```
# Bike-Sharing Demand Analysis.


#Q1) Load the data file.

import pandas as pd
df=pd.read_csv("C:/Users/shiva/Desktop/Shivani/SimpliLearn/Python/
hour.csv")
df
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
```

```
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
#Q2)Check for null values in the data and drop records with NAs.
```

```
df
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7
```

```
17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
df.columns
```

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

```
df["yr"]  # Yr column has 0 value
```

```
0        0
1        0
2        0
3        0
4        0
        ..
17374    1
17375    1
17376    1
17377    1
17378    1
Name: yr, Length: 17379, dtype: int64
```

```
df.dropna(subset=["yr"],axis=0,inplace=True)  # Dropping the missing
values of yr column
```

```
df.dropna(subset=["hr"],axis=0,inplace=True) # Dropping the missing
values of hr column
```

```
df.dropna(subset=["holiday"],axis=0,inplace=True) # Dropping the
missing values of holiday column

df.dropna(subset=["workingday"],axis=0,inplace=True)  # Dropping the
missing values of the workingday column

df.dropna(subset=["atemp"],axis=0,inplace=True)  # Dropping the
missing values of the atemp column

df.dropna(subset=["hum"],axis=0,inplace=True)  # Dropping the missing
values of the hum column

df.dropna(subset=["windspeed"],axis=0,inplace=True) # Dropping the
missing values of the windspeed column

df.dropna(subset=["casual"],axis=0,inplace=True)  # Dropping the
missing values of the casual column

df.dropna(subset=["registered"],axis=0,inplace=True) # Dropping the
missing values of the registered column

#Q3)--1--Sanity checks:

#Check if registered + casual = cnt for all the records. If not, the
row is junk and should be dropped.

#Month values should be 1-12 only

#Hour values should be 0-23

df
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5
```

```
3                0           1  0.24  0.2879  0.75     0.0000       3

4                0           1  0.24  0.2879  0.75     0.0000       0

...            ...         ...   ...     ...   ...        ...     ...

17374            1           2  0.26  0.2576  0.60     0.1642      11

17375            1           2  0.26  0.2576  0.60     0.1642       8

17376            1           1  0.26  0.2576  0.60     0.1642       7

17377            1           1  0.26  0.2727  0.56     0.1343      13

17378            1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]

df.dtypes

instant         int64
dteday         object
season          int64
yr              int64
mnth            int64
hr              int64
holiday         int64
weekday         int64
workingday      int64
weathersit      int64
temp          float64
atemp         float64
hum           float64
windspeed     float64
```

```
casual          int64
registered      int64
cnt             int64
dtype: object
```

```
df["registered"]+df["casual"]
```

```
0         16
1         40
2         32
3         13
4          1
        ...
17374    119
17375     89
17376     90
17377     61
17378     49
Length: 17379, dtype: int64
```

```
df["casual"].value_counts   # Count of casual columns
```

```
<bound method IndexOpsMixin.value_counts of 0         3
1         8
2         5
3         3
4         0
         ..
17374    11
17375     8
17376     7
17377    13
17378    12
Name: casual, Length: 17379, dtype: int64>
```

```
df["registered"]+df["casual"].value_counts()
```

```
0        1594.0
1        1114.0
2         825.0
3         707.0
4         562.0
          ...
17374       NaN
17375       NaN
17376       NaN
17377       NaN
17378       NaN
Length: 17379, dtype: float64
```

```
df["casual"].sum()
```

```
620017
```

```
a1=df["registered"].sum()
```

```
a1
```

```
2672662
```

```
a2=df["casual"].sum()
```

```
a2
```

```
620017
```

```
a3=a1+a2
```

```
a3
```

```
3292679
```

```
a4=df["cnt"].sum()
```

```
a4
```

```
3292679
```

```
# Here a3 is (registered+casual) =329267
# and a4 is (cnt) column summation=3292679

# Hence the values of both columns are equal and there are no junk
values in row

# Q3)-- 2--  Month values should be 1-12 only

df["mnth"]  # Analysis of Month Column. Month column only has values
from 1 to 12
```

```
0         1
1         1
2         1
3         1
4         1
         ..
17374    12
17375    12
17376    12
17377    12
17378    12
Name: mnth, Length: 17379, dtype: int64
```

```
df["mnth"].value_counts()
```

```
5     1488
7     1488
```

```
12    1483
8     1475
3     1473
10    1451
6     1440
4     1437
9     1437
11    1437
1     1429
2     1341
Name: mnth, dtype: int64
```

```
# Q3)-3- Hour  values should be 0-23 only
```

```
df
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13
```

```
17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
df["hr"]
```

```
0         0
1         1
2         2
3         3
4         4
         ..
17374    19
17375    20
17376    21
17377    22
17378    23
Name: hr, Length: 17379, dtype: int64
```

```
df["hr"].value_counts()  # Through Value_counts function hour value
from 0-23 is determined
```

```
17    730
16    730
13    729
15    729
14    729
12    728
22    728
21    728
20    728
19    728
18    728
23    728
11    727
10    727
```

```
9     727
8     727
7     727
0     726
6     725
1     724
5     717
2     715
4     697
3     697
Name: hr, dtype: int64
```

```
# Q4) The variables ‘casual’ and ‘registered’ are redundant and need
to be dropped.
#‘Instant’ is the index and needs to be dropped too.
#The date column dteday will not be used in the model building,
#and therefore needs to be dropped.
#Create a new dataframe named inp1.
```

```
df["casual"]
```

```
0         3
1         8
2         5
3         3
4         0
        ..
17374    11
17375     8
17376     7
17377    13
17378    12
Name: casual, Length: 17379, dtype: int64
```

```
df.drop(columns=["casual","registered","instant","dteday"])  #
Dropping the casual,registered,instant and dteday columns
```

```
       season  yr  mnth  hr  holiday  weekday  workingday  weathersit
temp  \
0           1   0     1   0        0        6           0           1
0.24
1           1   0     1   1        0        6           0           1
0.22
2           1   0     1   2        0        6           0           1
0.22
3           1   0     1   3        0        6           0           1
0.24
4           1   0     1   4        0        6           0           1
0.24
...       ...  ..   ...  ..      ...      ...         ...         ...
...
```

```
17374       1   1    12  19        0        1           1           2
0.26
17375       1   1    12  20        0        1           1           2
0.26
17376       1   1    12  21        0        1           1           1
0.26
17377       1   1    12  22        0        1           1           1
0.26
17378       1   1    12  23        0        1           1           1
0.26

        atemp   hum  windspeed  cnt
0      0.2879  0.81     0.0000   16
1      0.2727  0.80     0.0000   40
2      0.2727  0.80     0.0000   32
3      0.2879  0.75     0.0000   13
4      0.2879  0.75     0.0000    1
...       ...   ...        ...  ...
17374  0.2576  0.60     0.1642  119
17375  0.2576  0.60     0.1642   89
17376  0.2576  0.60     0.1642   90
17377  0.2727  0.56     0.1343   61
17378  0.2727  0.65     0.1343   49

[17379 rows x 13 columns]
```

```
inp1=df.drop(columns=["casual","registered","instant","dteday"])
```

```
inp1  # The new data frame with dropped columns
```

```
       season  yr  mnth  hr  holiday  weekday  workingday  weathersit
temp  \
0           1   0     1   0        0        6           0           1
0.24
1           1   0     1   1        0        6           0           1
0.22
2           1   0     1   2        0        6           0           1
0.22
3           1   0     1   3        0        6           0           1
0.24
4           1   0     1   4        0        6           0           1
0.24
...       ...  ..   ...  ..      ...      ...         ...         ...
...
17374       1   1    12  19        0        1           1           2
0.26
17375       1   1    12  20        0        1           1           2
0.26
17376       1   1    12  21        0        1           1           1
0.26
17377       1   1    12  22        0        1           1           1
```

```
0.26
17378       1   1    12  23        0        1           1           1
0.26

        atemp   hum  windspeed  cnt
0      0.2879  0.81     0.0000   16
1      0.2727  0.80     0.0000   40
2      0.2727  0.80     0.0000   32
3      0.2879  0.75     0.0000   13
4      0.2879  0.75     0.0000    1
...       ...   ...        ...  ...
17374  0.2576  0.60     0.1642  119
17375  0.2576  0.60     0.1642   89
17376  0.2576  0.60     0.1642   90
17377  0.2727  0.56     0.1343   61
17378  0.2727  0.65     0.1343   49

[17379 rows x 13 columns]
```

#Q5) A Univariate analysis:
 # 5-1 Describe the numerical fields in the dataset using pandas describe method.

```
df

       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0
```

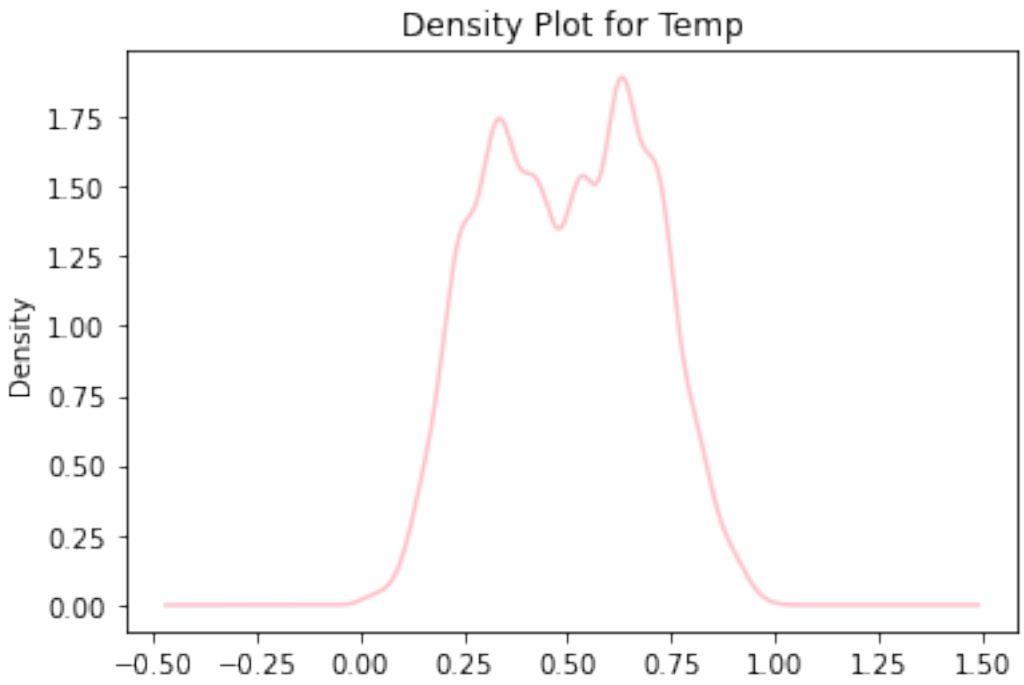
```
...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12
```

```
       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
df.dtypes
```

```
instant         int64
dteday         object
season          int64
yr              int64
mnth            int64
hr              int64
holiday         int64
weekday         int64
workingday      int64
weathersit      int64
temp          float64
atemp         float64
hum           float64
windspeed     float64
casual          int64
registered      int64
cnt             int64
dtype: object
```

```
# Hence the numerical fields are
instant,season,yr,mnth,hr,holiday,weekday,workingdat,
#weathersit,temp,atemp,hum,windspeed,casual,registered and cnt

# Q5) b) -Make density plot for temp. This would give a sense of the
centrality and the spread of the distribution.

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

df.temp

0        0.24
1        0.22
2        0.22
3        0.24
4        0.24
        ...
17374    0.26
17375    0.26
17376    0.26
17377    0.26
17378    0.26
Name: temp, Length: 17379, dtype: float64

# Density plot for temp attribute
df.temp.plot.density(color="pink")
plt.title("Density Plot for Temp")

Text(0.5, 1.0, 'Density Plot for Temp')
```




```
#Q5) c)  Boxplot for atemp . Are there any outliers?
```

