02052021 - Data Exploration

February 27, 2021

[1]:
```
%pylab inline

# help in creating the graphs in the notebook otherwise they will pop up
import pandas as pd
import seaborn as sns
```

```
Populating the interactive namespace from numpy and matplotlib
```

[2]:
```
train_data = pd.read_csv('train.csv')
train_data.head()
```

[2]:

	instant	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	\
0	1	1	0	1	0	0	6	0	1	
1	2	1	0	1	1	0	6	0	1	
2	3	1	0	1	2	0	6	0	1	
3	4	1	0	1	3	0	6	0	1	
4	5	1	0	1	4	0	6	0	1	

	temp	atemp	hum	windspeed	casual	registered	cnt
0	0.24	0.2879	0.81	0.0	3	13	16
1	0.22	0.2727	0.80	0.0	8	32	40
2	0.22	0.2727	0.80	0.0	5	27	32
3	0.24	0.2879	0.75	0.0	3	10	13
4	0.24	0.2879	0.75	0.0	0	1	1

1 Attribute Information:

- instant: record index
- season : season (1:spring, 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 [Web Link])
- 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 loudy

  - 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 derived via (t-t_min)/(t_max-t_min), t_min=-8, t_max=+39 (only in hourly scale)
- atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min), t_min=-16, t_max=+50 (only in hourly scale)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

[3]:
```
train_data.describe(include = 'all').transpose()
```

[3]:

	count	mean	std	min	25%	50%	75%	max
instant	13035.0	6518.000000	3763.024714	1.00	3259.5000	6518.0000	9776.5000	13035.0000
season	13035.0	2.214806	1.052064	1.00	1.0000	2.0000	3.0000	4.0000
yr	13035.0	0.336786	0.472629	0.00	0.0000	0.0000	1.0000	1.0000
mnth	13035.0	5.548293	3.297280	1.00	3.0000	5.0000	8.0000	12.0000
hr	13035.0	11.550288	6.912504	0.00	6.0000	12.0000	18.0000	23.0000
holiday	13035.0	0.027388	0.163217	0.00	0.0000	0.0000	0.0000	1.0000
weekday	13035.0	3.002762	2.006777	0.00	1.0000	3.0000	5.0000	6.0000
workingday	13035.0	0.683698	0.465050	0.00	0.0000	1.0000	1.0000	1.0000
weathersit	13035.0	1.425853	0.647530	1.00	1.0000	1.0000	2.0000	4.0000
temp	13035.0	0.482389	0.191656	0.02	0.3200	0.4800	0.6400	0.9800
atemp	13035.0	0.463317	0.171546	0.00	0.3182	0.4697	0.6061	1.0000
hum	13035.0	0.623282	0.199746	0.00	0.4600	0.6200	0.7900	1.0000
windspeed	13035.0	0.196035	0.124183	0.00	0.1045	0.1940		
casual	13035.0	32.527733	46.655799	0.00	3.0000	14.0000		
registered	13035.0	135.249405	131.879162	0.00	30.0000	103.0000		
cnt	13035.0	167.777138	160.786886	1.00	35.0000	124.0000		

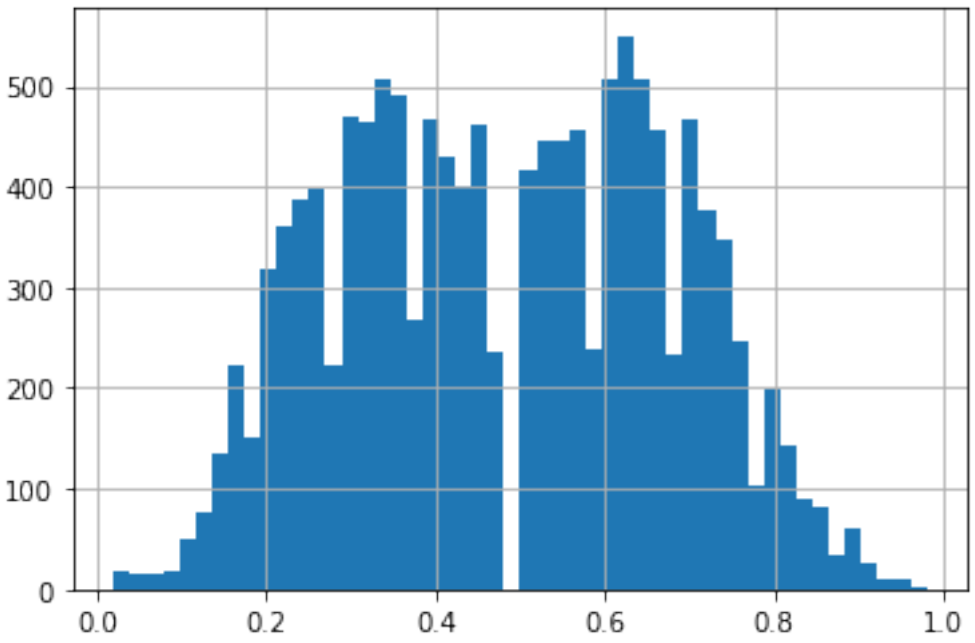
windspeed	0.2836	0.8507
casual	42.0000	367.0000
registered	194.0000	796.0000
cnt	246.0000	957.0000

