Project: Capstone Project 1: Data Wrangling

Data wrangling steps:
Checking if there is any missing values(NaN),
From the code and result shown below, there is no missing values.

```
In [63]: hour.isnull().sum().sum()
Out[63]: 0

In [64]: day.isnull().sum().sum()
Out[64]: 0
```

The following data is the first 5 rows in hour.csv and day.csv respectively.

```
hour.head()
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

```
day.head()
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

We can start with wrangling with day.csv, the season, month, holiday and workingday columns are well organized.
Temperature is a common sense factor, and it is normalized by 41, so, we need to multiply the temp value by 41 for better data visualization.
The cnt column is better to categorize into range, that is, 0-500, 501-1000 … .... 5501-6000, and 6000+ .
As the data type of dteday is string, we need to convert the entire column to date type:

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt	cnt_range
1	2011-01-01 00:00:00	1	0	1	0	6	0	2	14.110847	0.363625	0.805833	0.160446	331	654	985	501-1000
2	2011-01-02 00:00:00	1	0	1	0	0	0	2	14.902598	0.353739	0.696087	0.248539	131	670	801	501-1000
3	2011-01-03 00:00:00	1	0	1	0	1	1	1	8.050924	0.189405	0.437273	0.248309	120	1229	1349	1001-1500
4	2011-01-04 00:00:00	1	0	1	0	2	1	1	8.200000	0.212122	0.590435	0.160296	108	1454	1562	1501-2000
5	2011-01-05 00:00:00	1	0	1	0	3	1	1	9.305237	0.229270	0.436957	0.186900	82	1518	1600	1501-2000

Now, we need to check if there is repeated dates or missing dates, the code below shows that there is no repeated dates or missing dates.

```
day.dteday.nunique()
```

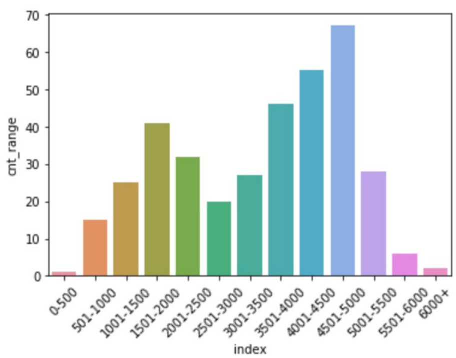
```
731
```

```
len(day)
```

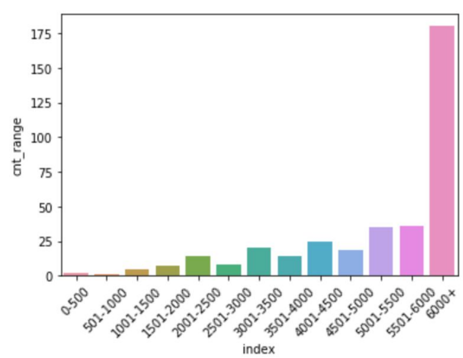
```
731
```

Check if there are any string type data in the DataFrame, the code below shows that all data is in int and float and timestamp type, from here, we now can check if there are outliers.

The following bar chart is the year of 2011 and 2012 respectively, it is clear that the 0-500 and 6000+ are outliers in 2011. But the usage is increasing exponentially in 2013. Therefore, we have to dive into year 2012 more detail.
It is obvious that the outliers are 0-500 and 6000+ in 2011; 0-500 and 501-1000 are outliers for 2012.

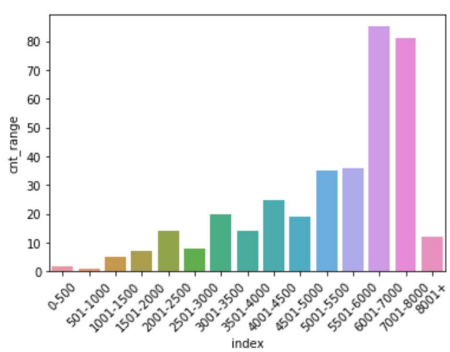


2011

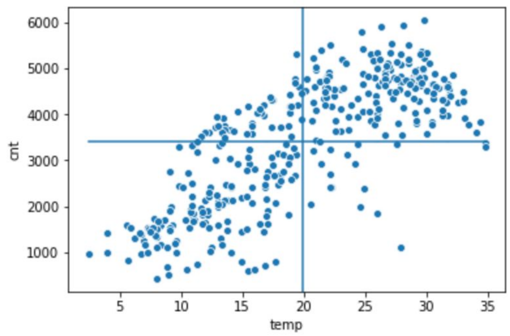


2012

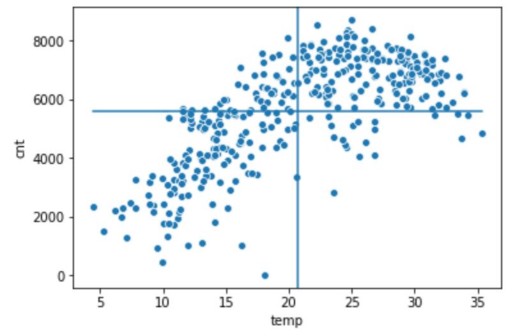
When expanding the 6000+ bar in 2012 into details:



2012

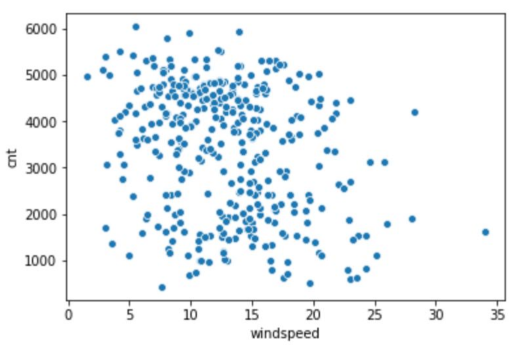


2011

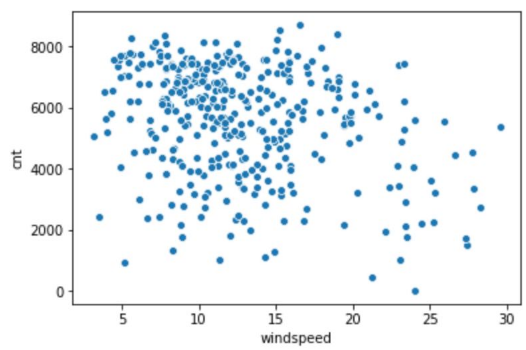


2012

Form the temperature perspective, when the temperature is lower than the average(the vertical line), the majority of the usage count is lower than the average(the horizontal line), when the temperature is higher than the average, the majority of the usage count is higher than the average. Therefore, the data points in the upper left and lower right are outliers.

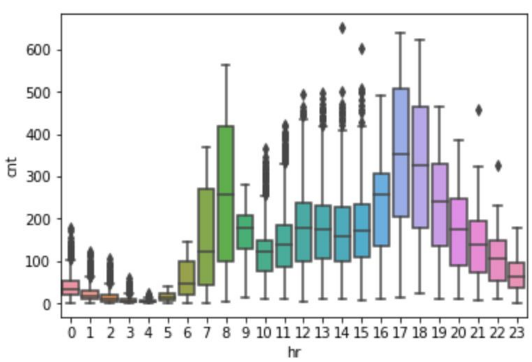


2011

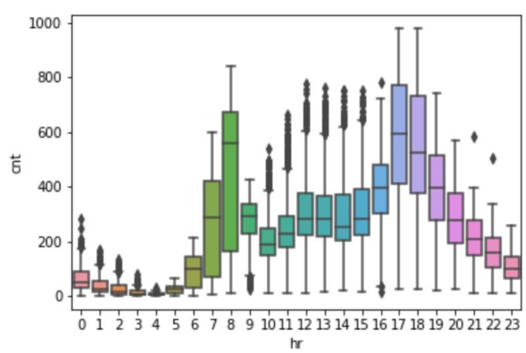


2012

From the wind speed perspective, high wind speed days are outliers.

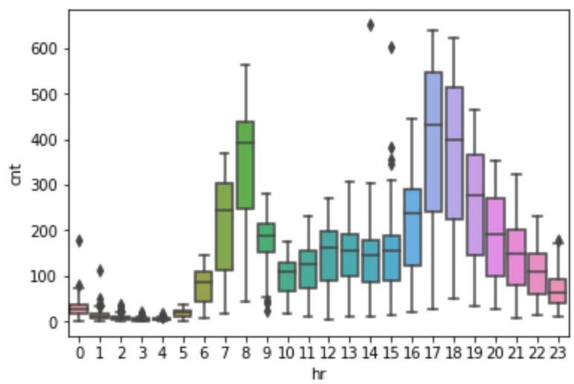


2011

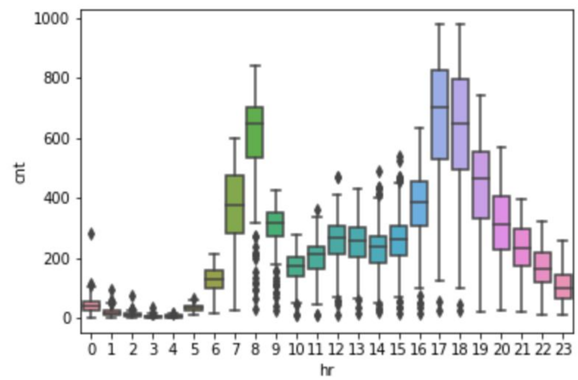


2012

From the hourly record data, the outliers are in between 12a.m to 4 a.m and from 10 a.m to 3 p.m



2011 working day



2012 working day

Now, it is clear that the outliers are mostly due to weekends or holidays.

We should do hypothesis testing for the outliers for calculating the demands. For example, when the weather will be bad(weather can be forecasted for weather channel), or when holidays and weekends are coming up.