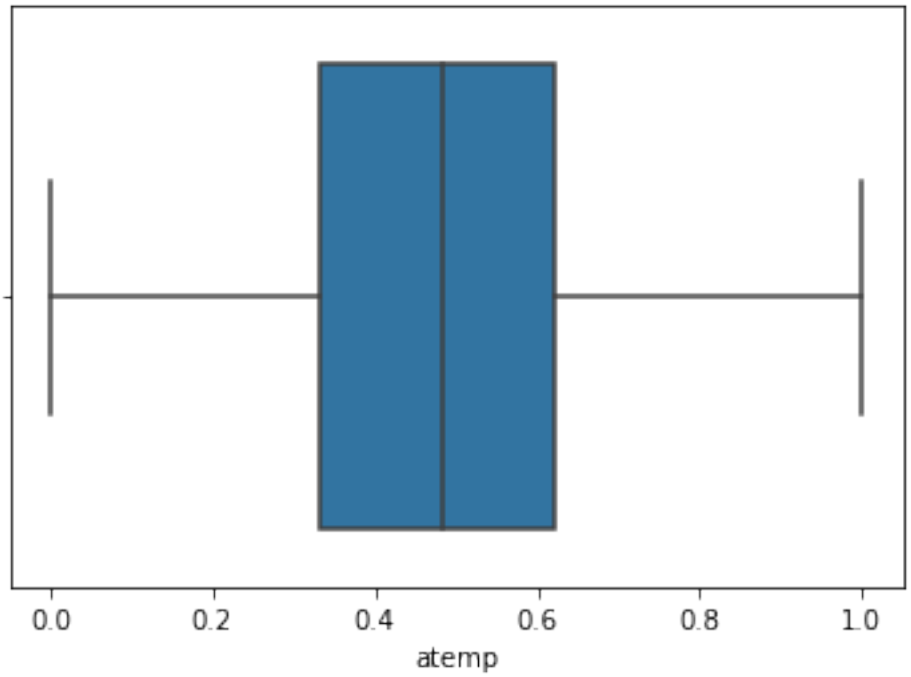
```
import pandas as pd
import matplotlib.pyplot as plt


import numpy as np
import seaborn as sns
```

```
df["atemp"]  # Column atemp of data frame
```

```
0        0.2879
1        0.2727
2        0.2727
3        0.2879
4        0.2879
         ...
17374    0.2576
17375    0.2576
17376    0.2576
17377    0.2727
17378    0.2727
Name: atemp, Length: 17379, dtype: float64
```

```
sns.boxplot(x="atemp",data=df)
```

```
<AxesSubplot:xlabel='atemp'>
```




```
df.describe()[["atemp"]]  # Statistical Analysis of Atemp Column
```

```
              atemp
count  17379.000000
mean       0.475775
std        0.171850
min        0.000000
25%        0.333300
50%        0.484800
75%        0.621200
max        1.000000
```

Here the maximum value of atemp column =1 and mean value is 0.475. The mean value of atemp is not sensitive to max indicating that max value is not an outlier

```
# Outlier Analysis
# Finding the Outlier through statistical analysis
```

```
df  # Data frame
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
```

```
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```python
# Finding outliers through statistical methods
def find_outliers_IQR(df):
```

```
    q1=df.quantile(0.25)

    q3=df.quantile(0.75)

    IQR=q3-q1

    outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]   # Formula
of Outlier

    return outliers

outliers = find_outliers_IQR(df["atemp"])

print("number of outliers: "+ str(len(outliers)))

print("max outlier value: "+ str(outliers.max()))

print("min outlier value: "+ str(outliers.min()))


number of outliers: 0
max outlier value: nan
min outlier value: nan

# Q5) D- Histogram for hum

#Do you detect any abnormally high values?

df

       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5
```

```
3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
df.columns
```

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

```
df["hum"]
```

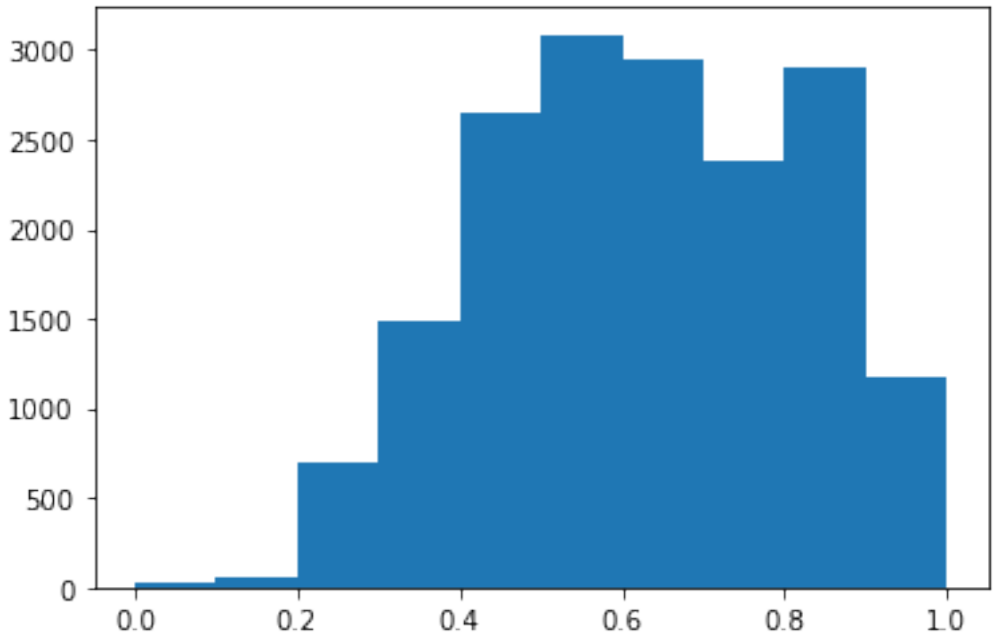
```
0        0.81
1        0.80
2        0.80
3        0.75
4        0.75
         ...
```

```
17374    0.60
17375    0.60
17376    0.60
17377    0.56
17378    0.65
Name: hum, Length: 17379, dtype: float64
```

```
x=df["hum"]
```

```
plt.hist(x)  # Normal Histogram of Humidity column
```

```
(array([  23.,   55.,  696., 1481., 2641., 3084., 2940., 2384., 2900.,
        1175.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```




```
# The concept of Binning can also be applied for humidity column as it
is a numerical column.
# Here the humidity column is further divided into low,medium and high
humidity for further analysis
```

```
bins=np.linspace(min(df["hum"]),max(df["hum"]),4)
```

```
group_names=["Low Humidity","Medium Humidity","High Humidity"]
```

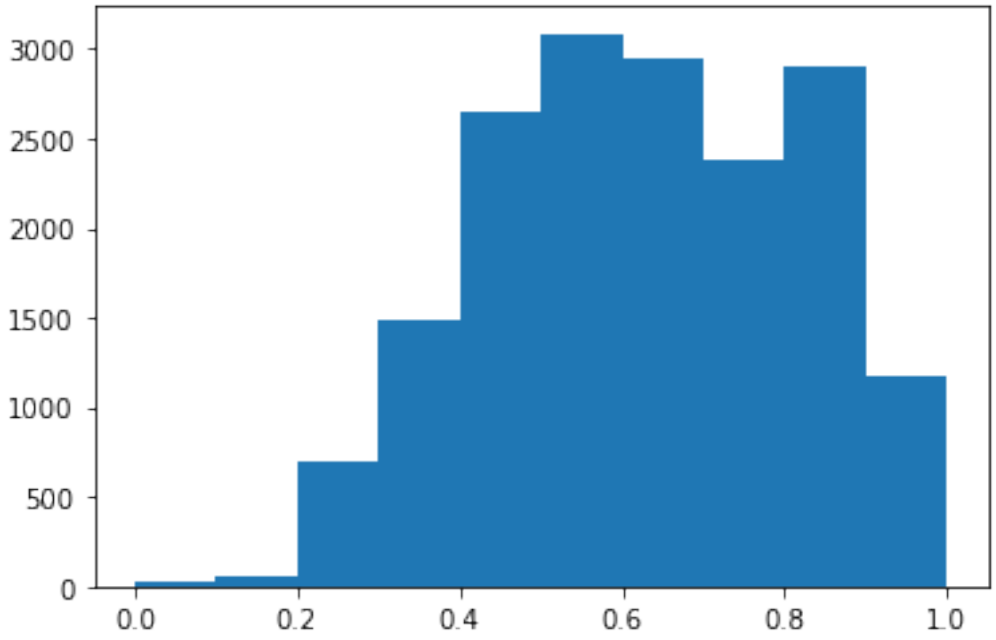
```
df["Humidity-
Analysis"]=pd.cut(df["hum"],bins,labels=group_names,include_lowest=Tru
e)
```

```
x=df["Humidity-Analysis"]
```

```
0          High Humidity
1          High Humidity
2          High Humidity
3          High Humidity
4          High Humidity
              ...
17374    Medium Humidity
17375    Medium Humidity
17376    Medium Humidity
17377    Medium Humidity
17378    Medium Humidity
Name: Humidity-Analysis, Length: 17379, dtype: category
Categories (3, object): ['Low Humidity' < 'Medium Humidity' < 'High
Humidity']
```

```
plt.hist(x)
```

```
(array([  23.,   55.,  696., 1481., 2641., 3084., 2940., 2384., 2900.,
        1175.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```




```
# From the Above Mentioned Histogram, it is clear that there is a
moderate to high humidity.
# The high humid values are in the range of 0.53 to 0.7
# Low Humid values are in the range of 0.0 to 0.2
# From 0.22 to 0.49 there is low moderate humid
# There is no such abonormal high values but humid conditions are
mostly moderate and high not low

# Q5- c)Density plot for windspeed
```

```
df
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt Humidity-Analysis
0              13   16     High Humidity
1              32   40     High Humidity
2              27   32     High Humidity
3              10   13     High Humidity
4               1    1     High Humidity
...           ...  ...               ...
17374         108  119   Medium Humidity
17375          81   89   Medium Humidity
```