2 Univariate Analysis

2.1 Continuous Data

2.1.1 Histogram

[4]:
```
train_data['temp'].hist(bins = 50); # normalized values - (t-t_min)/
↪(t_max-t_min)
```



Most people won't rent a bike when it is too hot or too cold. More bikes are rented when the temperature is average.

High number of bikes are rented when the temp is around 0.3 and 0.7

2.1.2 Q. Check histograms for atemp, hum and windspeed

2.1.3 Categorical Variable

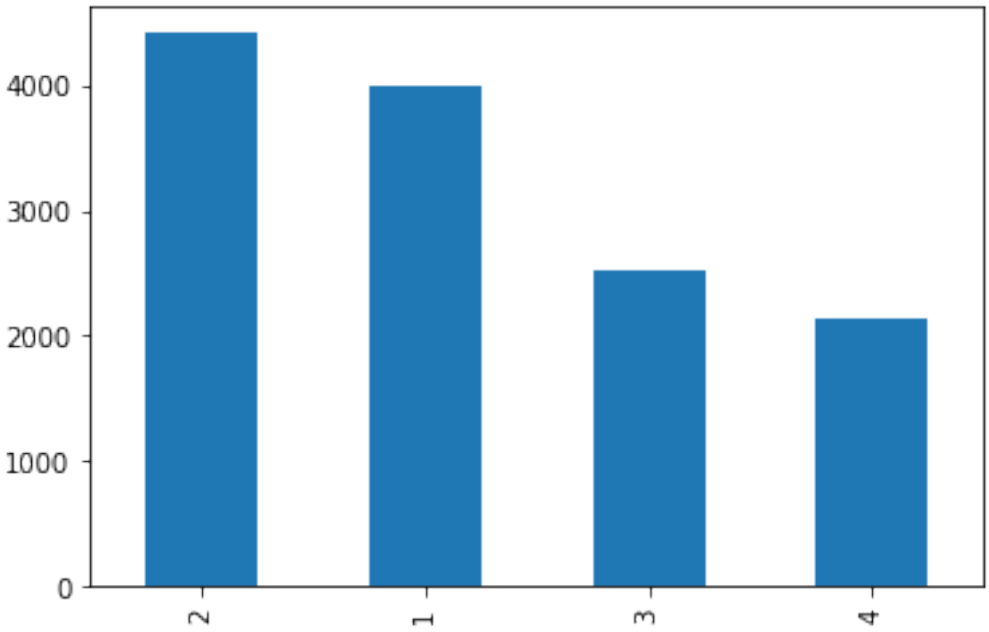
[5]:
```
train_data['season'].unique()
```

[5]:
```
array([1, 2, 3, 4])
```

[6]:
```
train_data['season'].value_counts() # 1:spring, 2:summer, 3:fall, 4:winter
```

[6]:
```
2    4409
1    3980
3    2512
4    2134
Name: season, dtype: int64
```

[7]:
```
train_data['season'].value_counts().plot(kind = 'bar');
```



[8]:
```
print (3980 + 4409)
print (2512 + 2134)

# 1:spring, 2:summer, 3:fall, 4:winter
```

```
8389
4646
```

