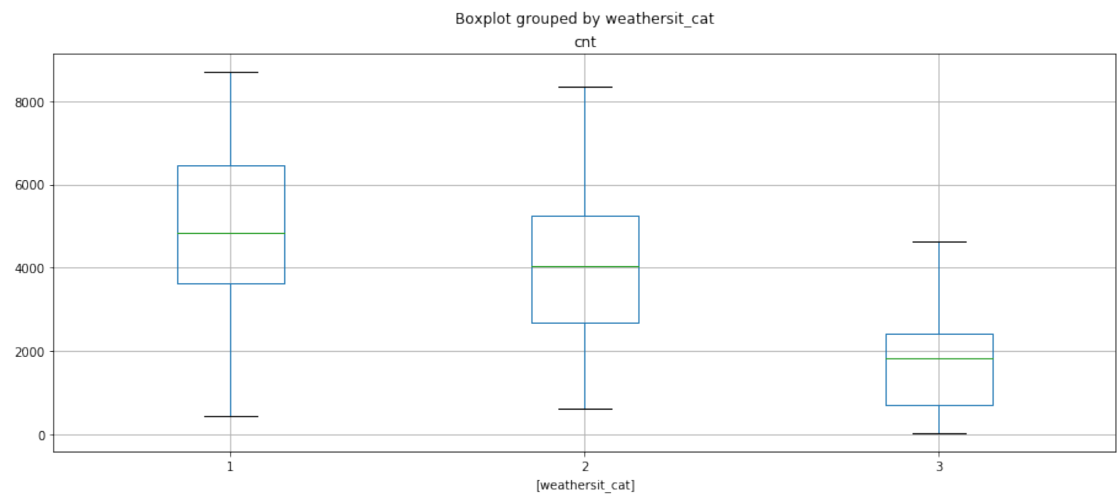
```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1d22da90>
```



Changing season to categorical and plotting boxplot

```
In [17]: df['weathersit_cat'] = pd.Categorical(df['weathersit'])
         df[['weathersit_cat', 'cnt']].boxplot(by = 'weathersit_cat')
```

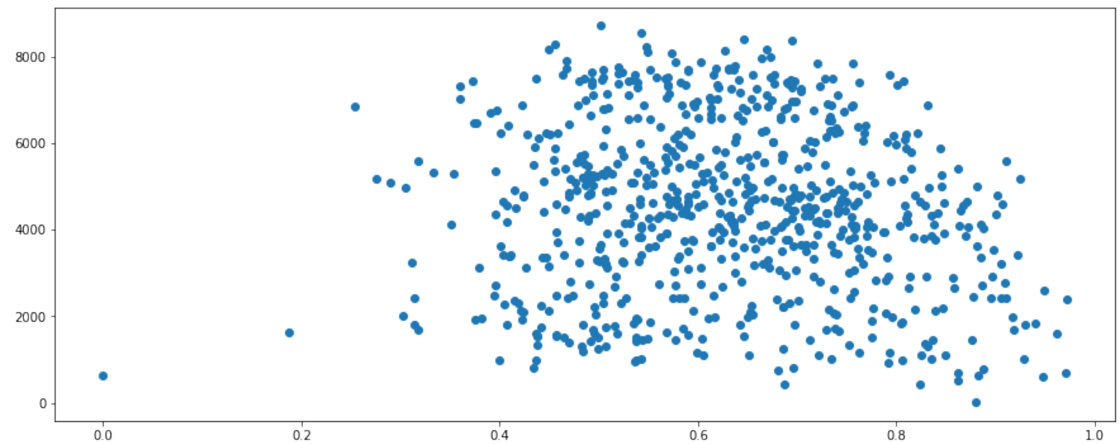
```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1a1d41f9b0>
```



Plottint cnt with respect to hum

```
In [18]: plt.plot(df['hum'], df['cnt'], 'o')
```

```
Out[18]: [<matplotlib.lines.Line2D at 0x1a1d79a1d0>]
```



It doesnt capture any meaningful relationship between humidity and target variable

Correlation between hum and cnt

```
In [19]: df['hum'].corr(df['cnt'])
```

```
Out[19]: -0.10065856213715531
```