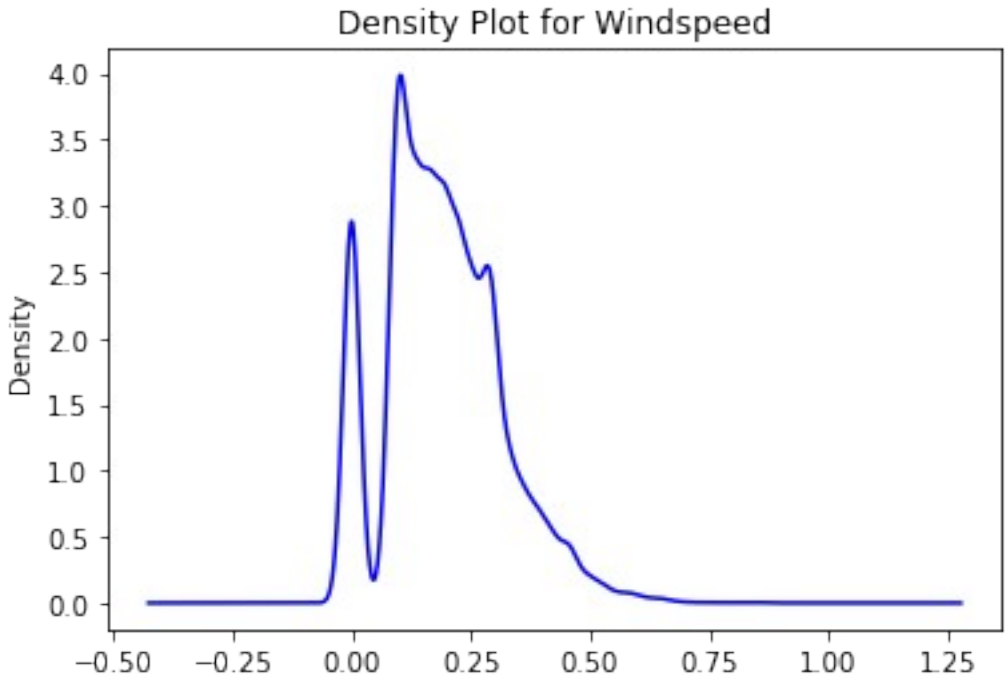
```
17376          83   90   Medium Humidity
17377          48   61   Medium Humidity
17378          37   49   Medium Humidity

[17379 rows x 18 columns]
```

```
df["windspeed"]  # Data Frame of Windspeed column
```

```
0        0.0000
1        0.0000
2        0.0000
3        0.0000
4        0.0000
          ...
17374    0.1642
17375    0.1642
17376    0.1642
17377    0.1343
17378    0.1343
Name: windspeed, Length: 17379, dtype: float64
```

```
df.windspeed.plot.density(color="blue")
plt.title("Density Plot for Windspeed")
```

```
Text(0.5, 1.0, 'Density Plot for Windspeed')
```




```
# From this density plot we can figure out that the windspeed between
#0.12 to 0.25  are most common for higher windspeed
```

```
# Q5 c) Box and density plot for cnt – this is the variable of
interest

#Do you see any outliers in the boxplot?

#Does the density plot provide a similar insight?

df  # The data frame
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt Humidity-Analysis
0              13   16     High Humidity
```

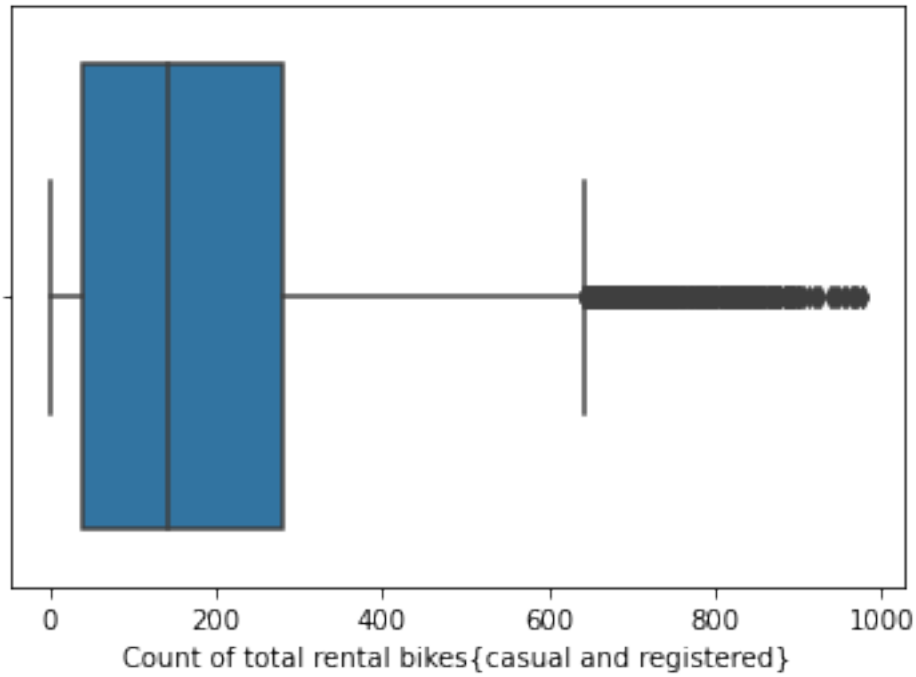
```
1              32   40     High Humidity
2              27   32     High Humidity
3              10   13     High Humidity
4               1    1     High Humidity
...           ...  ...               ...
17374         108  119   Medium Humidity
17375          81   89   Medium Humidity
17376          83   90   Medium Humidity
17377          48   61   Medium Humidity
17378          37   49   Medium Humidity

[17379 rows x 18 columns]
```

```
df["cnt"] # Data frame of  total rental bikes including both casual
and registered{count values}
```

```
0         16
1         40
2         32
3         13
4          1
        ...
17374    119
17375     89
17376     90
17377     61
17378     49
Name: cnt, Length: 17379, dtype: int64
```

```
# Box Plot of total rental bikes which includes both casual and
registered
import seaborn as sns
x=df["cnt"]
sns.boxplot(x=df["cnt"])
plt.xlabel("Count of total rental bikes{casual and registered}")
```

```
Text(0.5, 0, 'Count of total rental bikes{casual and registered}')
```




```
# Here I will pandas.describe() to find outliers
```

```
df["cnt"].describe()
```

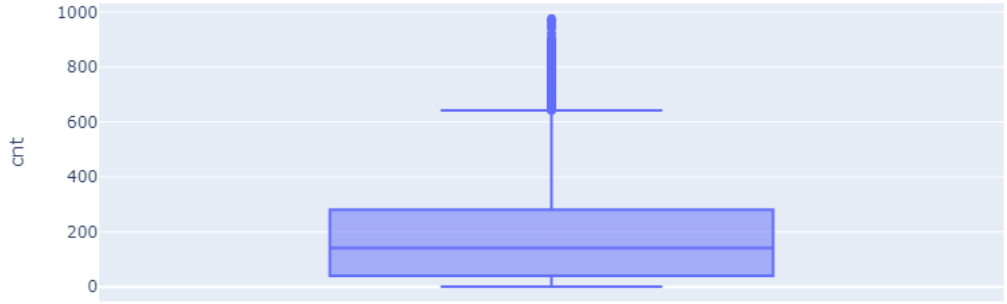
```
count    17379.000000
mean       189.463088
std        181.387599
min          1.000000
25%         40.000000
50%        142.000000
75%        281.000000
max        977.000000
Name: cnt, dtype: float64
```

```
# Cnt column is having the outlier. By Applying describe() to cnt
column, it is clear that
# maximum value of cnt is 977 and mean of cnt column is 189.46. The
mean is sensitive to the
# outliers,here the mean is so small to the max value indicating that
this cnt column have outlier
```

```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\shiva\anaconda3\lib\
site-packages (5.13.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\shiva\
anaconda3\lib\site-packages (from plotly) (8.2.1)
Note: you may need to restart the kernel to use updated packages.
```

```
# Finding the outlier through Box Plot Method
import numpy as np
import plotly.express as px

y=df["cnt"]

#create a box plot

fig = px.box(df, y="cnt")

fig.show()
```




```
# From the above mentioned box plot,it is clear that there are lots of
outliers. The thick line
# near 0 is the box part of our box plot. Above th box and the upper
fence are some points
# showing outliers. The box points can be viewed through hovering on
the charts

#Q5) c-Does the density plot provide a similar insight?


df  # The data frame
```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	\
0	1	01-01-2011	1	0	1	0	0	6	
1	2	01-01-2011	1	0	1	1	0	6	
2	3	01-01-2011	1	0	1	2	0	6	
3	4	01-01-2011	1	0	1	3	0	6	
4	5	01-01-2011	1	0	1	4	0	6	
...	...	...	...	..	...	..	...	...	
17374	17375	31-12-2012	1	1	12	19	0	1	
17375	17376	31-12-2012	1	1	12	20	0	1	
17376	17377	31-12-2012	1	1	12	21	0	1	
17377	17378	31-12-2012	1	1	12	22	0	1	
17378	17379	31-12-2012	1	1	12	23	0	1	

```
       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt Humidity-Analysis
0              13   16     High Humidity
1              32   40     High Humidity
2              27   32     High Humidity
3              10   13     High Humidity
4               1    1     High Humidity
...           ...  ...               ...
17374         108  119   Medium Humidity
17375          81   89   Medium Humidity
17376          83   90   Medium Humidity
17377          48   61   Medium Humidity
17378          37   49   Medium Humidity

[17379 rows x 18 columns]
```

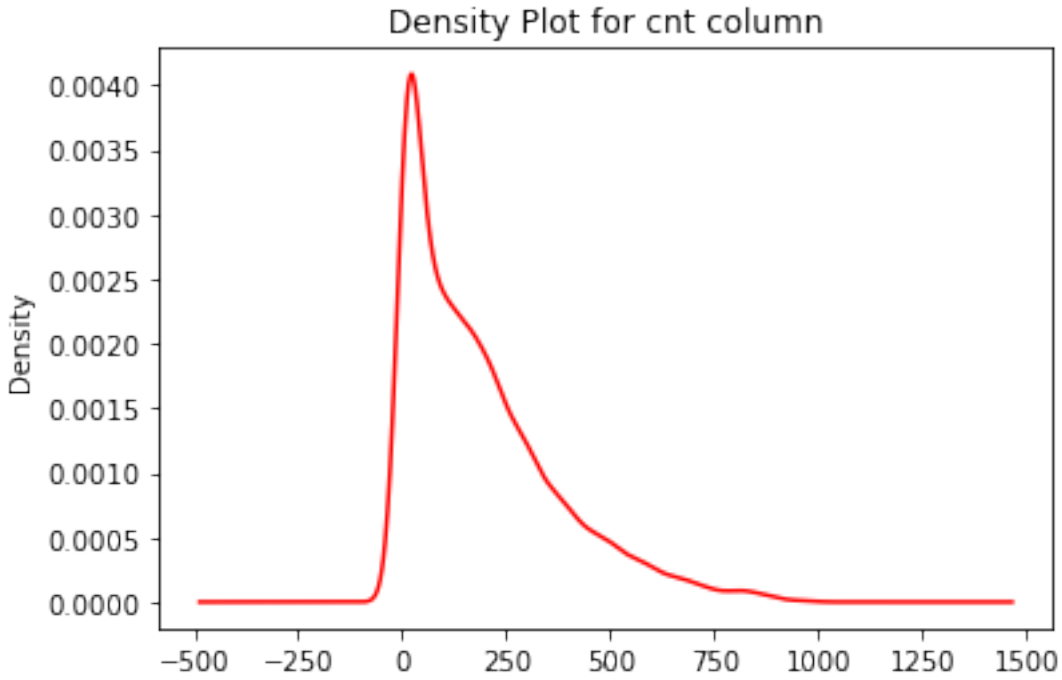
```
df["cnt"]  # Data frame of cnt column
```

```
0         16
1         40
2         32
3         13
4          1
        ...
```

```
17374    119
17375     89
17376     90
17377     61
17378     49
Name: cnt, Length: 17379, dtype: int64
```

```
df.cnt.plot.density(color="red")
plt.title("Density Plot for cnt column")
```

```
Text(0.5, 1.0, 'Density Plot for cnt column')
```