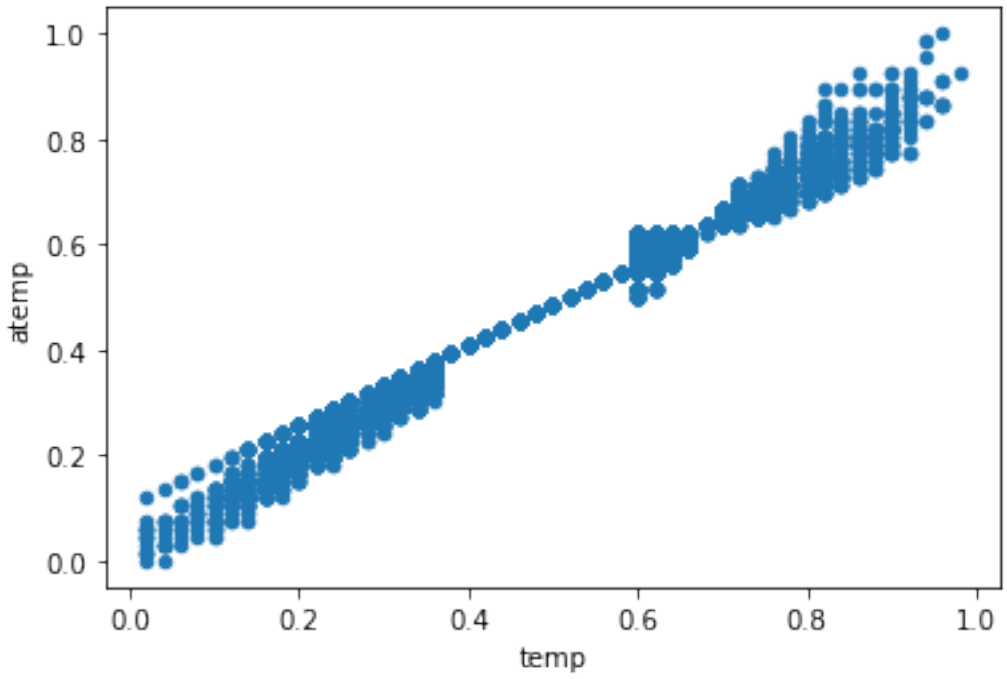
Season 1 and 2 have twice the number of instances of Bike Sharing than season 3 and 4. This, intuitively, makes sense as season 1 & 2 are spring and summer which would be pleasant and season 3 & 4 would be fall & winter which will be cold.

3 Bivariate Analysis

3.1 Continuous & Continuous

3.1.1 Scatter Plot

[9]:
```
train_data.plot.scatter('temp', 'atemp');
```



Temp and atemp have a positive linear relation

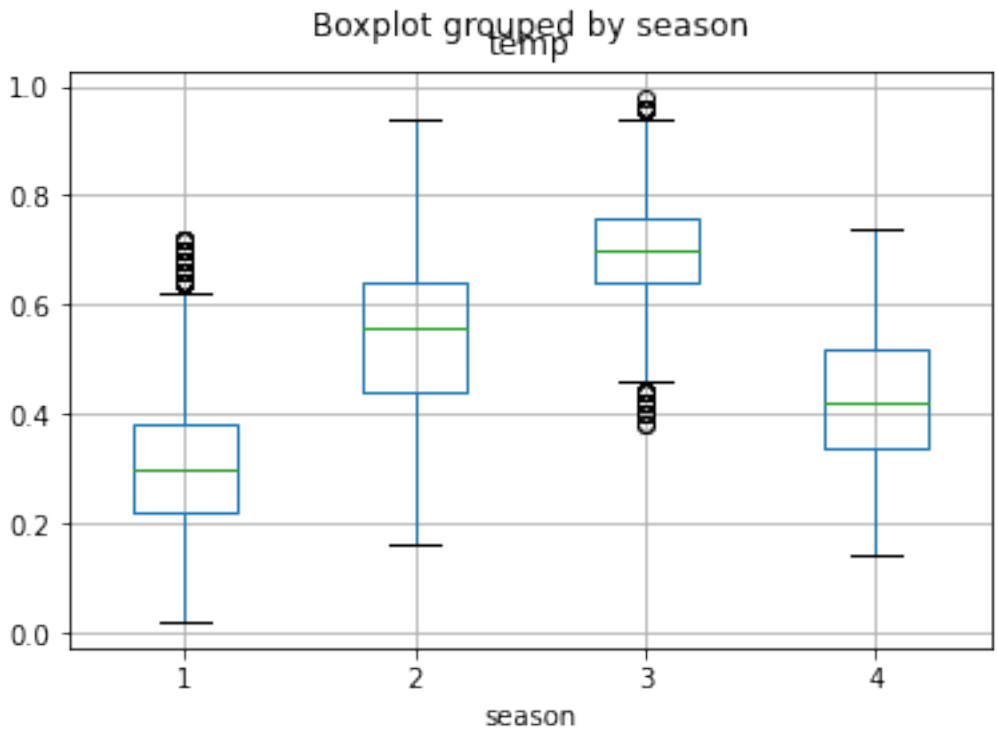
[10]:
```
train_data.temp.corr(train_data.atemp)
```