There is also a worst correlation between humidity and cnt

```
In [20]: df['windspeed'].describe()
```

```
Out[20]: count    731.000000
         mean       0.190486
         std        0.077498
         min        0.022392
         25%        0.134950
         50%        0.180975
         75%        0.233214
         max        0.507463
         Name: windspeed, dtype: float64
```

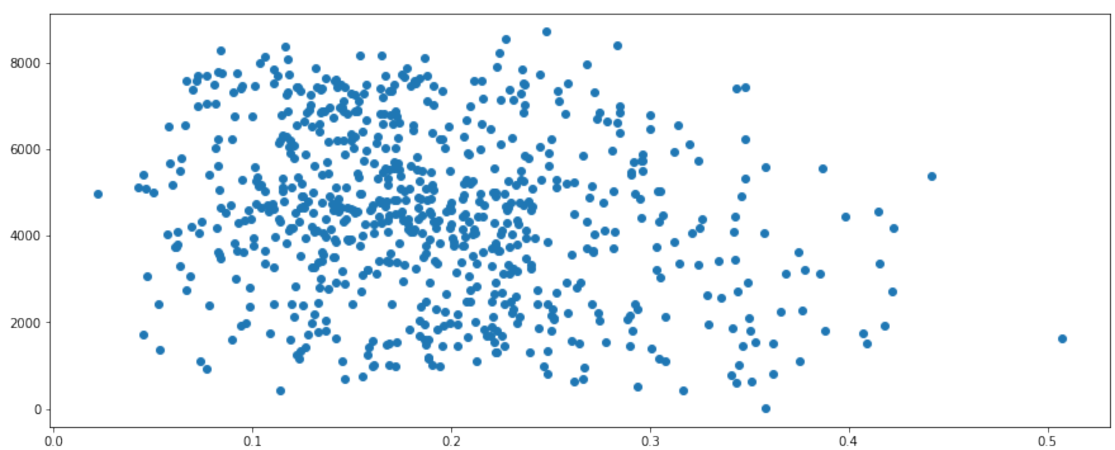
Correlation between windspeed and cnt

```
In [21]: df['windspeed'].corr(df['cnt'])
```

```
Out[21]: -0.23454499742167
```

```
In [22]: plt.plot(df['windspeed'], df['cnt'], 'o')
```

```
Out[22]: [<matplotlib.lines.Line2D at 0x1a1d7ecba8>]
```




```
In [23]: df['dteday'] = df['dteday'].map(datetime.toordinal)
         print(df['dteday'])
```

0	734138
1	734139
2	734140
3	734141
4	734142
5	734143
6	734144
7	734145
8	734146
9	734147
10	734148
11	734149
12	734150
13	734151
14	734152
15	734153
16	734154
17	734155
18	734156
19	734157
20	734158
21	734159
22	734160
23	734161
24	734162
25	734163
26	734164
27	734165
28	734166
29	734167
	...
701	734839
702	734840
703	734841
704	734842
705	734843
706	734844
707	734845
708	734846
709	734847
710	734848
711	734849
712	734850
713	734851
714	734852
715	734853
716	734854
717	734855

```
718    734856
719    734857
720    734858
721    734859
722    734860
723    734861
724    734862
725    734863
726    734864
727    734865
728    734866
729    734867
730    734868
Name: dteday, Length: 731, dtype: int64
```

In [24]:
```
dates = df['dteday']
dates = (dates-dates.min())/(dates.max()-dates.min())
df['dteday'] = dates
print(df['dteday'])
```

```
0      0.000000
1      0.001370
2      0.002740
3      0.004110
4      0.005479
5      0.006849
6      0.008219
7      0.009589
8      0.010959
9      0.012329
10     0.013699
11     0.015068
12     0.016438
13     0.017808
14     0.019178
15     0.020548
16     0.021918
17     0.023288
18     0.024658
19     0.026027
20     0.027397
21     0.028767
22     0.030137
23     0.031507
24     0.032877
25     0.034247
26     0.035616
```

```
27     0.036986
28     0.038356
29     0.039726
         ...
701    0.960274
702    0.961644
703    0.963014
704    0.964384
705    0.965753
706    0.967123
707    0.968493
708    0.969863
709    0.971233
710    0.972603
711    0.973973
712    0.975342
713    0.976712
714    0.978082
715    0.979452
716    0.980822
717    0.982192
718    0.983562
719    0.984932
720    0.986301
721    0.987671
722    0.989041
723    0.990411
724    0.991781
725    0.993151
726    0.994521
727    0.995890
728    0.997260
729    0.998630
730    1.000000
Name: dteday, Length: 731, dtype: float64
```