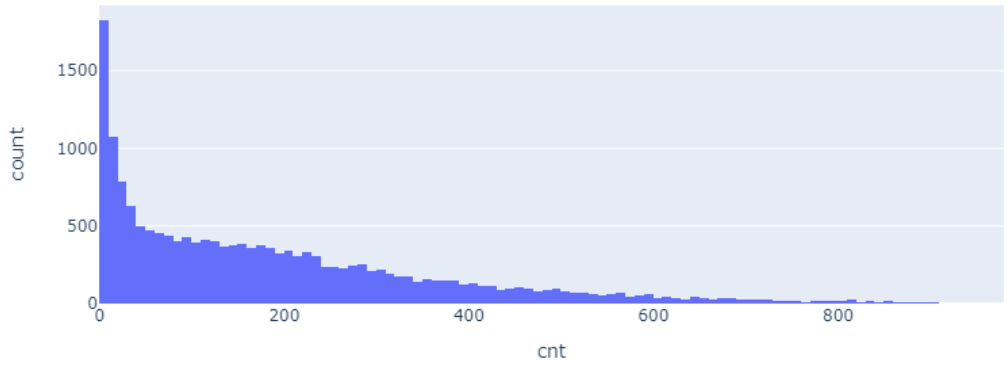



```
# From the density plot it is clear that most of the count of the
total rental
# bikes are in the range of 60 to 150
# This density plot is different from the box plot

# Q6) Outlier treatment:

# Q6) 1--Cnt looks like some hours have rather high values.
#You'll need to treat these outliers
#so that they don't skew the analysis and the model.

df["cnt"]  # data frame of total number of riders including casual and
registered
```

```
0         16
1         40
2         32
3         13
```

```
4          1
         ...
17374    119
17375     89
17376     90
17377     61
17378     49
Name: cnt, Length: 17379, dtype: int64
```

```
# Finding outliers and viewing the data through histogram

import numpy as np
import plotly.express as px

x=df["cnt"]  # Count of bike riders including registered and casual

px.histogram(df,x)
```




```
# Q6) 1--  Find out the following percentiles: 10, 25, 50, 75, 90, 95, 99

df  # The data frame
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
```

```
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt Humidity-Analysis
0              13   16     High Humidity
1              32   40     High Humidity
2              27   32     High Humidity
3              10   13     High Humidity
4               1    1     High Humidity
...           ...  ...               ...
17374         108  119   Medium Humidity
17375          81   89   Medium Humidity
17376          83   90   Medium Humidity
17377          48   61   Medium Humidity
17378          37   49   Medium Humidity

[17379 rows x 18 columns]
```

```python
# Q6) 1-- Percentile Analysis through Function
def outlieranalysis(df):
    q1=df.quantile(0.10)  # 10 percentile
    print("The 10 percentile is:",q1)
    print("\n\n")
    q2=df.quantile(0.25)  # 10 percentile
    print("The 25 percentile is:",q2)
    print("\n")
    q3=df.quantile(0.50)  # 10 percentile
    print("The 50 percentile is:",q3)
```

```
    print("\n")
    q4=df.quantile(0.75)  # 10 percentile
    print("The 75 percentile is:",q4)
    q5=df.quantile(0.90)  # 10 percentile
    print("The 90 percentile is:",q5)
    q6=df.quantile(0.95)  # 10 percentile
    print("The 90 percentile is:",q6)
    q7=df.quantile(0.99)  # 10 percentile
    print("The 99 percentile is:",q7)

    percentile_list=[q1,q2,q3,q4,q5,q6,q7]
    print("The percentile list is ",percentile_list)
```

```
outlieranalysis(df)
```

```
The 10 percentile is: instant       1738.8000
season            1.0000
yr                0.0000
mnth              2.0000
hr                2.0000
holiday           0.0000
weekday           0.0000
workingday        0.0000
weathersit        1.0000
temp              0.2400
atemp             0.2424
hum               0.3700
windspeed         0.0000
casual            1.0000
registered        7.0000
cnt               9.0000
Name: 0.1, dtype: float64
```

```
The 25 percentile is: instant       4345.5000
season            2.0000
yr                0.0000
mnth              4.0000
hr                6.0000
holiday           0.0000
weekday           1.0000
workingday        0.0000
weathersit        1.0000
temp              0.3400
```

```
atemp            0.3333
hum              0.4800
windspeed        0.1045
casual           4.0000
registered      34.0000
cnt             40.0000
Name: 0.25, dtype: float64


The 50 percentile is: instant       8690.0000
season           3.0000
yr               1.0000
mnth             7.0000
hr              12.0000
holiday          0.0000
weekday          3.0000
workingday       1.0000
weathersit       1.0000
temp             0.5000
atemp            0.4848
hum              0.6300
windspeed        0.1940
casual          17.0000
registered     115.0000
cnt            142.0000
Name: 0.5, dtype: float64


The 75 percentile is: instant       13034.5000
season            3.0000
yr                1.0000
mnth             10.0000
hr               18.0000
holiday           0.0000
weekday           5.0000
workingday        1.0000
weathersit        2.0000
temp              0.6600
atemp             0.6212
hum               0.7800
windspeed         0.2537
casual           48.0000
registered      220.0000
cnt             281.0000
Name: 0.75, dtype: float64
The 90 percentile is: instant       15641.2000
season            4.0000
yr                1.0000
mnth             11.0000
hr               21.0000
```

```
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        2.0000
temp              0.7400
atemp             0.6970
hum               0.8800
windspeed         0.3582
casual           92.0000
registered      354.0000
cnt             451.2000
Name: 0.9, dtype: float64
The 90 percentile is: instant       16510.1000
season            4.0000
yr                1.0000
mnth             12.0000
hr               22.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8000
atemp             0.7424
hum               0.9300
windspeed         0.4179
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64
The 99 percentile is: instant       17205.2200
season            4.0000
yr                1.0000
mnth             12.0000
hr               23.0000
holiday           1.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8844
atemp             0.8182
hum               1.0000
windspeed         0.5224
casual          240.0000
registered      699.2200
cnt             782.2200
Name: 0.99, dtype: float64
The percentile list is  [instant       1738.8000
season           1.0000
yr               0.0000
mnth             2.0000
```

```
hr              2.0000
holiday         0.0000
weekday         0.0000
workingday      0.0000
weathersit      1.0000
temp            0.2400
atemp           0.2424
hum             0.3700
windspeed       0.0000
casual          1.0000
registered      7.0000
cnt             9.0000
Name: 0.1, dtype: float64, instant       4345.5000
season          2.0000
yr              0.0000
mnth            4.0000
hr              6.0000
holiday         0.0000
weekday         1.0000
workingday      0.0000
weathersit      1.0000
temp            0.3400
atemp           0.3333
hum             0.4800
windspeed       0.1045
casual          4.0000
registered     34.0000
cnt            40.0000
Name: 0.25, dtype: float64, instant       8690.0000
season          3.0000
yr              1.0000
mnth            7.0000
hr             12.0000
holiday         0.0000
weekday         3.0000
workingday      1.0000
weathersit      1.0000
temp            0.5000
atemp           0.4848
hum             0.6300
windspeed       0.1940
casual         17.0000
registered    115.0000
cnt           142.0000
Name: 0.5, dtype: float64, instant       13034.5000
season           3.0000
yr               1.0000
mnth            10.0000
hr              18.0000
holiday          0.0000
```

```
weekday           5.0000
workingday        1.0000
weathersit        2.0000
temp              0.6600
atemp             0.6212
hum               0.7800
windspeed         0.2537
casual           48.0000
registered      220.0000
cnt             281.0000
Name: 0.75, dtype: float64, instant       15641.2000
season            4.0000
yr                1.0000
mnth             11.0000
hr               21.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        2.0000
temp              0.7400
atemp             0.6970
hum               0.8800
windspeed         0.3582
casual           92.0000
registered      354.0000
cnt             451.2000
Name: 0.9, dtype: float64, instant       16510.1000
season            4.0000
yr                1.0000
mnth             12.0000
hr               22.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8000
atemp             0.7424
hum               0.9300
windspeed         0.4179
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64, instant       17205.2200
season            4.0000
yr                1.0000
mnth             12.0000
hr               23.0000
holiday           1.0000
weekday           6.0000
workingday        1.0000
```

```
weathersit        3.0000
temp              0.8844
atemp             0.8182
hum               1.0000
windspeed         0.5224
casual          240.0000
registered      699.2200
cnt             782.2200
Name: 0.99, dtype: float64]
```

```
# Q6) 2--  Decide the cutoff percentile and drop records with values
higher than the cutoff.
#Name the new dataframe as inp2.
```

```
listp=[10,25,50,75,90,95,99]
print(f"The given percentile list is {listp} ")
```

```
The given percentile list is [10, 25, 50, 75, 90, 95, 99]
```

```
c=10+25+50+75+90+95+99
print(c)  # Addition of percentile
```

```
444
```

```
p=7
print(p)  # Number of values
```

```
7
```

```
cut_of_percentile=c/p
print(cut_of_percentile)
```

```
63.42857142857143
```

```
# Hence from the given list the cut of percentile is 63.4
# According to scenario we have to drop the values which are higher
than cut off
```

```
df  # The data frame
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1
```

```
       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt Humidity-Analysis
0              13   16     High Humidity
1              32   40     High Humidity
2              27   32     High Humidity
3              10   13     High Humidity
4               1    1     High Humidity
...           ...  ...               ...
17374         108  119   Medium Humidity
17375          81   89   Medium Humidity
17376          83   90   Medium Humidity
17377          48   61   Medium Humidity
17378          37   49   Medium Humidity

[17379 rows x 18 columns]
```

```python
def dropping_outliers(df):
    q4=df.quantile(0.75)  # 10 percentile
    print("The 75 percentile is:",q4)
    q5=df.quantile(0.90)  # 10 percentile
    print("The 90 percentile is:",q5)
    q6=df.quantile(0.95)  # 10 percentile
    print("The 90 percentile is:",q6)
    q7=df.quantile(0.99)  # 10 percentile
    print("The 99 percentile is:",q7)
```

```
dropping_outliers(df)
```

```
The 75 percentile is: instant       13034.5000
season             3.0000
yr                 1.0000
mnth              10.0000
hr                18.0000
holiday            0.0000
weekday            5.0000
workingday         1.0000
weathersit         2.0000
temp               0.6600
atemp              0.6212
hum                0.7800
windspeed          0.2537
casual            48.0000
registered       220.0000
cnt              281.0000
Name: 0.75, dtype: float64
The 90 percentile is: instant       15641.2000
season             4.0000
yr                 1.0000
mnth              11.0000
hr                21.0000
holiday            0.0000
weekday            6.0000
workingday         1.0000
weathersit         2.0000
temp               0.7400
atemp              0.6970
hum                0.8800
windspeed          0.3582
casual            92.0000
registered       354.0000
cnt              451.2000
Name: 0.9, dtype: float64
The 90 percentile is: instant       16510.1000
season             4.0000
yr                 1.0000
mnth              12.0000
hr                22.0000
holiday            0.0000
weekday            6.0000
workingday         1.0000
weathersit         3.0000
temp               0.8000
atemp              0.7424
hum                0.9300
windspeed          0.4179
```

```
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64
The 99 percentile is: instant       17205.2200
season            4.0000
yr                1.0000
mnth             12.0000
hr               23.0000
holiday           1.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8844
atemp             0.8182
hum               1.0000
windspeed         0.5224
casual          240.0000
registered      699.2200
cnt             782.2200
Name: 0.99, dtype: float64
```

```
# Dropping the quantiles
q4=df.quantile(0.75)
print(q4)
q5=df.quantile(0.90)
print(q5)
q6=df.quantile(0.95)
print(q6)
q7=df.quantile(0.99)
print(q7)


list2=[q4,q5,q6,q7]

print(list2)
```

```
instant       13034.5000
season            3.0000
yr                1.0000
mnth             10.0000
hr               18.0000
holiday           0.0000
weekday           5.0000
workingday        1.0000
weathersit        2.0000
temp              0.6600
atemp             0.6212
hum               0.7800
windspeed         0.2537
```

```
casual           48.0000
registered      220.0000
cnt             281.0000
Name: 0.75, dtype: float64
instant       15641.2000
season            4.0000
yr                1.0000
mnth             11.0000
hr               21.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        2.0000
temp              0.7400
atemp             0.6970
hum               0.8800
windspeed         0.3582
casual           92.0000
registered      354.0000
cnt             451.2000
Name: 0.9, dtype: float64
instant       16510.1000
season            4.0000
yr                1.0000
mnth             12.0000
hr               22.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8000
atemp             0.7424
hum               0.9300
windspeed         0.4179
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64
instant       17205.2200
season            4.0000
yr                1.0000
mnth             12.0000
hr               23.0000
holiday           1.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8844
atemp             0.8182
hum               1.0000
```

```
windspeed         0.5224
casual          240.0000
registered      699.2200
cnt             782.2200
Name: 0.99, dtype: float64
[instant       13034.5000
season             3.0000
yr                 1.0000
mnth              10.0000
hr                18.0000
holiday            0.0000
weekday            5.0000
workingday         1.0000
weathersit         2.0000
temp               0.6600
atemp              0.6212
hum                0.7800
windspeed          0.2537
casual            48.0000
registered       220.0000
cnt              281.0000
Name: 0.75, dtype: float64, instant       15641.2000
season             4.0000
yr                 1.0000
mnth              11.0000
hr                21.0000
holiday            0.0000
weekday            6.0000
workingday         1.0000
weathersit         2.0000
temp               0.7400
atemp              0.6970
hum                0.8800
windspeed          0.3582
casual            92.0000
registered       354.0000
cnt              451.2000
Name: 0.9, dtype: float64, instant       16510.1000
season             4.0000
yr                 1.0000
mnth              12.0000
hr                22.0000
holiday            0.0000
weekday            6.0000
workingday         1.0000
weathersit         3.0000
temp               0.8000
atemp              0.7424
hum                0.9300
windspeed          0.4179
```

```
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64, instant       17205.2200
season             4.0000
yr                 1.0000
mnth              12.0000
hr                23.0000
holiday            1.0000
weekday            6.0000
workingday         1.0000
weathersit         3.0000
temp               0.8844
atemp              0.8182
hum                1.0000
windspeed          0.5224
casual           240.0000
registered       699.2200
cnt              782.2200
Name: 0.99, dtype: float64]
```

```
# Removing the quantiles through list
list2=[q4,q5,q6,q7]
print(list2)
list2.remove(q4)   # Removing the 0.75 quantile
list2.remove(q5)  # Removing the 0.90 quantile
list2.remove(q6)   # Removing the 0.95 quantile
list2.remove(q7)   # Removing the 0.99 quantile
```

```
[instant       13034.5000
season             3.0000
yr                 1.0000
mnth              10.0000
hr                18.0000
holiday            0.0000
weekday            5.0000
workingday         1.0000
weathersit         2.0000
temp               0.6600
atemp              0.6212
hum                0.7800
windspeed          0.2537
casual            48.0000
registered       220.0000
cnt              281.0000
Name: 0.75, dtype: float64, instant       15641.2000
season             4.0000
yr                 1.0000
mnth              11.0000
hr                21.0000
holiday            0.0000
```

```
weekday           6.0000
workingday        1.0000
weathersit        2.0000
temp              0.7400
atemp             0.6970
hum               0.8800
windspeed         0.3582
casual           92.0000
registered      354.0000
cnt             451.2000
Name: 0.9, dtype: float64, instant       16510.1000
season            4.0000
yr                1.0000
mnth             12.0000
hr               22.0000
holiday           0.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8000
atemp             0.7424
hum               0.9300
windspeed         0.4179
casual          138.1000
registered      465.0000
cnt             563.1000
Name: 0.95, dtype: float64, instant       17205.2200
season            4.0000
yr                1.0000
mnth             12.0000
hr               23.0000
holiday           1.0000
weekday           6.0000
workingday        1.0000
weathersit        3.0000
temp              0.8844
atemp             0.8182
hum               1.0000
windspeed         0.5224
casual          240.0000
registered      699.2200
cnt             782.2200
Name: 0.99, dtype: float64]
```

```
# Creation of data new data frame  inp2

import pandas as pd

df
q1=df.quantile(0.10)  # 10 percentile
print(q1)
```

```
q2=df.quantile(0.25)  # 25 percentile
print(q2)
q3=df.quantile(0.50)  # 50 percentile
print(q3)
thenewdata=[q1,q2,q3]
inp2 = pd.DataFrame(thenewdata, columns=['cnt'])
inp2  # The new data frame
```

```
instant       1738.8000
season           1.0000
yr               0.0000
mnth             2.0000
hr               2.0000
holiday          0.0000
weekday          0.0000
workingday       0.0000
weathersit       1.0000
temp             0.2400
atemp            0.2424
hum              0.3700
windspeed        0.0000
casual           1.0000
registered       7.0000
cnt              9.0000
Name: 0.1, dtype: float64
instant       4345.5000
season           2.0000
yr               0.0000
mnth             4.0000
hr               6.0000
holiday          0.0000
weekday          1.0000
workingday       0.0000
weathersit       1.0000
temp             0.3400
atemp            0.3333
hum              0.4800
windspeed        0.1045
casual           4.0000
registered      34.0000
cnt             40.0000
Name: 0.25, dtype: float64
instant       8690.0000
season           3.0000
yr               1.0000
mnth             7.0000
hr              12.0000
holiday          0.0000
weekday          3.0000
```

```
workingday       1.0000
weathersit       1.0000
temp             0.5000
atemp            0.4848
hum              0.6300
windspeed        0.1940
casual          17.0000
registered     115.0000
cnt            142.0000
Name: 0.5, dtype: float64

        cnt
0.10    9.0
0.25   40.0
0.50  142.0
```

```
#Q7)  Bivariate analysis

# Q7-1) Make boxplot for cnt vs. hour

#What kind of pattern do you see?

df # The data frame
```

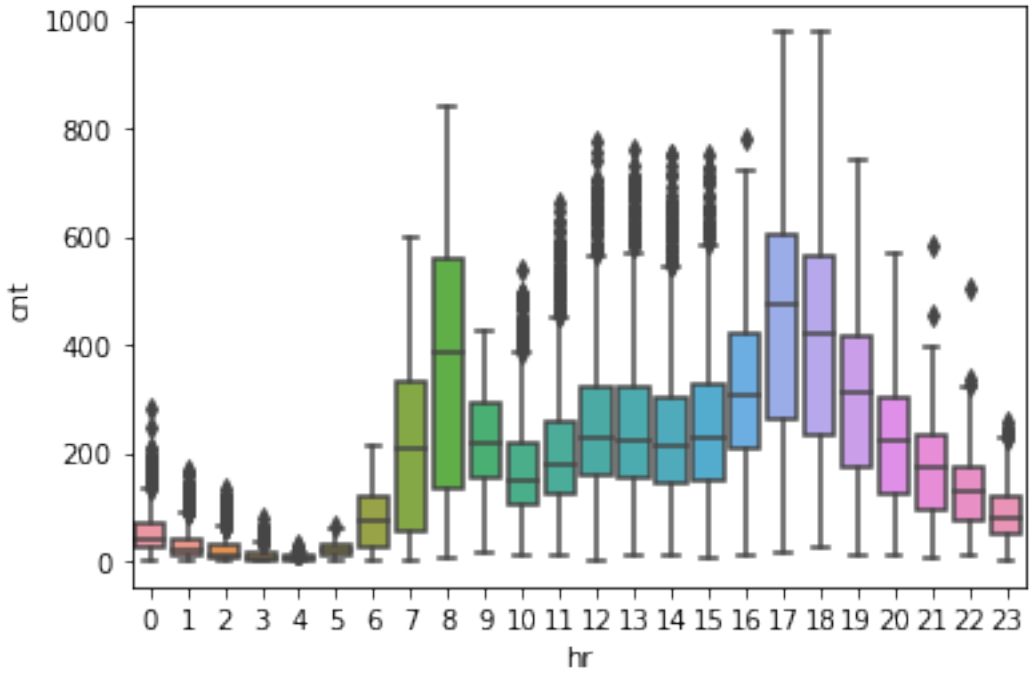
```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...
```

```
17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12
```

```
       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
import seaborn as sns

sns.boxplot(x="hr",y="cnt",data=df)   # Box Plot Analysis of cnt vs hr
```

```
<AxesSubplot:xlabel='hr', ylabel='cnt'>
```



From this box plot I can see that hr is the hour values which ranges from 0-23 cnt is the count of total number of registered and casual bike riders which ranges from 1 to 977(approx)

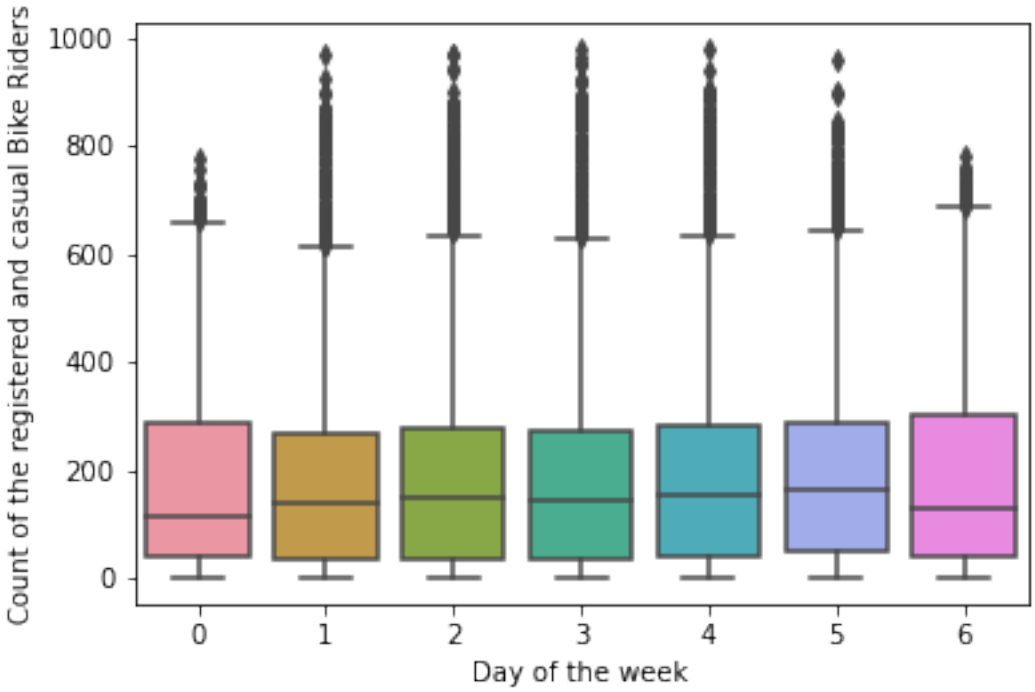
From this Box Plot it is clear that

1) In 0th hour the count of the bike riders is somewhere around 350 2) In 1st hour the count is almost 170 3) In 2nd Hour the count is 150

The count of the bike riders(casual+registerd) are decreasing from 0 to 5th hour The count of the bike riders are increasing from 6th to 8th hour Then there is a drastic drop of bike riders from 8th to 10th hour From 11th to 13th hour the count of bike riders are increasing in a slow pace. Then there is a small drop of bike riders from 13th to 14th hour, then it is increasing from 14th to 18th hour. After that there is a huge drop of bike riders from 19th to 23 hour

The maximum number of bike riders are in 17th hour and the minimum value of the bike riders are in 4th hour. Hence, box plot gives the clear view of bike riders count

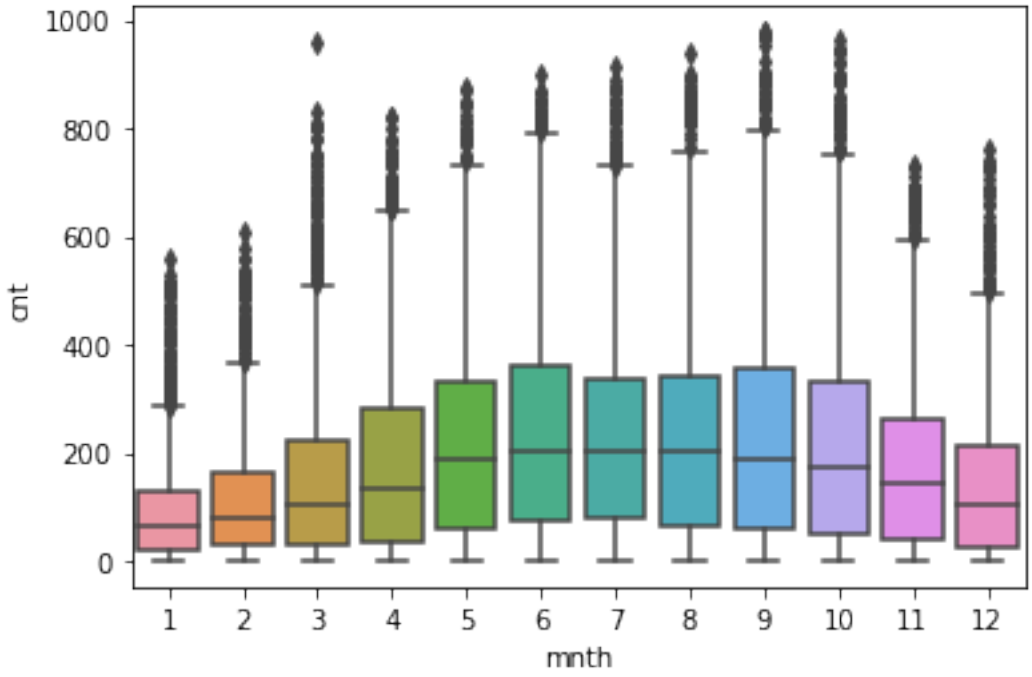
```
# Q7) 2--  Make boxplot for cnt vs. weekday

#Is there any difference in the rides by days of the week?

sns.boxplot(x="weekday",y="cnt",data=df)
plt.xlabel("Day of the week")
plt.ylabel("Count of the registered and casual Bike Riders")

Text(0, 0.5, 'Count of the registered and casual Bike Riders')
```



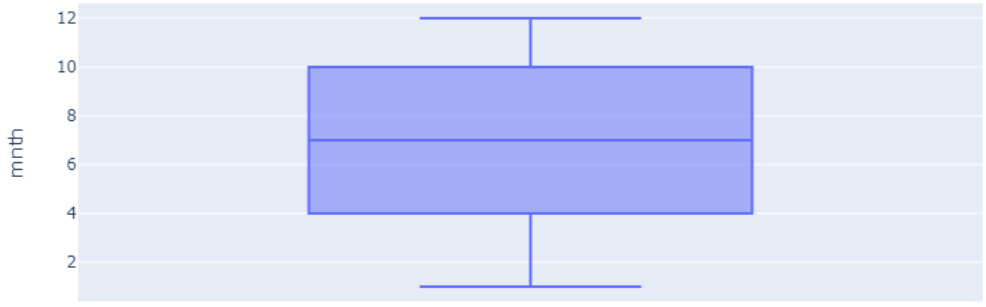
From this Box Plot it is clear that the total number of registered and casual Bike riders are lowest in 6th week and second lowest in 0th week However, the bike riders count are highest in 4th hour So in short, the bike riders are count are lowest in 0th hour and then from 1st to 5th week,there is a significant increase in the count of the bike riders and then in the last week which is the 6th one the bike riders count is the lowest

```
# Q7) 3-Make boxplot for cnt vs. month

# 7) 3-1) Look at the median values. Any month(s) that stand out?

import pandas as pd

sns.boxplot(x="mnth",y="cnt",data=df)  #  Box Plot of cnt vs Month
```

```
<AxesSubplot:xlabel='mnth', ylabel='cnt'>
```




```
import plotly
import plotly.express as px
fig=px.box(df,y="mnth")
fig.show()
```

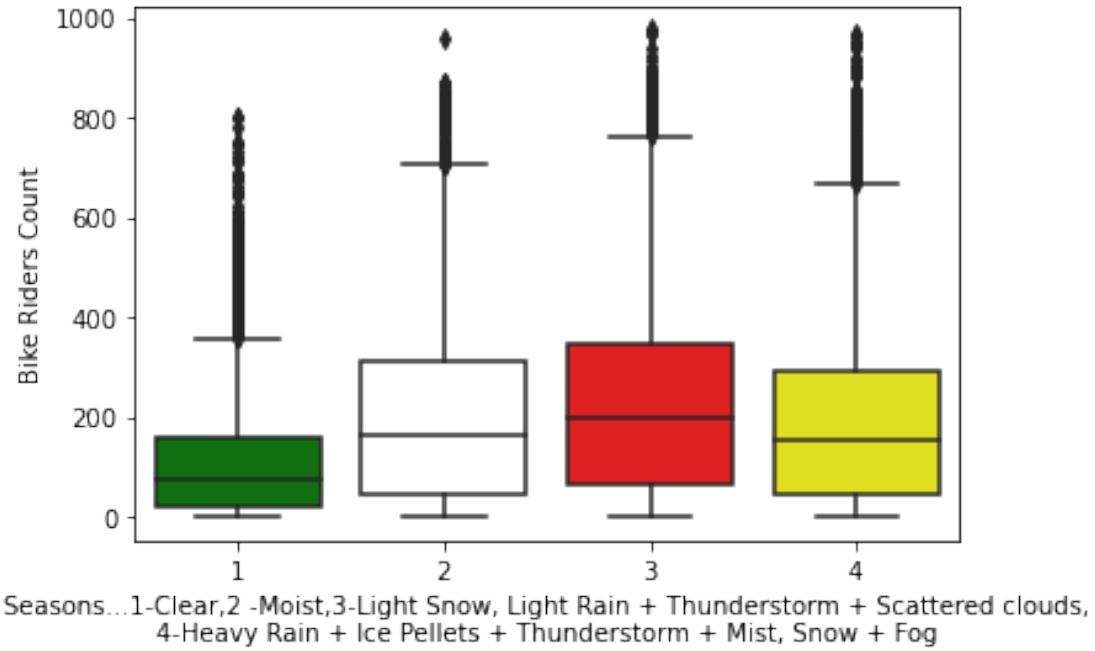


As the month ranges from 1 to 12. The median month is 7th month and the count of bike riders in this month is 977 and the month which stands out is the 9th month where the count of bike riders are 997(approx) very close to 1000 which is the maximum count of the bike riders

```
# Q7) 4--  Make boxplot for cnt vs. season
```

```
# 1--Which season has the highest rides in general? Expected?


sns.boxplot(x="season",y="cnt",data=df)  # Box Plot of count of number
of riders in different seasons
plt.xlabel('''Seasons...1-Clear,2 -Moist,3-Light Snow, Light Rain +
Thunderstorm + Scattered clouds,
4-Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog

''')
plt.ylabel("Bike Riders Count")
```

```
Text(0, 0.5, 'Bike Riders Count')
```



The third(3rd) season which is Light Snow, Light Rain + Thunderstorm + Scattered clouds is having the maximum number of bike riders. Count of Bike riders in 3rd season is almost 997 very close to 1000

```
#Q7)--Q5)Make a bar plot with the median value of cnt for each hr

#Q7) 1--Does this paint a different picture from the box plot?


import seaborn as sns
import statistics

df["cnt"]  # Data frame of total number of bike riders including
casual and registered
```

```
0         16
1         40
2         32
3         13
4          1
        ...
17374    119
17375     89
17376     90
17377     61
17378     49
Name: cnt, Length: 17379, dtype: int64
```

```
median_value_cnt=df["cnt"].quantile(0.50)  # Median Value of cnt
```

```
df.insert(16, "Median Value of Cnt", median_value_cnt, True)
```

```
df
```

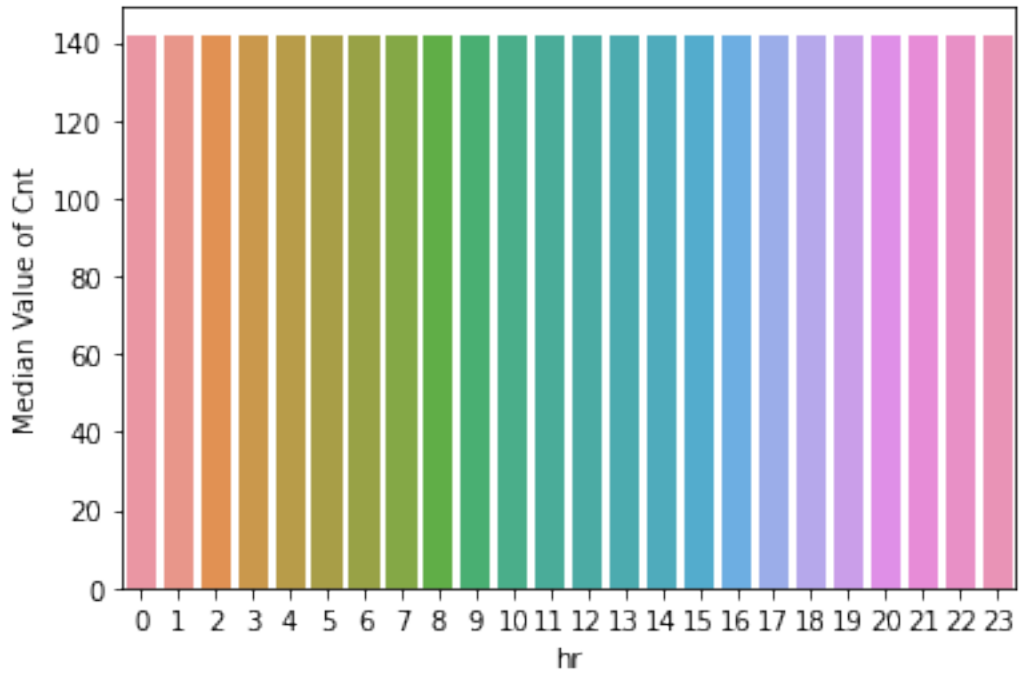
```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8
```

```
17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12
```

```
       registered  Median Value of Cnt  cnt
0              13                142.0   16
1              32                142.0   40
2              27                142.0   32
3              10                142.0   13
4               1                142.0    1
...           ...                  ...  ...
17374         108                142.0  119
17375          81                142.0   89
17376          83                142.0   90
17377          48                142.0   61
17378          37                142.0   49

[17379 rows x 18 columns]
```

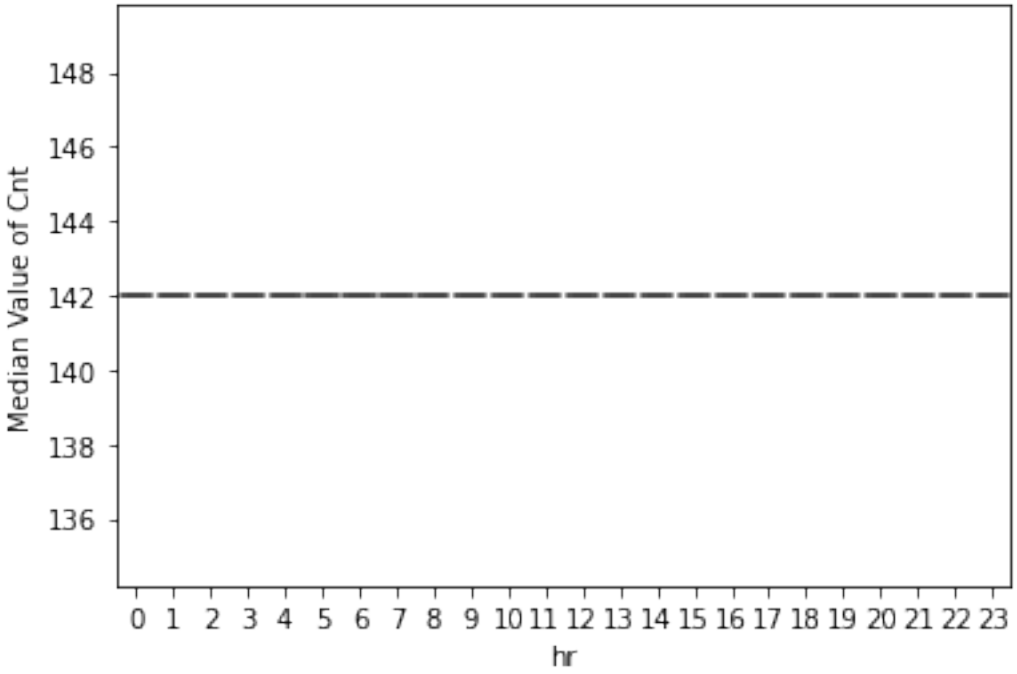
```
sns.barplot(x="hr",y="Median Value of Cnt",data=df)  # Bar plot of
median value of cnt for each hour
```

```
<AxesSubplot:xlabel='hr', ylabel='Median Value of Cnt'>
```




```
sns.boxplot(x="hr",y="Median Value of Cnt",data=df)  # Box plot of
median value of cnt for each hour
```

```
<AxesSubplot:xlabel='hr', ylabel='Median Value of Cnt'>
```



Yes,The boxplot of Median Value of Cnt vs Total Count of Bike riders paints the different picture if compare to boxplot

```
#Q7) 6--Make a correlation matrix for variables atemp, temp, hum, and
windspeed
#Which variables have the highest correlation?

df[["atemp","temp","hum","windspeed"]]
```

```
        atemp  temp   hum  windspeed
0      0.2879  0.24  0.81     0.0000
1      0.2727  0.22  0.80     0.0000
2      0.2727  0.22  0.80     0.0000
3      0.2879  0.24  0.75     0.0000
4      0.2879  0.24  0.75     0.0000
...       ...   ...   ...        ...
17374  0.2576  0.26  0.60     0.1642
17375  0.2576  0.26  0.60     0.1642
17376  0.2576  0.26  0.60     0.1642
17377  0.2727  0.26  0.56     0.1343
17378  0.2727  0.26  0.65     0.1343

[17379 rows x 4 columns]
```

```
df[["atemp","temp","hum","windspeed"]].corr()
```

```
               atemp      temp       hum  windspeed
atemp      1.000000  0.987672 -0.051918  -0.062336
```

```
temp       0.987672  1.000000 -0.069881  -0.023125
hum       -0.051918 -0.069881  1.000000  -0.290105
windspeed -0.062336 -0.023125 -0.290105   1.000000
```

From the above mentioned matrix one can analyze that

atemp and atemp column(variables) having the correlation 1 temp and temp column(variables) having the correlation 1 hum and hum column(variables) having the correlation 1 windspeed and windpseep column(variables) having the correlation 1

It basically means one to one relationship and is the perfect correlation

```
# Q8) Data preprocessing

#A few key considerations for the preprocessing:

#There are plenty of categorical features. Since these categorical features can’t be used in the predictive model, you need to convert to a suitable numerical representation. Instead of creating dozens of new dummy variables, try to club levels of categorical features wherever possible. For a feature with high number of categorical levels, you can club the values that are very similar in value for the target variable.


# Treating mnth column

# 1-- For values 5,6,7,8,9,10, replace with a single value 5. This is because these have very similar values for cnt.

# 2--Get dummies for the updated 6 mnth values



df # The data frame
```

```
       instant      dteday  season  yr  mnth  hr  holiday  weekday  \
0            1  01-01-2011       1   0     1   0        0        6
1            2  01-01-2011       1   0     1   1        0        6
2            3  01-01-2011       1   0     1   2        0        6
3            4  01-01-2011       1   0     1   3        0        6
4            5  01-01-2011       1   0     1   4        0        6
...        ...         ...     ...  ..   ...  ..      ...      ...
17374    17375  31-12-2012       1   1    12  19        0        1
17375    17376  31-12-2012       1   1    12  20        0        1
17376    17377  31-12-2012       1   1    12  21        0        1
17377    17378  31-12-2012       1   1    12  22        0        1
17378    17379  31-12-2012       1   1    12  23        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
```

```
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8

2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
df["mnth"] # Values of the mnth column
```

```
0         1
1         1
2         1
3         1
4         1
        ..
17374    12
17375    12
```

```
17376    12
17377    12
17378    12
Name: mnth, Length: 17379, dtype: int64
```

```
df["mnth"].replace(to_replace=[5,6,7,8,9,10],value=5)  # Replacing the
5,6,7,8,9,10 values to a single value 5
```

```
0         1
1         1
2         1
3         1
4         1
         ..
17374    12
17375    12
17376    12
17377    12
17378    12
Name: mnth, Length: 17379, dtype: int64
```

```
# 8 -1. 2) Get dummies for the updated 6 mnth values


df["mnth"]
```

```
0         1
1         1
2         1
3         1
4         1
         ..
17374    12
17375    12
17376    12
17377    12
17378    12
Name: mnth, Length: 17379, dtype: int64
```

```
pd.get_dummies(df["mnth"])  # Categorical Analysis of Month column
```

```
       1   2   3   4   5   6   7   8   9   10  11  12
0       1   0   0   0   0   0   0   0   0   0   0   0
1       1   0   0   0   0   0   0   0   0   0   0   0
2       1   0   0   0   0   0   0   0   0   0   0   0
3       1   0   0   0   0   0   0   0   0   0   0   0
4       1   0   0   0   0   0   0   0   0   0   0   0
...    ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
17374   0   0   0   0   0   0   0   0   0   0   0   1
17375   0   0   0   0   0   0   0   0   0   0   0   1
17376   0   0   0   0   0   0   0   0   0   0   0   1
17377   0   0   0   0   0   0   0   0   0   0   0   1
```

```
17378   0   0   0   0   0   0   0   0   0   0   0   1

[17379 rows x 12 columns]
```

```
pd.get_dummies(df["mnth"]).loc[:,0:6]  # Get dummies for the updated 6
mnth values

# Here : is all rows and  0: 6 is all 6 columns of the data frame
```

```
       1  2  3  4  5  6
0      1  0  0  0  0  0
1      1  0  0  0  0  0
2      1  0  0  0  0  0
3      1  0  0  0  0  0
4      1  0  0  0  0  0
...   .. .. .. .. .. ..
17374  0  0  0  0  0  0
17375  0  0  0  0  0  0
17376  0  0  0  0  0  0
17377  0  0  0  0  0  0
17378  0  0  0  0  0  0

[17379 rows x 6 columns]
```

```
# 8) 2-  Treating hr column

# 8) 2.1 --Create new mapping: 0-5: 0, 11-15: 11; other values are
untouched.
#Again, the bucketing is done in a way that hr values with similar
levels of cnt are treated the same.


df["hr"] # Data Frame of Hour column
```

```
0         0
1         1
2         2
3         3
4         4
         ..
17374    19
17375    20
17376    21
17377    22
17378    23
Name: hr, Length: 17379, dtype: int64
```

```
hours =
pd.Series(['0','1','2','3','4','5','6','7','8','9','10','11',"12","13"
,"14","15","16","17","18",
                  "19","20","21","22","23"])
```

```
print(hours)
```

```
0      0
1      1
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9
10    10
11    11
12    12
13    13
14    14
15    15
16    16
17    17
18    18
19    19
20    20
21    21
22    22
23    23
dtype: object
```

```
# Create Mapping

mapping_hours={"0": "0",
               "1": "0",
               "2": "0",
               "3": "0",
               "4": "0",
               "5": "0",
               "11":"11",
               "12": "11",
               "13": "11",
               "14": "11",
               "15": "11"
        }

print("The result of Mapping")
hours.map(mapping_hours)
```

```
The result of Mapping

0       0
1       0
2       0
```

```
3       0
4       0
5       0
6     NaN
7     NaN
8     NaN
9     NaN
10    NaN
11     11
12     11
13     11
14     11
15     11
16    NaN
17    NaN
18    NaN
19    NaN
20    NaN
21    NaN
22    NaN
23    NaN
dtype: object
```

```
# 8) 3-  Get dummy columns for season, weathersit, weekday, mnth, and
hr. You needn't club these further as the levels seem to have
different values
#for the median cnt, when seen
#from the box plots.

df # The data frame
```

```
       instant      dteday  season  yr  mnth   hr  holiday  weekday  \
0            1  01-01-2011       1   0     1  NaN        0        6
1            2  01-01-2011       1   0     1  NaN        0        6
2            3  01-01-2011       1   0     1  NaN        0        6
3            4  01-01-2011       1   0     1  NaN        0        6
4            5  01-01-2011       1   0     1  NaN        0        6
...        ...         ...     ...  ..   ...  ...      ...      ...
17374    17375  31-12-2012       1   1    12  NaN        0        1
17375    17376  31-12-2012       1   1    12  NaN        0        1
17376    17377  31-12-2012       1   1    12  NaN        0        1
17377    17378  31-12-2012       1   1    12  NaN        0        1
17378    17379  31-12-2012       1   1    12  NaN        0        1

       workingday  weathersit  temp   atemp   hum  windspeed
casual  \
0               0           1  0.24  0.2879  0.81     0.0000       3

1               0           1  0.22  0.2727  0.80     0.0000       8
```

```
2               0           1  0.22  0.2727  0.80     0.0000       5

3               0           1  0.24  0.2879  0.75     0.0000       3

4               0           1  0.24  0.2879  0.75     0.0000       0

...           ...         ...   ...     ...   ...        ...     ...

17374           1           2  0.26  0.2576  0.60     0.1642      11

17375           1           2  0.26  0.2576  0.60     0.1642       8

17376           1           1  0.26  0.2576  0.60     0.1642       7

17377           1           1  0.26  0.2727  0.56     0.1343      13

17378           1           1  0.26  0.2727  0.65     0.1343      12


       registered  cnt
0              13   16
1              32   40
2              27   32
3              10   13
4               1    1
...           ...  ...
17374         108  119
17375          81   89
17376          83   90
17377          48   61
17378          37   49

[17379 rows x 17 columns]
```

```
pd.get_dummies(df["season"])  # Dummy column for season
```

```
       1  2  3  4
0      1  0  0  0
1      1  0  0  0
2      1  0  0  0
3      1  0  0  0
4      1  0  0  0
...   .. .. .. ..
17374  1  0  0  0
17375  1  0  0  0
17376  1  0  0  0
17377  1  0  0  0
17378  1  0  0  0
```

```
[17379 rows x 4 columns]
```

```
pd.get_dummies(df["weathersit"]) # Dummy column for weathersit
```

```
       1  2  3  4
0      1  0  0  0
1      1  0  0  0
2      1  0  0  0
3      1  0  0  0
4      1  0  0  0
...   .. .. .. ..
17374  0  1  0  0
17375  0  1  0  0
17376  1  0  0  0
17377  1  0  0  0
17378  1  0  0  0

[17379 rows x 4 columns]
```

```
pd.get_dummies(df["weekday"]) #  Dummies value of weekday column
```

```
       0  1  2  3  4  5  6
0      0  0  0  0  0  0  1
1      0  0  0  0  0  0  1
2      0  0  0  0  0  0  1
3      0  0  0  0  0  0  1
4      0  0  0  0  0  0  1
...   .. .. .. .. .. .. ..
17374  0  1  0  0  0  0  0
17375  0  1  0  0  0  0  0
17376  0  1  0  0  0  0  0
17377  0  1  0  0  0  0  0
17378  0  1  0  0  0  0  0

[17379 rows x 7 columns]
```

```
pd.get_dummies(df["mnth"]) # Dummy columns for mnth column
```

```
       1   2   3   4   5   6   7   8   9   10  11  12
0       1   0   0   0   0   0   0   0   0   0   0   0
1       1   0   0   0   0   0   0   0   0   0   0   0
2       1   0   0   0   0   0   0   0   0   0   0   0
3       1   0   0   0   0   0   0   0   0   0   0   0
4       1   0   0   0   0   0   0   0   0   0   0   0
...    ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
17374   0   0   0   0   0   0   0   0   0   0   0   1
17375   0   0   0   0   0   0   0   0   0   0   0   1
17376   0   0   0   0   0   0   0   0   0   0   0   1
17377   0   0   0   0   0   0   0   0   0   0   0   1
17378   0   0   0   0   0   0   0   0   0   0   0   1
```

```
[17379 rows x 12 columns]
```

```
pd.get_dummies(df["hr"]) # Dummies value for hr column
```

```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, ...]

[17379 rows x 0 columns]
```

```
# Q9) Train test split: Apply 70-30 split.
#call the new dataframes df_train and df_test

import sklearn

from sklearn.model_selection import train_test_split

x=df["season"]  # Placing the season variable in x because it is a
categorical variable

y=df["cnt"]  # Cnt is the count of total number of registered and
unregistered bike riders

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,rand
om_state=0)

x_train
```

```
8112     4
12671    2
3889     2
10805    2
13741    3
        ..
9225     1
13123    3
9845     1
10799    2
2732     2
Name: season, Length: 12165, dtype: int64
```

```
df_train = pd.DataFrame(x_train, columns=['season'])

df_train   # Refined data frame
```

```
       season
8112        4
```

```
12671       2
3889        2
10805       2
13741       3
...       ...
9225        1
13123       3
9845        1
10799       2
2732        2

[12165 rows x 1 columns]
```

```
df_test = pd.DataFrame(x_test, columns=['season'])
```

```
df_test
```

```
       season
3439        2
6542        4
15470       4
9851        1
12640       2
...       ...
13321       3
5252        3
12510       2
6842        4
11262       2

[5214 rows x 1 columns]
```

```
#Q10) Separate X and Y for df_train and df_test. For example, you
should have X_train, y_train from df_train.
#y_train should be the cnt column from inp3 and X_train should be all
other columns.
```

```
df_train = pd.DataFrame(x_train, columns=['season'])
```

```
df_train
```

```
       season
8112        4
12671       2
3889        2
10805       2
13741       3
...       ...
9225        1
```

```
13123       3
9845        1
10799       2
2732        2

[12165 rows x 1 columns]
```

```
df_test = pd.DataFrame(x_test, columns=['season'])
```

```
df_test
```

```
       season
3439        2
6542        4
15470       4
9851        1
12640       2
...       ...
13321       3
5252        3
12510       2
6842        4
11262       2

[5214 rows x 1 columns]
```

```
inp3 = pd.DataFrame(y_train, columns=['cnt'])
```

```
inp3
```

```
       cnt
8112   250
12671   18
3889   107
10805  145
13741  857
...    ...
9225   257
13123  102
9845     6
10799   69
2732   530

[12165 rows x 1 columns]
```

```
inp3test = pd.DataFrame(y_test, columns=['cnt'])
```

```
inp3test
```

```
       cnt
3439     7
6542     5
```

```
15470  743
9851   208
12640  333
...    ...
13321   35
5252   571
12510  499
6842   302
11262  229

[5214 rows x 1 columns]
```

```
#Q10)  Model building

#Use linear regression as the technique

#Report the R2 on the train set

import sklearn

from sklearn.linear_model import LinearRegression

l=LinearRegression()  # Make the constructor of Linear Regression

x=df[["mnth"] ] # Taking the month column on x -axis

y=df["cnt"]  # Take the bike riders count on y-axis

l.fit(x,y)

LinearRegression()

l.intercept_

147.86081951758243

ywhat=l.predict(x)  # Linear Regression

ywhat

array([154.22418704, 154.22418704, 154.22418704, ..., 224.22122976,
       224.22122976, 224.22122976])

l.coef_

array([6.36336752])
```

In R2, the concept of Ridge Regression is used. Ridge Regression prevents overfitting

```
import sklearn

from sklearn.linear_model import Ridge

RigeModel=Ridge(alpha=0.1)
```

```
refinedr2=RigeModel.fit(df_test,inp3test)
```

```
refinedr2
```

```
Ridge(alpha=0.1)
```

```
# Q11) Make predictions on test set and report R2.
```

```
import statistics
```

```
import numpy
```

```
import sklearn
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.model_selection import cross_val_score
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.model_selection import cross_val_predict
```

Here I am taking the month column of data frame as an independent variable on x axis and cnt as total count of registered and casual bikers on y axis

```
x=df[["mnth"]]
```

```
y=df["cnt"]
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,rand
om_state=0)  # Here I am taking 30% of data for testing
```

```
l=LinearRegression()  # Make the constructor of Linear Regression
```

```
l.fit(x_test,y_test)  # Fitting the trained set of x and y
```

```
LinearRegression()
```

```
prediction_of_month=l.predict(x_test)
```

```
prediction_of_month
```

```
array([180.78303196, 209.30308909, 209.30308909, ..., 186.48704339,
       209.30308909, 175.07902054])
```

```
l.intercept_
```

```
152.2629748343664
```

```
l.coef_
```

```
array([5.70401143])
```

```
from sklearn.linear_model import Ridge
```

```
RigeModel=Ridge(alpha=0.1)
```

```
RigeModel.fit(x_test,y_test)
```

```
Ridge(alpha=0.1)
```

```
RigeModel.predict(x_test)
```

```
array([180.78304668, 209.30305783, 209.30305783, ..., 186.48704891,
       209.30305783, 175.07904445])
```

```
RigeModel=Ridge(alpha=0.6)
```

```
RigeModel.fit(x_test,y_test)
```

```
Ridge(alpha=0.6)
```

```
RigeModel.predict(x_test)
```

```
array([180.78312026, 209.30290151, 209.30290151, ..., 186.48707651,
       209.30290151, 175.07916401])
```

```
RigeModel=Ridge(alpha=0.9)
```

```
RigeModel.fit(x_test,y_test)
```

```
Ridge(alpha=0.9)
```

```
RigeModel.predict(x_test)
```

```
array([180.78316441, 209.30280773, 209.30280773, ..., 186.48709307,
       209.30280773, 175.07923575])
```

Here the concept of RidgeModel is used for analyzing overfiiting. x is the month column while y is the total number of registered and casual bike riders. x_test and y_test are basically the trained models of month and bike columns. This predict method is applied on the trained set of month and bike columns. If I increase the alpha the values in the arrays are fluctuating a bit

```
 pip install nbconvert[webpdf]
```

```
Requirement already satisfied: nbconvert[webpdf] in c:\users\shiva\
anaconda3\lib\site-packages (6.1.0)
Requirement already satisfied: testpath in c:\users\shiva\anaconda3\
lib\site-packages (from nbconvert[webpdf]) (0.5.0)
Requirement already satisfied: pygments>=2.4.1 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (2.10.0)
Requirement already satisfied: traitlets>=5.0 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (5.1.0)
Requirement already satisfied: defusedxml in c:\users\shiva\anaconda3\
lib\site-packages (from nbconvert[webpdf]) (0.7.1)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\
shiva\anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.5.3)
Requirement already satisfied: jupyter-core in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (4.8.1)
```

```
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (1.4.3)
Requirement already satisfied: jinja2>=2.4 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (2.11.3)
Requirement already satisfied: bleach in c:\users\shiva\anaconda3\lib\
site-packages (from nbconvert[webpdf]) (4.0.0)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.3)
Requirement already satisfied: jupyterlab-pygments in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.1.2)
Requirement already satisfied: nbformat>=4.4 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (5.1.3)
Requirement already satisfied: pyppeteer==0.2.2 in c:\users\shiva\
anaconda3\lib\site-packages (from nbconvert[webpdf]) (0.2.2)
Requirement already satisfied: websockets<9.0,>=8.1 in c:\users\shiva\
anaconda3\lib\site-packages (from pyppeteer==0.2.2->nbconvert[webpdf])
(8.1)
Requirement already satisfied: tqdm<5.0.0,>=4.42.1 in c:\users\shiva\
anaconda3\lib\site-packages (from pyppeteer==0.2.2->nbconvert[webpdf])
(4.62.3)
Requirement already satisfied: urllib3<2.0.0,>=1.25.8 in c:\users\
shiva\anaconda3\lib\site-packages (from pyppeteer==0.2.2-
>nbconvert[webpdf]) (1.26.7)
Requirement already satisfied: appdirs<2.0.0,>=1.4.3 in c:\users\
shiva\anaconda3\lib\site-packages (from pyppeteer==0.2.2-
>nbconvert[webpdf]) (1.4.4)
Requirement already satisfied: pyee<8.0.0,>=7.0.1 in c:\users\shiva\
anaconda3\lib\site-packages (from pyppeteer==0.2.2->nbconvert[webpdf])
(7.0.4)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\shiva\
anaconda3\lib\site-packages (from jinja2>=2.4->nbconvert[webpdf])
(1.1.1)
Requirement already satisfied: async-generator in c:\users\shiva\
anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0-
>nbconvert[webpdf]) (1.10)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\users\
shiva\anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0-
>nbconvert[webpdf]) (6.1.12)
Requirement already satisfied: nest-asyncio in c:\users\shiva\
anaconda3\lib\site-packages (from nbclient<0.6.0,>=0.5.0-
>nbconvert[webpdf]) (1.5.1)
Requirement already satisfied: pyzmq>=13 in c:\users\shiva\anaconda3\
lib\site-packages (from jupyter-client>=6.1.5->nbclient<0.6.0,>=0.5.0-
>nbconvert[webpdf]) (22.2.1)
Requirement already satisfied: tornado>=4.1 in c:\users\shiva\
anaconda3\lib\site-packages (from jupyter-client>=6.1.5-
>nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (6.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\shiva\
```

```
anaconda3\lib\site-packages (from jupyter-client>=6.1.5-
>nbclient<0.6.0,>=0.5.0->nbconvert[webpdf]) (2.8.2)
Requirement already satisfied: pywin32>=1.0 in c:\users\shiva\
anaconda3\lib\site-packages (from jupyter-core->nbconvert[webpdf])
(228)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\
shiva\anaconda3\lib\site-packages (from nbformat>=4.4-
>nbconvert[webpdf]) (3.2.0)
Requirement already satisfied: ipython-genutils in c:\users\shiva\
anaconda3\lib\site-packages (from nbformat>=4.4->nbconvert[webpdf])
(0.2.0)
Requirement already satisfied: setuptools in c:\users\shiva\anaconda3\
lib\site-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.4-
>nbconvert[webpdf]) (58.0.4)
Requirement already satisfied: pyrsistent>=0.14.0 in c:\users\shiva\
anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4-
>nbformat>=4.4->nbconvert[webpdf]) (0.18.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\shiva\
anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4-
>nbformat>=4.4->nbconvert[webpdf]) (21.2.0)
Requirement already satisfied: six>=1.11.0 in c:\users\shiva\
anaconda3\lib\site-packages (from jsonschema!=2.5.0,>=2.4-
>nbformat>=4.4->nbconvert[webpdf]) (1.16.0)
Requirement already satisfied: colorama in c:\users\shiva\anaconda3\
lib\site-packages (from tqdm<5.0.0,>=4.42.1->pyppeteer==0.2.2-
>nbconvert[webpdf]) (0.4.4)
Requirement already satisfied: webencodings in c:\users\shiva\
anaconda3\lib\site-packages (from bleach->nbconvert[webpdf]) (0.5.1)
Requirement already satisfied: packaging in c:\users\shiva\anaconda3\
lib\site-packages (from bleach->nbconvert[webpdf]) (21.0)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\shiva\
anaconda3\lib\site-packages (from packaging->bleach-
>nbconvert[webpdf]) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```