[10]:
```
0.9918673349213908
```

3.2 Categorical & Continuous

3.2.1 Boxplots of Continuous Variable over the categories of Categorical Variable

[11]:
```
train_data.boxplot(column = 'temp', by = 'season');  # 1:spring, 2:summer, 3:
↪fall, 4:winter
```



Similar to for temp and season

3.3 Categorical & Categorical

3.3.1 Pivot Tables

[13]:
```
train_data.head()
```

[13]:

	instant	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	\
0	1	1	0	1	0	0	6	0	1	
1	2	1	0	1	1	0	6	0	1	
2	3	1	0	1	2	0	6	0	1	
3	4	1	0	1	3	0	6	0	1	
4	5	1	0	1	4	0	6	0	1	

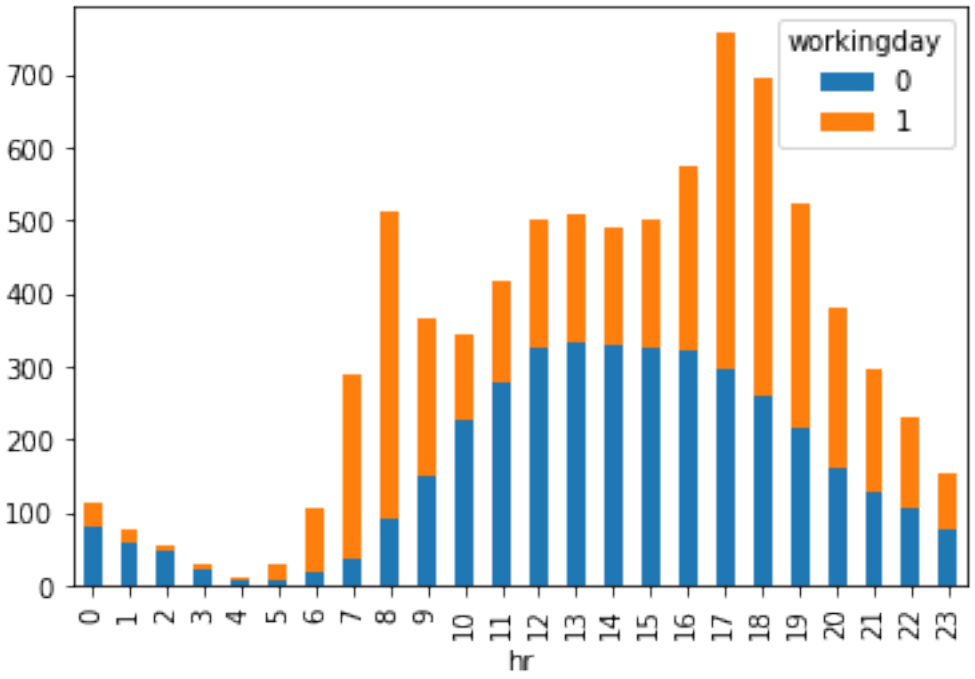
	temp	atemp	hum	windspeed	casual	registered	cnt
0	0.24	0.2879	0.81	0.0	3	13	16
1	0.22	0.2727	0.80	0.0	8	32	40
2	0.22	0.2727	0.80	0.0	5	27	32
3	0.24	0.2879	0.75	0.0	3	10	13
4	0.24	0.2879	0.75	0.0	0	1	1