In [25]:
```python
print(df['yr'])
```

```
0      0
1      0
2      0
3      0
4      0
5      0
6      0
7      0
8      0
```

9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
	..
701	1
702	1
703	1
704	1
705	1
706	1
707	1
708	1
709	1
710	1
711	1
712	1
713	1
714	1
715	1
716	1
717	1
718	1
719	1
720	1
721	1
722	1
723	1
724	1
725	1
726	1

```
727    1
728    1
729    1
730    1
Name: yr, Length: 731, dtype: int64
```

```
In [26]: dates = df['yr']
         dates = (dates-dates.min())/(dates.max()-dates.min())
         df['yr'] = dates
         print(df['yr'])
```

```
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
5      0.0
6      0.0
7      0.0
8      0.0
9      0.0
10     0.0
11     0.0
12     0.0
13     0.0
14     0.0
15     0.0
16     0.0
17     0.0
18     0.0
19     0.0
20     0.0
21     0.0
22     0.0
23     0.0
24     0.0
25     0.0
26     0.0
27     0.0
28     0.0
29     0.0
      ...
701    1.0
702    1.0
703    1.0
704    1.0
705    1.0
```

```
706    1.0
707    1.0
708    1.0
709    1.0
710    1.0
711    1.0
712    1.0
713    1.0
714    1.0
715    1.0
716    1.0
717    1.0
718    1.0
719    1.0
720    1.0
721    1.0
722    1.0
723    1.0
724    1.0
725    1.0
726    1.0
727    1.0
728    1.0
729    1.0
730    1.0
Name: yr, Length: 731, dtype: float64
```

Correlation between dteday and cnt

In [27]:
```
df['dteday'].corr(df['cnt'])
```

Out[27]:
```
0.6288302722083061
```

Correlation between windspeed and temp

In [28]:
```
df['season'].corr(df['temp'])
```

Out[28]:
```
0.3343148563990949
```

In [29]:
```
df.describe()
```

Out[29]:
```
          instant      dteday      season          yr        mnth     holiday  \
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
mean   366.000000    0.500000    2.496580    0.500684    6.519836    0.028728
std    211.165812    0.289268    1.110807    0.500342    3.451913    0.167155
min      1.000000    0.000000    1.000000    0.000000    1.000000    0.000000
25%    183.500000    0.250000    2.000000    0.000000    4.000000    0.000000
50%    366.000000    0.500000    3.000000    1.000000    7.000000    0.000000
```

```
75%     548.500000    0.750000    3.000000    1.000000   10.000000    0.000000
max     731.000000    1.000000    4.000000    1.000000   12.000000    1.000000

          weekday  workingday  weathersit        temp       atemp         hum  \
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
mean     2.997264    0.683995    1.395349    0.495385    0.474354    0.627894
std      2.004787    0.465233    0.544894    0.183051    0.162961    0.142429
min      0.000000    0.000000    1.000000    0.059130    0.079070    0.000000
25%      1.000000    0.000000    1.000000    0.337083    0.337842    0.520000
50%      3.000000    1.000000    1.000000    0.498333    0.486733    0.626667
75%      5.000000    1.000000    2.000000    0.655417    0.608602    0.730209
max      6.000000    1.000000    3.000000    0.861667    0.840896    0.972500

        windspeed      casual   registered          cnt
count  731.000000  731.000000   731.000000   731.000000
mean     0.190486  848.176471  3656.172367  4504.348837
std      0.077498  686.622488  1560.256377  1937.211452
min      0.022392    2.000000    20.000000    22.000000
25%      0.134950  315.500000  2497.000000  3152.000000
50%      0.180975  713.000000  3662.000000  4548.000000
75%      0.233214 1096.000000  4776.500000  5956.000000
max      0.507463 3410.000000  6946.000000  8714.000000
```

Cost fuction

```
In [30]: def cost_function(X, Y, B):
             m = len(Y)
             J = np.sum((X.dot(B) - Y) ** 2)
             J = J / (2 * m)
             return J
```

Function to find RMSE

```
In [31]: def RMSE(X, Y, B):
             m = len(Y)
             J = np.sum((X.dot(B).astype(int) - Y) ** 2)
             J = J**0.5
             J = J / (2 * m)
             return J
```

Function to find R-squared value

```
In [32]: def R_sqrd(X, Y, B):
           Ybar = np.mean(Y)
           ssreg = np.sum((X.dot(B) - Ybar)**2)
           sstot = np.sum((Y - Ybar)**2)
           return ssreg/sstot
```

Function to do Linear Regression

```
In [33]: def linear_regression(df, features, target):

             train_df = df[:int(len(df)*0.7)]
             test_df = df[int(len(df)*0.7):]
             m = len(train_df)
             x0 = np.ones(m)
             X = np.array([x0] + [train_df[x] for x in features]).T
             B = np.array([0] + [0 for x in features])
             Y = np.array(train_df[target])
             alpha = 0.0001
             inital_cost = cost_function(X, Y, B)
             print('Initial cost:', inital_cost)
             newB, cost_history = gradient_descent(X, Y, B, alpha, 100000)
             print(newB)
             print('RMSE on train:', RMSE(X, Y, newB))
             print('R squared value on train:', R_sqrd(X, Y, newB))
             print(test_df)
             m = len(test_df)
             x0 = np.ones(m)
             X_ = np.array([x0] + [test_df[x] for x in features]).T
             Y_ = np.array(test_df[target])
             print('RMSE on test:', RMSE(X_, Y_, newB))
             print('R squared value on train:', R_sqrd(X_, Y_, newB))
             return newB
```

Function to do gradient descent

```
In [34]: def gradient_descent(X, Y, B, alpha, iterations):
               cost_history = [0] * iterations
               m = len(Y)

               for iteration in range(iterations):
                   h = X.dot(B)
                   loss = h - Y
                   gradient = X.T.dot(loss) / m
                   B = B - alpha * gradient
                   cost = cost_function(X, Y, B)
                   cost_history[iteration] = cost

               return B, cost_history
```

Doing linear regression with the features and finding RMSE and R-squared

```
In [35]: features = ['dteday', 'holiday', 'weekday',
                 'workingday', 'hum', 'windspeed']
         features = ['season', 'yr', 'mnth', 'temp', 'weathersit']
         linear_regression(df, features, 'cnt')
         print('min count',min(df['cnt']))
         print('max count',max(df['cnt']))
```

```
Initial cost: 8453737.078277886
[1445.91356325  759.03302101 2061.06937797   -7.92474148 1655.87187535
 -478.89947287]
RMSE on train: 22.029837680382563
R squared value on train: 0.45231669359976956
     instant    dteday  season   yr  mnth  holiday  weekday  workingday  \
511      512  0.700000       2  1.0     5        0        6           0
512      513  0.701370       2  1.0     5        0        0           0
513      514  0.702740       2  1.0     5        1        1           0
514      515  0.704110       2  1.0     5        0        2           1
515      516  0.705479       2  1.0     5        0        3           1
516      517  0.706849       2  1.0     5        0        4           1
517      518  0.708219       2  1.0     6        0        5           1
518      519  0.709589       2  1.0     6        0        6           0
519      520  0.710959       2  1.0     6        0        0           0
520      521  0.712329       2  1.0     6        0        1           1
521      522  0.713699       2  1.0     6        0        2           1
522      523  0.715068       2  1.0     6        0        3           1
523      524  0.716438       2  1.0     6        0        4           1
524      525  0.717808       2  1.0     6        0        5           1
525      526  0.719178       2  1.0     6        0        6           0
526      527  0.720548       2  1.0     6        0        0           0
527      528  0.721918       2  1.0     6        0        1           1
528      529  0.723288       2  1.0     6        0        2           1
529      530  0.724658       2  1.0     6        0        3           1
530      531  0.726027       2  1.0     6        0        4           1
531      532  0.727397       2  1.0     6        0        5           1
532      533  0.728767       2  1.0     6        0        6           0
533      534  0.730137       2  1.0     6        0        0           0
534      535  0.731507       2  1.0     6        0        1           1
535      536  0.732877       2  1.0     6        0        2           1
536      537  0.734247       2  1.0     6        0        3           1
537      538  0.735616       3  1.0     6        0        4           1
538      539  0.736986       3  1.0     6        0        5           1
539      540  0.738356       3  1.0     6        0        6           0
540      541  0.739726       3  1.0     6        0        0           0
..       ...       ...     ...  ...   ...      ...      ...         ...
701      702  0.960274       4  1.0    12        0        0           0
702      703  0.961644       4  1.0    12        0        1           1
703      704  0.963014       4  1.0    12        0        2           1
704      705  0.964384       4  1.0    12        0        3           1
705      706  0.965753       4  1.0    12        0        4           1
706      707  0.967123       4  1.0    12        0        5           1
707      708  0.968493       4  1.0    12        0        6           0
708      709  0.969863       4  1.0    12        0        0           0
709      710  0.971233       4  1.0    12        0        1           1
710      711  0.972603       4  1.0    12        0        2           1
711      712  0.973973       4  1.0    12        0        3           1
```

712	713	0.975342	4	1.0	12	0	4	1
713	714	0.976712	4	1.0	12	0	5	1
714	715	0.978082	4	1.0	12	0	6	0
715	716	0.979452	4	1.0	12	0	0	0
716	717	0.980822	4	1.0	12	0	1	1
717	718	0.982192	4	1.0	12	0	2	1
718	719	0.983562	4	1.0	12	0	3	1
719	720	0.984932	4	1.0	12	0	4	1
720	721	0.986301	1	1.0	12	0	5	1
721	722	0.987671	1	1.0	12	0	6	0
722	723	0.989041	1	1.0	12	0	0	0
723	724	0.990411	1	1.0	12	0	1	1
724	725	0.991781	1	1.0	12	1	2	0
725	726	0.993151	1	1.0	12	0	3	1
726	727	0.994521	1	1.0	12	0	4	1
727	728	0.995890	1	1.0	12	0	5	1
728	729	0.997260	1	1.0	12	0	6	0
729	730	0.998630	1	1.0	12	0	0	0
730	731	1.000000	1	1.0	12	0	1	1

	weathersit	temp	atemp	hum	windspeed	casual	registered \
511	1	0.692500	0.642696	0.732500	0.198992	2855	3681
512	1	0.690000	0.641425	0.697083	0.215171	3283	3308
513	1	0.712500	0.679300	0.676250	0.196521	2557	3486
514	1	0.722500	0.672992	0.684583	0.295400	880	4863
515	2	0.656667	0.611129	0.670000	0.134329	745	6110
516	1	0.680000	0.631329	0.492917	0.195279	1100	6238
517	2	0.654167	0.607962	0.755417	0.237563	533	3594
518	1	0.583333	0.566288	0.549167	0.186562	2795	5325
519	1	0.602500	0.575133	0.493333	0.184087	2494	5147
520	1	0.597500	0.578283	0.487083	0.284833	1071	5927
521	2	0.540833	0.525892	0.613333	0.209575	968	6033
522	1	0.554167	0.542292	0.611250	0.077125	1027	6028
523	1	0.602500	0.569442	0.567083	0.157350	1038	6456
524	1	0.649167	0.597862	0.467917	0.175383	1488	6248
525	1	0.710833	0.648367	0.437083	0.144287	2708	4790
526	1	0.726667	0.663517	0.538333	0.133721	2224	4374
527	2	0.720833	0.659721	0.587917	0.207713	1017	5647
528	2	0.653333	0.597875	0.833333	0.214546	477	4495
529	1	0.655833	0.611117	0.582083	0.343279	1173	6248
530	1	0.648333	0.624383	0.569583	0.253733	1180	6183
531	1	0.639167	0.599754	0.589583	0.176617	1563	6102
532	1	0.631667	0.594708	0.504167	0.166667	2963	4739
533	1	0.592500	0.571975	0.598750	0.144904	2634	4344
534	2	0.568333	0.544842	0.777917	0.174746	653	4446
535	1	0.688333	0.654692	0.690000	0.148017	968	5857
536	1	0.782500	0.720975	0.592083	0.113812	872	5339
537	1	0.805833	0.752542	0.567917	0.118787	778	5127

538	1	0.777500	0.724121	0.573750	0.182842	964	4859
539	1	0.731667	0.652792	0.534583	0.179721	2657	4801
540	1	0.743333	0.674254	0.479167	0.145525	2551	4340
..	...	...	...	...	...	...	...
701	2	0.347500	0.359208	0.823333	0.124379	892	3757
702	1	0.452500	0.455796	0.767500	0.082721	555	5679
703	1	0.475833	0.469054	0.733750	0.174129	551	6055
704	1	0.438333	0.428012	0.485000	0.324021	331	5398
705	1	0.255833	0.258204	0.508750	0.174754	340	5035
706	2	0.320833	0.321958	0.764167	0.130600	349	4659
707	2	0.381667	0.389508	0.911250	0.101379	1153	4429
708	2	0.384167	0.390146	0.905417	0.157975	441	2787
709	2	0.435833	0.435575	0.925000	0.190308	329	4841
710	2	0.353333	0.338363	0.596667	0.296037	282	5219
711	2	0.297500	0.297338	0.538333	0.162937	310	5009
712	1	0.295833	0.294188	0.485833	0.174129	425	5107
713	1	0.281667	0.294192	0.642917	0.131229	429	5182
714	1	0.324167	0.338383	0.650417	0.106350	767	4280
715	2	0.362500	0.369938	0.838750	0.100742	538	3248
716	2	0.393333	0.401500	0.907083	0.098258	212	4373
717	1	0.410833	0.409708	0.666250	0.221404	433	5124
718	1	0.332500	0.342162	0.625417	0.184092	333	4934
719	2	0.330000	0.335217	0.667917	0.132463	314	3814
720	2	0.326667	0.301767	0.556667	0.374383	221	3402
721	1	0.265833	0.236113	0.441250	0.407346	205	1544
722	1	0.245833	0.259471	0.515417	0.133083	408	1379
723	2	0.231304	0.258900	0.791304	0.077230	174	746
724	2	0.291304	0.294465	0.734783	0.168726	440	573
725	3	0.243333	0.220333	0.823333	0.316546	9	432
726	2	0.254167	0.226642	0.652917	0.350133	247	1867
727	2	0.253333	0.255046	0.590000	0.155471	644	2451
728	2	0.253333	0.242400	0.752917	0.124383	159	1182
729	1	0.255833	0.231700	0.483333	0.350754	364	1432
730	2	0.215833	0.223487	0.577500	0.154846	439	2290

	cnt	season_cat	weathersit_cat
511	6536	2	1
512	6591	2	1
513	6043	2	1
514	5743	2	1
515	6855	2	2
516	7338	2	1
517	4127	2	2
518	8120	2	1
519	7641	2	1
520	6998	2	1
521	7001	2	2
522	7055	2	1

523	7494	2	1
524	7736	2	1
525	7498	2	1
526	6598	2	1
527	6664	2	2
528	4972	2	2
529	7421	2	1
530	7363	2	1
531	7665	2	1
532	7702	2	1
533	6978	2	1
534	5099	2	2
535	6825	2	1
536	6211	2	1
537	5905	3	1
538	5823	3	1
539	7458	3	1
540	6891	3	1
..	...	...	...
701	4649	4	2
702	6234	4	1
703	6606	4	1
704	5729	4	1
705	5375	4	1
706	5008	4	2
707	5582	4	2
708	3228	4	2
709	5170	4	2
710	5501	4	2
711	5319	4	2
712	5532	4	1
713	5611	4	1
714	5047	4	1
715	3786	4	2
716	4585	4	2
717	5557	4	1
718	5267	4	1
719	4128	4	2
720	3623	1	2
721	1749	1	1
722	1787	1	1
723	920	1	2
724	1013	1	2
725	441	1	3
726	2114	1	2
727	3095	1	2
728	1341	1	2
729	1796	1	1

```
730  2729          1              2

[220 rows x 18 columns]
RMSE on test: 47.11257892511518
R squared value on train: 0.17404490604303183
min count 22
max count 8714
```