[12]:
```
# Average number of bikes rented hourly on a working and a non-workingday
display (train_data.pivot_table(values = 'cnt', index = "hr", columns =␣
 ↪'workingday', aggfunc = 'mean'))

# rows of the table == index = "hr"
# col of the table == columns = categories of workingday
# data of the table is aggfunc applied to the col mentioned in the values =␣
 ↪'cnt' == cnt.mean()
```

workingday	0	1
hr		
0	79.732558	32.461126
1	60.482558	15.051075
2	47.429412	8.005479
3	23.441176	4.525714
4	7.213018	4.707736
5	7.597561	20.726542
6	16.923977	89.077540
7	38.445087	252.385027
8	92.427746	418.442359
9	149.878613	214.691689
10	226.815029	118.420912
11	279.647399	138.198391
12	326.450867	174.377005
13	334.815029	173.582888
14	329.265896	159.735294
15	326.664740	173.136364
16	322.017341	252.352000
17	296.069364	461.733333
18	260.744186	434.860963
19	214.959302	307.251337
20	161.081395	220.505348
21	129.313953	166.556150
22	106.267442	123.604278
23	77.825581	77.278075

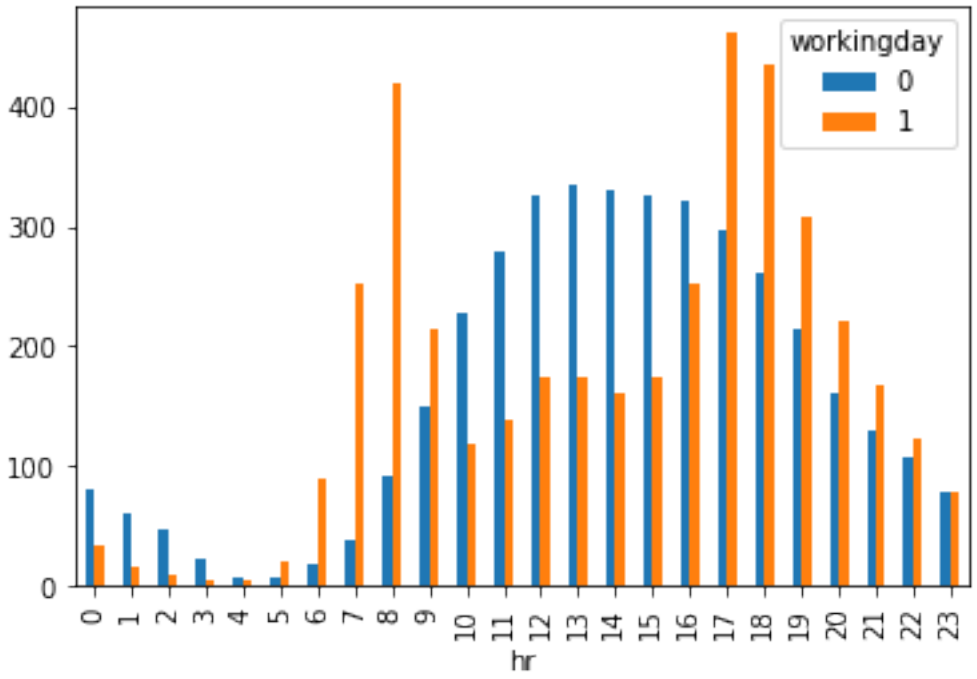
[15]:
```
temp_table = train_data.pivot_table(values = 'cnt', index = "hr", columns =␣
 ↪'workingday', aggfunc = 'mean')
temp_table.plot(kind = 'bar', stacked = True);
```



Shows us hourly distribution of count of rented bikes.

- In the hours 0-6, when people will be sleeping, we have low amount of rented bikes.
- Around 7th to 9th hour and 17th to 19th hour, we see a hike in the number of biked rented. This would be the hours when people go and come back from work on a working day.

[16]:
```
temp_table.plot(kind = 'bar');
```


workingday
0
1
400
300
200
100
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hr

[ ]: