Bike Share Rental Analysis

BootCamp on Data Science and Tools

Submitted

To

CRC-Training

By

Vansheta Sharma
Roll Number: 2102901549006

Priya
Roll Number: 2102901549003

Shubham Pal
Roll Number: 2002901540085

Under the guidance of
Mr. Gaurav Kansal / Mr. Gopal Gupta

TABLE OF CONTENTS

CERTIFICATE

This is to certify that Project Report entitled **“Bike Share Rental Analysis”** which is submitted by **Vansheta Sharma, Priya** and **Shubham Pal** in partial fulfillment of the requirement for the “Bootcamp on Data Science and Tools” in Department of CRC-Training of ABES Institute of Technology, is a record of the candidate own work carried out by him under my/our supervision.

Mr. Gaurav Kansal

Mr. Gopal Gupta

Date: 10/04/2023

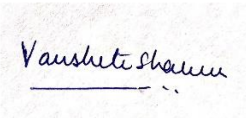
ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the "Bootcamp on Data Science and Tools" undertaken during B.Tech, 3rd Year. We owe special debt of gratitude to Mr. Gopal Gupta and Mr. Gaurav Kansal for his constant support and guidance throughout the course of our work. His constant motivation has been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of team members of CRC-Training for their full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the motivation of Department Of Computer Science And Engineering, ABES Institute of Technology to provide us the opportunity to undergo training at CRC-Training.

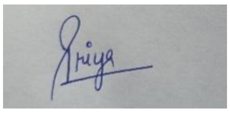
Signature:



Name : Vansheta Sharma

Roll No.: 2102901549006

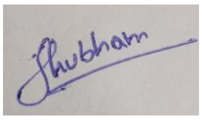
Signature:



Name : Priya

Roll No.: 2102901549006

Signature:



Name : Shubham Pal

Roll No.: 2002901540085

ABSTRACT

A growing number of people are using public bicycle rentals in Washington, DC, because to their greater convenience and environmental sustainability. This study examines how the demand for shared public bikes in Washington, DC, vary with the weather.

The dataset has been downloaded from Kaggle: Machine Learning and Data Science Community. Bike-sharing systems are meant to rent bicycle and return to the different place for the bike sharing purpose in Washington DC. It is the rental data spanning for 2 years.

Therefore, the main objective of this study was to understand demand factors for bicycles and their relationship to demand by answering business questions. In this study, Python software and Jupyter Notebook were used to determine the relationships between datasets.

We also used the dataset for data visualization. To get insight into how the values submitted into each attribute relate to the quantity of entries, we did univariate analysis on each category. Finally, we ran a bivariate analysis of that specific attribute with the date set target variable to examine how the values entered for that attribute correspond to the target variable.

The analysis's findings indicate that temperature is the most crucial factor, and there is a link between favorable weather and the demand for bicycles. From the correlation chart it was seen that the month, weather and temperature played most crucial role in bike renting whereas the on the weekly basis the renting was more dependent on the customer type, whether they were registered users or casual users.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Definition: -

- The need for renting bicycles is growing every day, so controlling the demand is essential for providing good service.
- In order to determine which type of weather is more responsible for shifting rental bike demand in each hour, and what are the areas that needs more improvement for increasing revenue this research analysis is helpful.
- Business can then select what kind of measures to implement to balance the current supply and demand for bicycles

1.2 Motivation:

There are several reasons why we wanted to analyze the Bike Rental Systems data, as there are many benefits to it as follows:

- Convenience: Bike rental services offer a convenient way to get around a city or town, without the need to own or maintain a bike.
- Cost savings: Renting a bike can be a more cost-effective option for short-term use compared to purchasing a bike, especially for those who only need a bike occasionally.
- Environmental benefits: Using a bike instead of a car or public transportation can reduce carbon emissions and help to promote a more sustainable way of commuting.
- Health benefits: Biking is a form of exercise and can provide a low-impact workout while commuting or exploring a new area.

1.3 Objective of the Project:

- Find the relationship between variables and provide accurate analysis to enhance the business decisions.
- Identifying seasonal variation in bike renting.
- Identify most busy hours and provide extra supply services for all users.

.

1.4 Scope of the Project: A bike renting system analysis report can provide valuable insights into the bike rental industry, and help stakeholders make informed decisions on how to develop and operate a successful bike rental service.

1.5 Need of Work: - Despite the growing popularity of bike rental services, there is still a lack of awareness and adoption in some areas, indicating a need for more targeted marketing and outreach efforts.

CHAPTER 2
RELATED WORK

We have read many articles on websites about our projects. We got much useful information from it about data set and attributes in our data sets. This section will contain reference about two such papers. After reading the papers we got these basic ideas about dataset:

- The first paper is of Laboratory of Artificial Intelligence and Decision Support (LIAAD), University of Porto, INESC Porto, Campus da FEUP
- Under the supervision of Hadi Fanaee-T.
- Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

- The second paper is written by four people: Santiago Brusseau, Eric Chen, Cherie Hua, Adrienne Wang
- Their dataset contains daily ridership data of the Capital Bikeshare system in Washington D.C. of the years 2011 and 2012.
- There are 731 observations in total, where the data of each day is recorded as a single observation with 14 variables.
- Their objective included following:

  - Who are our main customers?
  - When do people use bike sharing?
  - Under what conditions do people use bike sharing?

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Dataset Description

- The data we will look into is downloaded and extracted from Kaggle.
- This bike share rental data of Capital Bikeshare only contains entries sampled from Washington D.C. spanning two years dating from January 1st, 2011 to December 19th, 2012.
- The dataset is also joined by the weather statistics for the corresponding date and time.
- It has total of 12 attributes. They are as follows:

1. "datetime", containing hourly date in timestamp format;
2. "season", containing integers 1 to 4 representing "Winter", "Spring","Summer","Fall";
3. "holiday", containing Boolean expressions in 1s and 0s representing whether the day of the observation is a holiday or not;
4. "workingday", containing Boolean expressions in 1s and 0s representing whether the day of the observation is a working day or not;
5. "weather", containing integers 1 to 4 representing four different lists of weather conditions:
    - 1: Clear or cloudy
    - 2: Mists
    - 3: Light rain or snow,
    - 4: Heavy rain, snow or even worse weather.
6. "temp", containing values of temperature at the given time;
7. "atemp", containing values of feeling temperature at the given time;
8. "humidity", containing values of relative humidity level at the given time, in the scale of 1 to 100;
9. "windspeed", containing values of wind speed, in mph (miles per hour);
10. "casual", containing the count of non-registered user rentals, across all stations;
11. "registered", containing the count of registered user rentals, across all stations;
12. "count", containing the total count of rentals at the given hour, across all stations.

3.2 Methods:

After downloading the data set, we begin from analyzing the data set’s attributes, then preprocessing it.

With the help of **matplotlib** and **seaborn** we visual each attribute with target variable.

1. Then we used univariate analysis, bivariate analysis and multivariate analysis to find relations and insights from our data.

Exploratory Data Analysis (EDA)

Exploratory data analysis is the process of studying or analyzing the data and drawing conclusions about its primary features. The two categories into which EDA is often divided are graphical analysis and non-graphical analysis.

EDA is crucial since, before getting your hands dirty, it's a good idea to comprehend the issue statement and the numerous connections between the data characteristics.

EDA has the following basic steps:

- Examine the data distribution
- Handling missing values of the dataset(a most common issue with every dataset)
- Handling the outliers
- Removing duplicate data
- Encoding the categorical variables
- Normalizing and Scaling

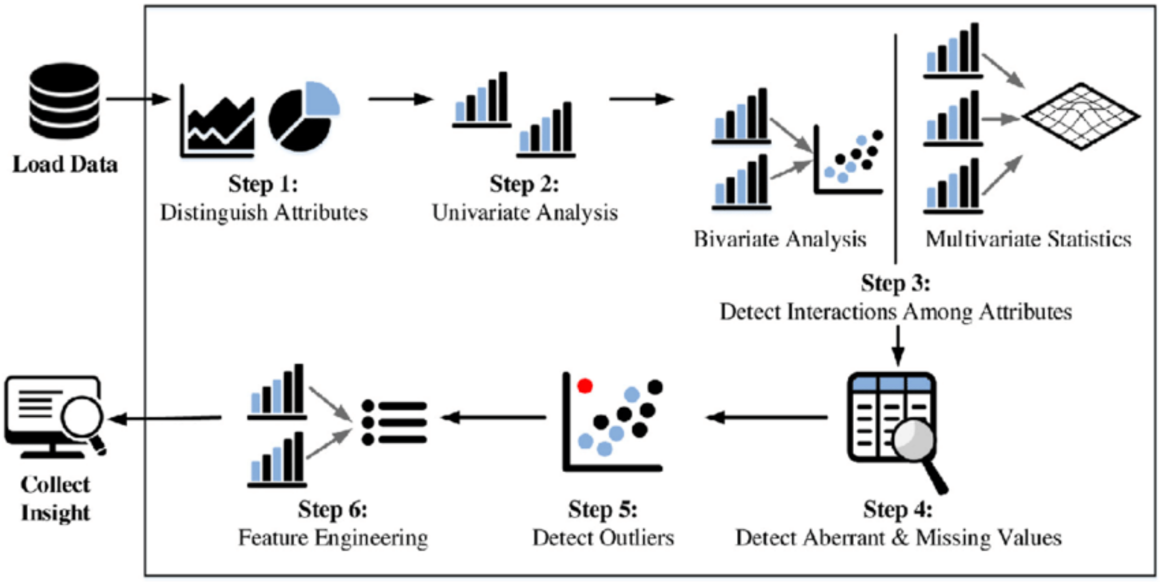


Fig. 3.1: Process of EDA

I. Univariate Analysis:

This kind of data just has one variable. As there is only one variable that varies, univariate data analysis is the most straightforward type of analysis. The analysis's primary goal is to explain the data and identify any patterns in it; it does not deal with causes or correlations. Height is an example of a univariate data.

II. **Bivariate Analysis:**

This type of data involves **two different variables**. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables. Example of bivariate data can be temperature and ice cream sales in summer season.

III. **Multivariate Analysis:**

Multivariate data refers to data that has three or more variables. For instance, if an online marketer wanted to compare the popularity of four ads, they might analyse the click rates for men and women and then look at the correlations between the variables. It is comparable to bivariate but has more dependent variables than that. The methods used to analyse this data depend on the objectives to be met. Regression analysis, path analysis, factor analysis, and multivariate analysis of variance are a few of the methodologies (MANOVA).

3.3 Hardware / Software Requirements

Minimums Hardware Requirements:

RAM: 2 GB
Processor: Intel Core 2 Duo
Hard disk: 50GB

Minimums Software Requirements:

Pandas: 0.24.2
Numpy; 1.16.4
Matplotlib: 3.1.0
Seaborn: 0.9.0
Python: 3.7
Scikit-Learn: 0.21.2

3.4 Our Methodology:

- Firstly, we downloaded the data set from Kaggle: Your Machine Learning and Data Science Community
- Then we loaded the data set into Jupyter Notebook

```
In [1]:  import pandas as pd
         import numpy as np
         from datetime import datetime
         import matplotlib.pyplot as plt
         import seaborn as sns
```

Fig. 3.2: Importing numpy and pandas

- Importing numpy , pandas, datetime, matplotlib and seaborn packages and module.
- NumPy is a python library used for working with arrays.
- Pandas is a high-level data manipulation tool.
- Datetime module supplies classes to work with date and time.
- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

```
In [2]:  data = pd.read_csv('train_bikes.csv')
         data
```

Fig. 3.3: read_csv()

- **Describing the data:**

```
In [6]:  data.describe()
Out[6]:
```

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000

Fig. 3.4: describe() function

Using **describe()** function to get the basic idea of all the Int variable of data set.

- **Getting Information about data:**

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  object
 1   season      10886 non-null  int64
 2   holiday     10886 non-null  int64
 3   workingday  10886 non-null  int64
 4   weather     10886 non-null  int64
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64
 10  registered  10886 non-null  int64
 11  count       10886 non-null  int64
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

Fig. 3.5: info() function

The **info()** function is used to information about each and every attribute.

- There is 1 object variable
- There are 8 int64 variable
- There are 3 float64 variable
- There are 10886 rows entries

The total size of data set is 1020.7+ KB

```
data.head(40)
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000	5	27	32
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000	3	10	13
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0000	0	1	1
5	2011-01-01 05:00:00	1	0	0	2	9.84	12.880	75	6.0032	0	1	1

Fig. 3.6: head() function

Using the **head() function** to get a basic idea about how the data is entered.

- **Looking at the Shape of dataset :**

```
In [4]:  data.shape

Out[4]: (10886, 12)
```

Fig. 3.7: Insignificant columns

Using shape attribute we have observed that there were 10886 rows and 12 columns in our Bike Rental dataset.

- **Checking missing values for each column :**

```
In [8]:  data.isnull().sum()

Out[8]: datetime      0
        season        0
        holiday       0
        workingday    0
        weather       0
        temp          0
        atemp         0
        humidity      0
        windspeed     0
        casual        0
        registered    0
        count         0
        dtype: int64
```

Fig. 3.8: isin() function

After running this we get that there are no missing or null values present in our dataset.

- **Splitting the Datetime Object**

  - ✓ First we define a splitting function as **split_d()**
  - ✓ Then using **split_d()** function we split the date time object into separate columns as:
    - Day
    - Month
    - Year
    - Hour
    - Dates
    - Time
    - Weekday
  - ✓ It can be seen that total row is reduced to 30178 from 32561.

```
def split_d(data):
    data['Day'] = pd.to_datetime(data['datetime']).dt.day
    data['Month'] = pd.to_datetime(data['datetime']).dt.month
    data['Year'] = pd.to_datetime(data['datetime']).dt.year
    data['hour'] = pd.to_datetime(data['datetime']).dt.hour
    data['Dates'] = pd.to_datetime(data['datetime']).dt.date
    data['Time'] = pd.to_datetime(data['datetime']).dt.time
    data['Weekday'] = pd.to_datetime(data['datetime']).dt.dayofweek
    return data
```

Fig. 3.9: Splitting function

```
newdata = split_d(data)
newdata.head(5)
```

Fig. 3.16: Splitting Object

Day	Month	Year	hour	Dates	Time	Weekday
1	1	2011	0	2011-01-01	00:00:00	5
1	1	2011	1	2011-01-01	01:00:00	5
1	1	2011	2	2011-01-01	02:00:00	5
1	1	2011	3	2011-01-01	03:00:00	5

Fig. 3.17: After Splitting

After splitting the datetime object:

- The new columns are split according to day/date, month of the year, hour of observation, the whole date, time in hh:mm:ss format and weekday.
- Weekdays are categorized as, 0-monday 1-Tuesday, 2-Wednesday, 3-Thursday, 4-Friday, 5-Saturday, 6-Sunday.

```
datetime       object
season          int64
holiday         int64
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
Day             int64
Month           int64
Year            int64
hour            int64
Dates          object
Time           object
Weekday         int64
dtype: object
```

Fig. 3.18: dtype of newdata

- **Analysing each attribute :**

<u>Season</u>

```
season
Skew : -0.01
```

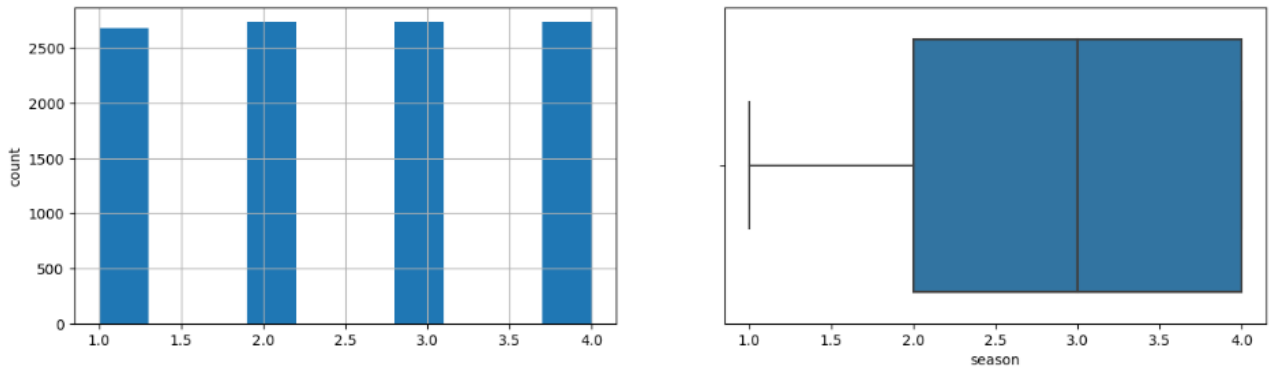


Fig. 3.10: Season univariate analysis

- Above Visualization, shows the season distribution among the entries in our dataset.
- The season is categorical attribute with values as 1, 2, 3, and 4 which shows four different seasons.
- According to our observations, the season are Spring, Fall, Summer and Winter respectively.

Holiday

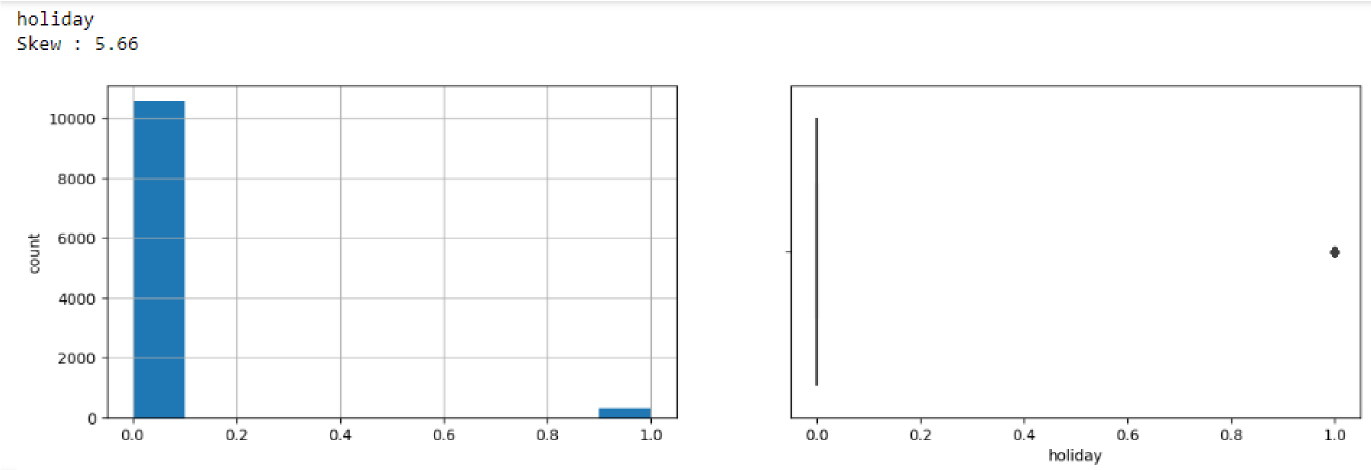


Fig. 3.11: Holiday univariate analysis

- In holiday attribute the two values show the Boolean True and False observations.
- 1 for holiday being true and 0 for holiday being false.
- In the dataset it is observed that there were not a lot days with holiday.

WorkingDay

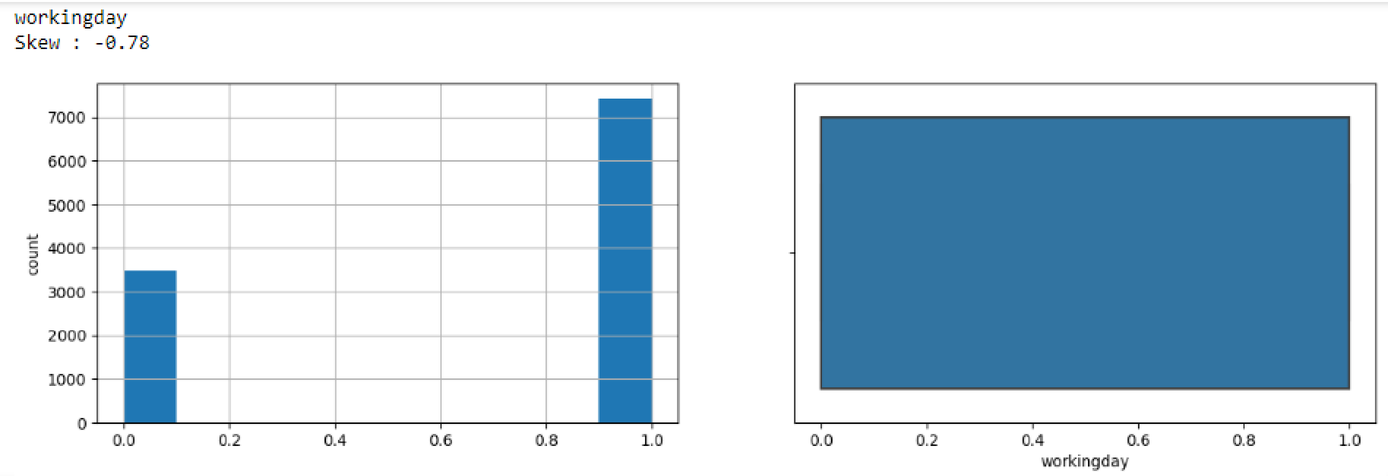


Fig. 3.12: Workingday univariate analysis

- In workingday attribute the two values show the Boolean True and False observations.
- 1 for working day being true and 0 for working day being false.
- In the dataset it is observed that there were not a lot days with holiday as compared to workingdays.

Weather

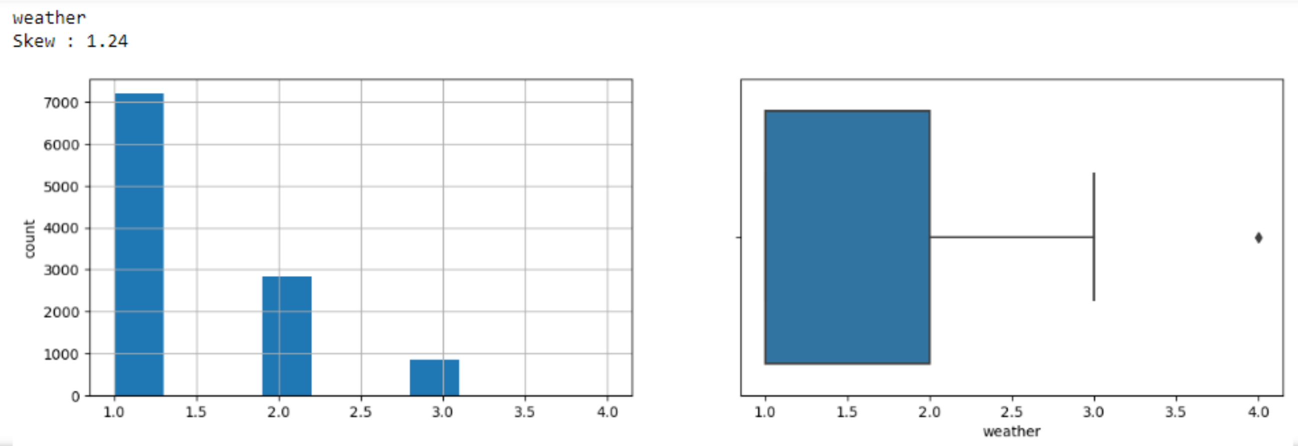


Fig. 3.13: Weather univariate analysis

- There are four categories of weather, in which mostly bike renting happened in 1st category.
- Least bike renting is seen in 4th category of weather.
- Later in analysis it is found weather is directly related to season and month attributes.

Temperature

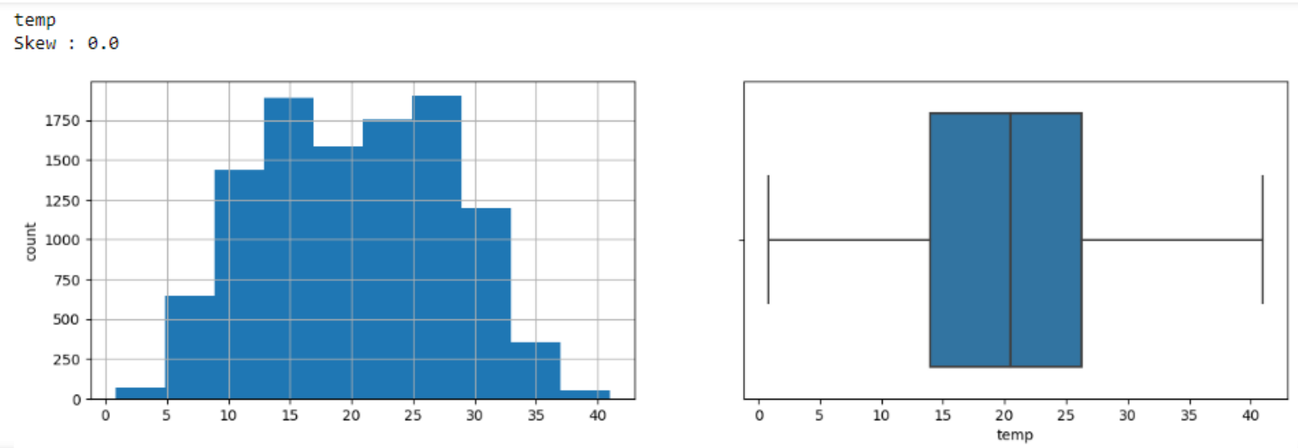


Fig. 3.14: Temperature univariate analysis

From analysing temp variable we found:

- The bike renting observed also had some relation with temperature.
- Maximum renting occurred during when the average temperature ranged from 15-25 degree Celsius.

Humidity

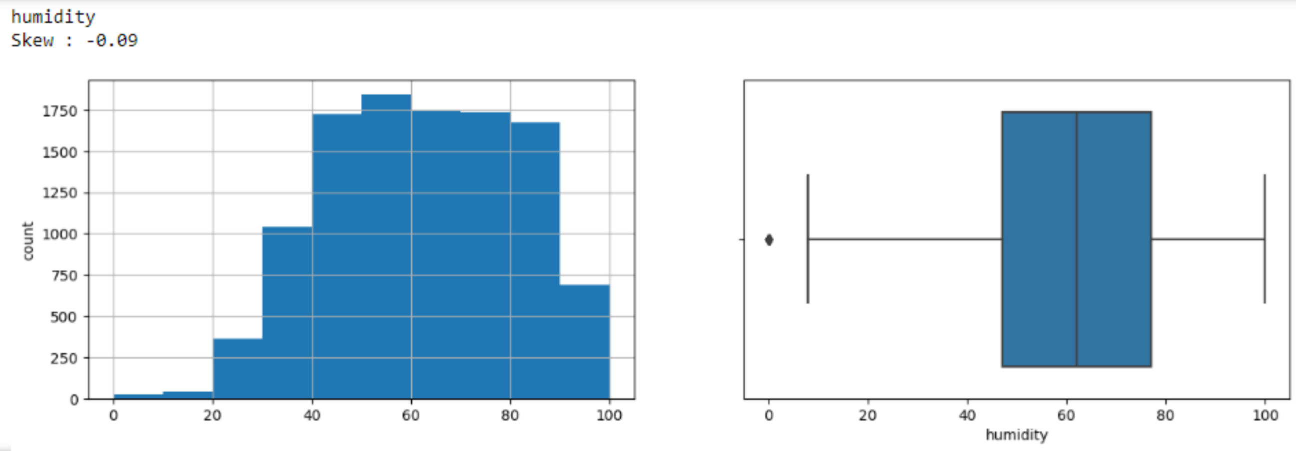


Fig. 3.15: Humidity univariate analysis

From analysing humidity variable we found:

- Humidity variable states the level of humidity on the day or hour the entry was made of bike renting.
- It is observed that maximum bike rental has occurred when level was in the range of 50-60.
- This indicates that light pouring has increased bike rental, and minimum is when the humidity is 0.

Windspeed

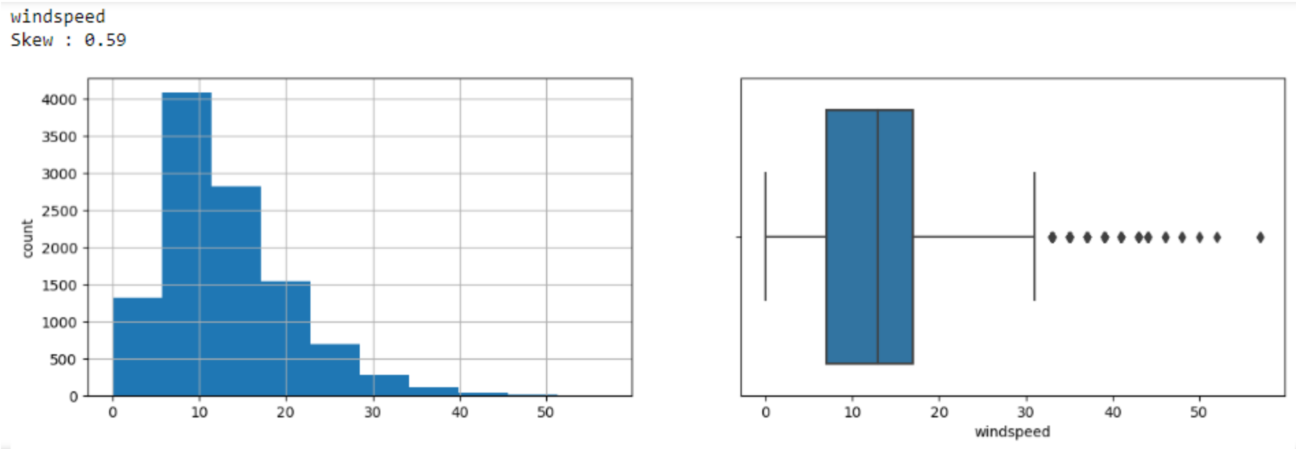


Fig. 3.16: Widspeed univariate analysis

From analysing windspeed variable we found:

- Windspeed variable states the velocity of wind on the day or hour particular entry was made for bike renting.
- It is observed that maximum bike rental has occurred when velocity was around 10 units.
- This indicates that people preferred bikes during pleasant weather and on light windy days.

Casual Renting

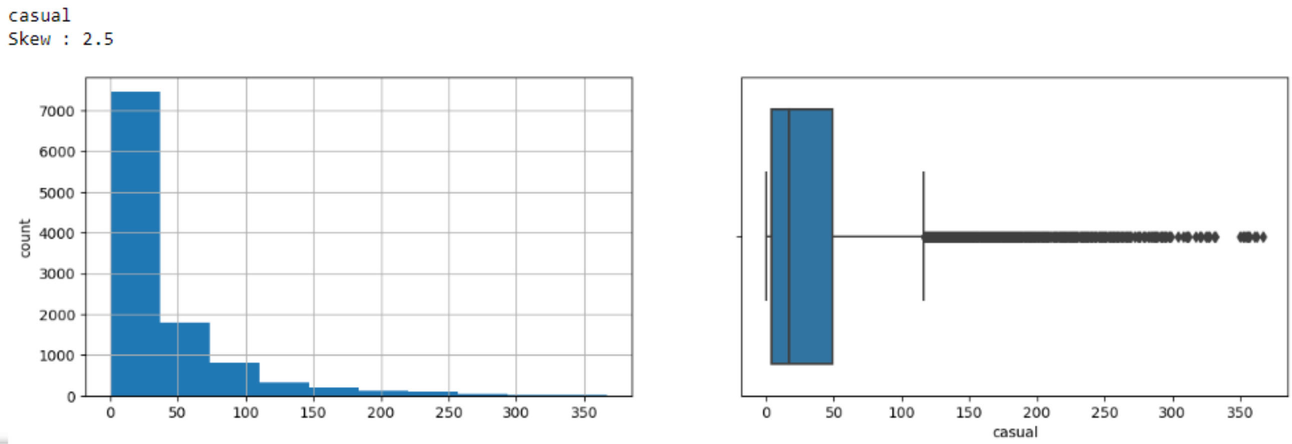


Fig. 3.17: Casual Renting univariate analysis

From analysing casual variable we found:

- Casual variable indicates the number of non-registered users that used the service.
- It is observed that the average casual bike renting varies from 0-50 and maximum is 110.

Registered Renting

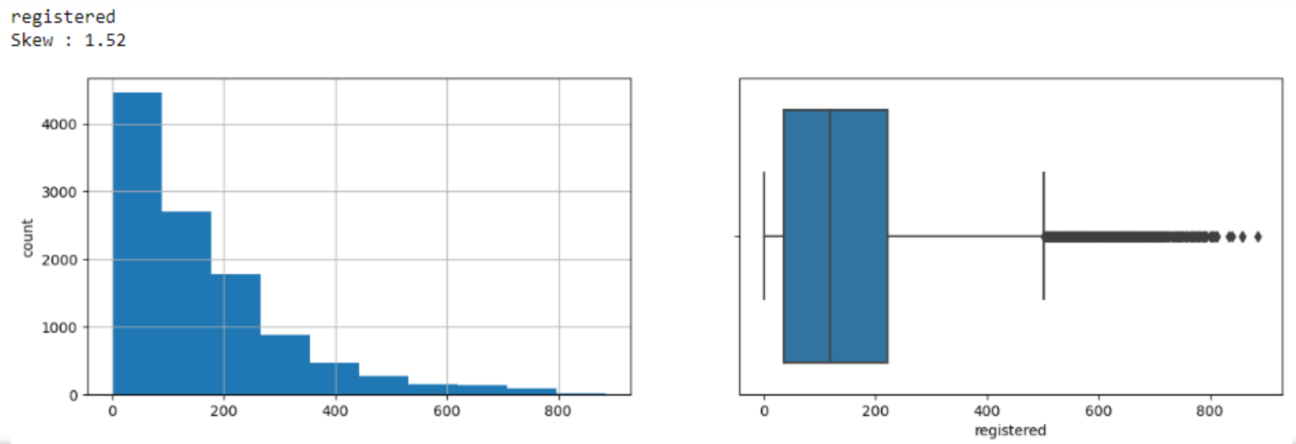


Fig. 3.18: Registered Renting univariate analysis

From analysing registered variable we found:

- Registered variable indicates the number of registered users that used the service.
- It is observed that the registered bike renting is maximum range is from 0-200+ with count 4000+ and maximum is 400+ (excluding the outliers).

Total Count

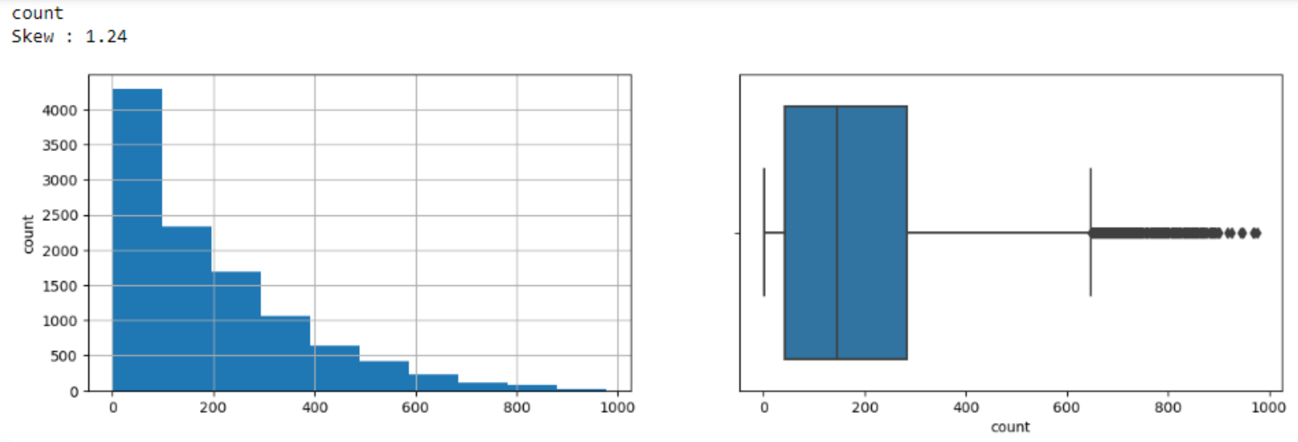


Fig. 3.19: Total Count univariate analysis

From analysing count variable we found:

- Count variable indicates the total number of bike rental her hour of the day, including both casual users and registered users.
- The maximum range of bike renting has been 0-200+.

Weekday

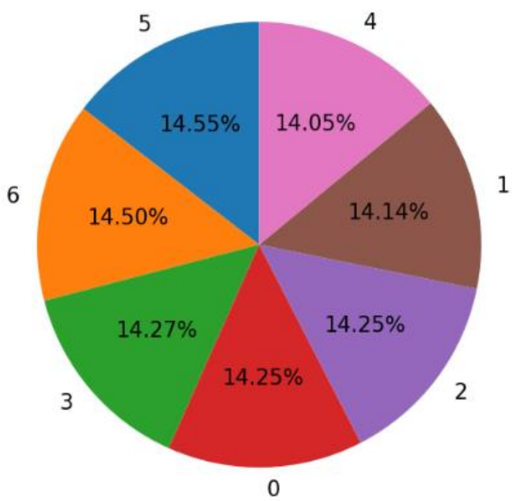


Fig. 3.20: Workclass vs. No. of entries

From analysing weekday variable we found:

- The weekday variable shows the day on which the bike was rented.
- On the weekends, like Saturday Sunday the bike renting is most busy and when it comes to working days Thursdays are most likely to be busy for bike renting organization.
- The least busy weekday is 4th (Friday) from a very little margin as shown in above figure.

Season vs Count:

```
b = sns.barplot(x="season",y="count",data=newdata,hue="Year")
plt.ylabel(' Rental bikes counts')
plt.xlabel('Season')
plt.title("The count of user in year based on season")
plt.legend(title='Year', loc='best', labels=['2011', '2012'], frameon=True)
plt.show(b);
```

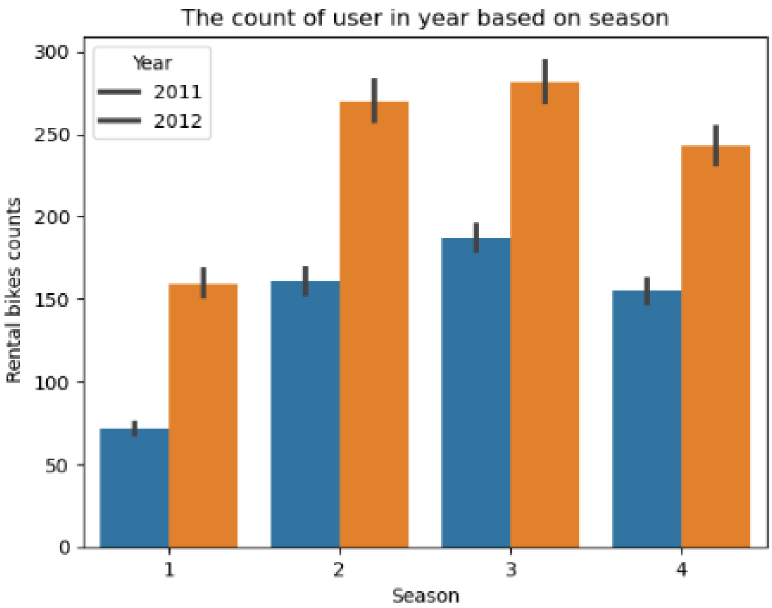


Fig. 3.21: Season vs. Count

From the Graph of Season and Total Count:

- Least bike renting count in season-1 and most in season-3.
- The bike renting in year 2012 was more as compared to year 2011.

Registered vs Casual with other attributes:

Week:

- For registered users Thursday is the busiest day among all the working days, but the weekends show the least bike renting.
- For casual users the bike renting is maximum on weekends and least on Wednesday or other working days.

Hour:

- For registered users the busiest hour is 5pm (17), and least is 4am.
- For casual users there is not much difference as most busy hour is 2pm (14), and least is 4am and 5 am.

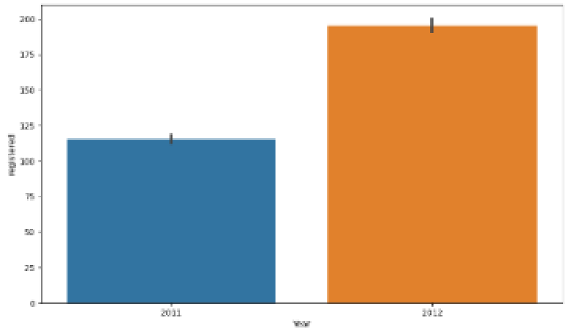
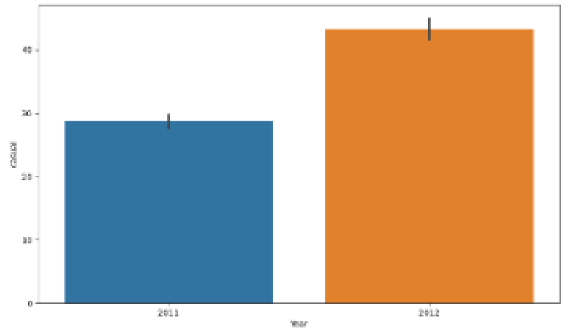
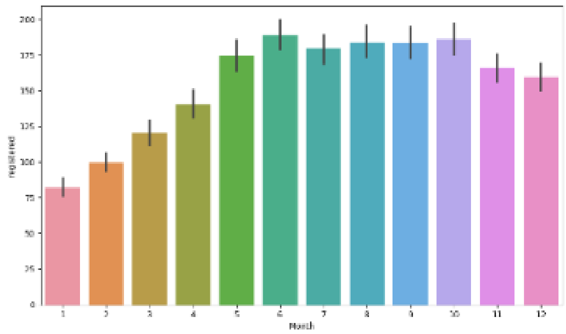
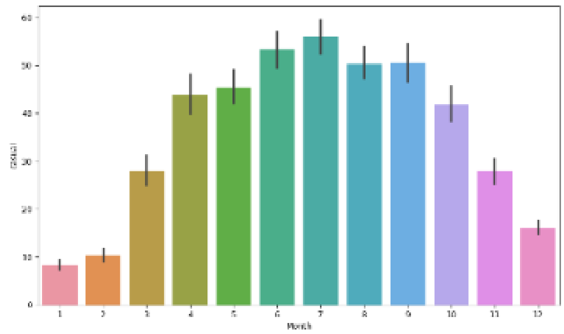
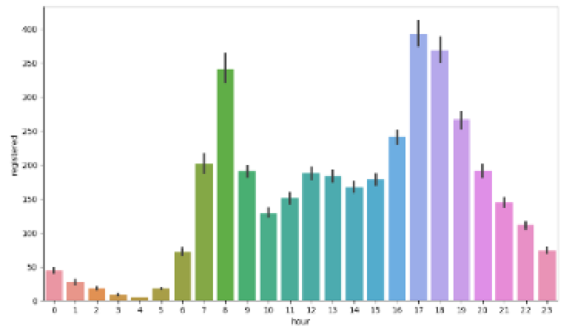
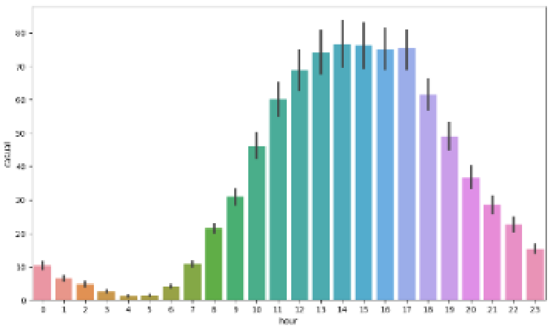
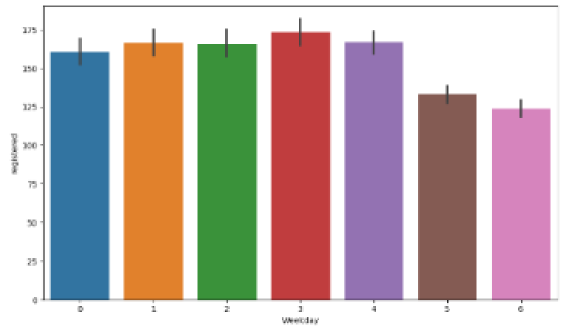
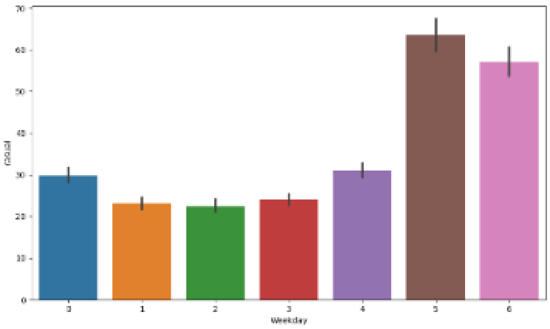
Month:

- For registered users June was the busiest month and January was the least busy month. It can be seen as the season is directly proportional to the months, from season 1 to 3 monthly renting increases and after that it varies.

Year:

- For both the registered and casual users the year 2012 was more busy than 2011.

Registered ***Casual***

Weather vs Count :

```
sns.barplot(x='weather', y= 'count',data= newdata)
plt.show()
```

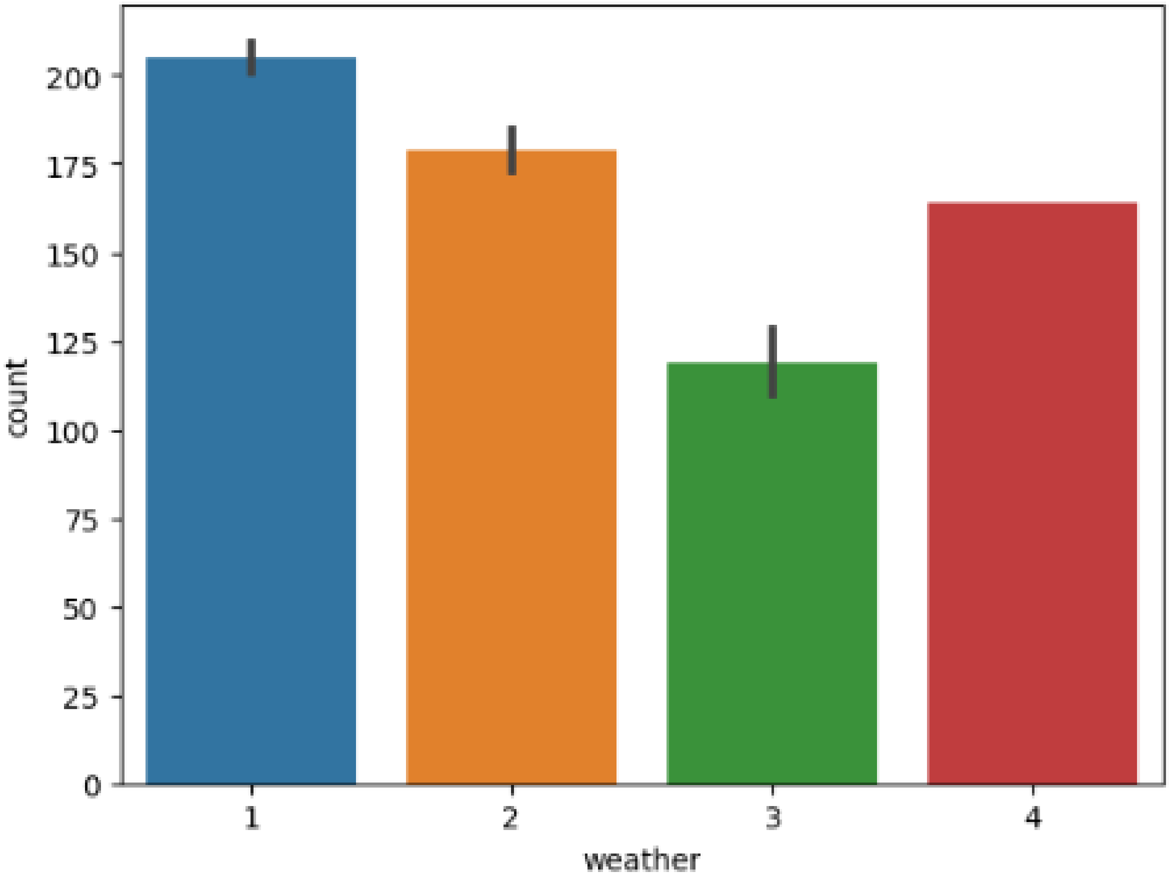


Fig. 3.22: Weather vs. Count

From the Graph of Weather and Count:

- As the weather value increases the temperature decreases. So we can say that 1-hot weather and 4-cold weather, and 2 and 3 lies in between.
- From the bar plot and data frame above it is clear that people prefer bikes mostly in hot weather and least prefer in 3rd weather which seems to be either most humid(assumed rainy) or windy day.

Registered vs Casual Users Overall :

In [34]:
```
plt.figure(figsize=(7,7))
c=newdata["casual"].sum()
r=newdata["registered"].sum()
l=[c,r]
print(l)
mylabels = ["casual", "registered"]
plt.pie(l,labels = mylabels,autopct ='%.2f')
plt.legend(title = "users type demand count:")
plt.show()
```

```
[392135, 1693341]
```

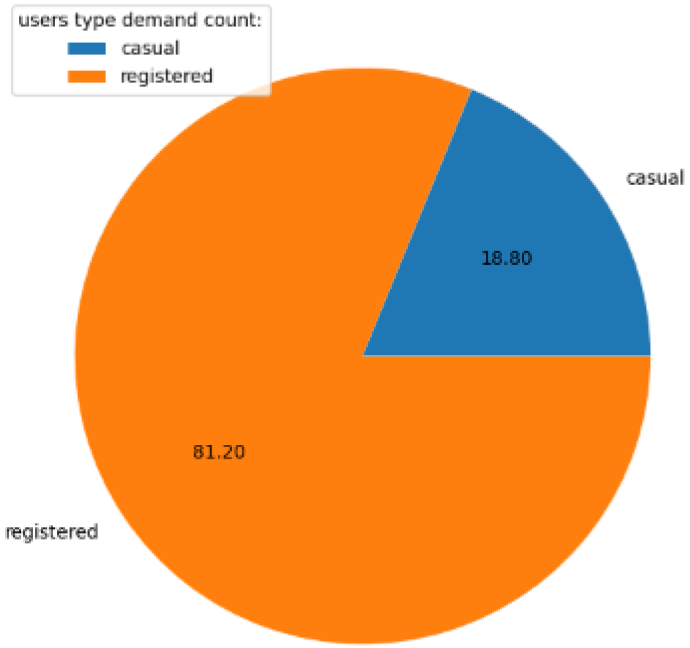


Fig. 3.23: Registered vs Casual Users Overall

- The percentage of casual users is very low as compared to the registered users. To convert these casual users to registered or regular users the company can use advertising or marketing strategies.
- The Orange Colour shows the percentage of registered users that is 81.20%.
- The Blue Colour shows the percentage of casual users that is 18.80%.
- It's a huge difference, to overcome this problem it is suggested to pay more attention towards the casual users and try making them registered users to increase companies profit.

Heatmap :

```
plt.figure(figsize=(12, 7))
sns.heatmap(newdata.corr('pearson') ,annot = True, vmin = -1, vmax = 1)
plt.show()
```

Fig. 3.24: Heatmap code

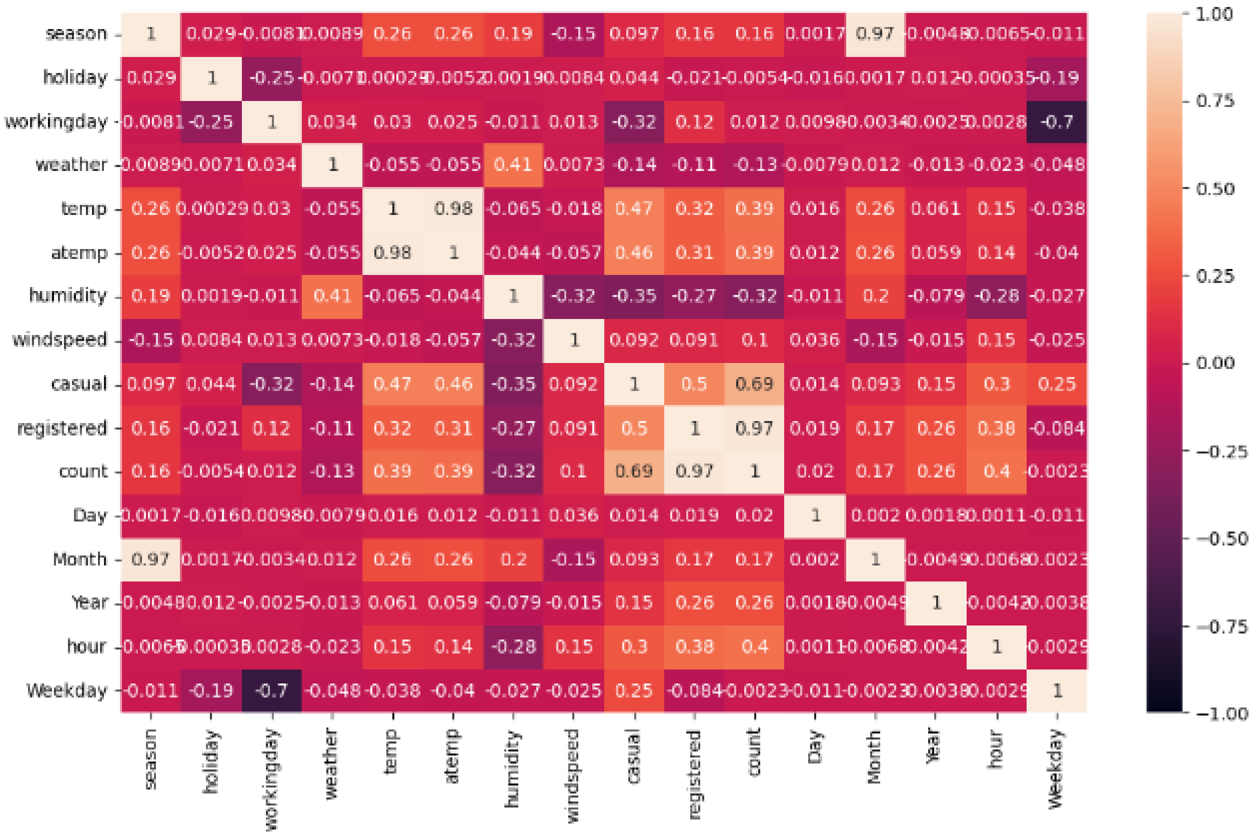


Fig. 3.25: HEATMAP

- ✓ This is the correlation matrix of all attribute in the form of Heat map.
- ✓ The darker the color, the lesser correlation it has.
- ✓ Lighter the colour more the positive correlation exists and darker the colour more negative the correlation is.

CHAPTER 4
CONCLUSION

5.1 Discussion

- The bike rental company needs to work on casual users, as they might not be in majority here but they still would help making profit if turned into registered users.
- The Renting rate is very low in winter season (season-1) and in the initial months of year, so it needs some attention to be paid towards to make more renting possible.
- Within the span of one year the company made a good improvement as 2012 had more renting count than 2011. So if company keeps working on its keep areas it will most probably make more profit and increase in renting count in future.
- The bike renting is also very poor on holidays as compared to the workingdays, to attract more customer on holidays as well company can release some offers or family combos.
- From the hour we can say that registered renting occurs more in 7-9am or 5-7 pm, so this tells more people are working

5.2 Future Work

Future work on our project are as follows:

- Apply Machine Learning.
- Make the machine learning model to predict future bike renting.

REFERENCES

Downloading data set:

https://www.kaggle.com/datasets/marklvl/bike-sharing-dataset

Loading and analyzing

https://pandas.pydata.org/

https://numpy.org/

Pre-processing

https://github.com/siyuanligit/Bike-Sharing-Demand-Kaggle/blob/master/Bike%20Share%20Rental%20Analysis%20Report%20Siyuan%20Li.md

https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/

https://github.com/siyuanligit/Bike-Sharing-Demand-Kaggle/blob/master/Bike%20Share%20Rental%20Analysis%20Report%20Siyuan%20Li.md

https://www.kaggle.com/code/balmeetkaur/rental-bike-insights-predictions

Data Visualization

https://matplotlib.org/tutorials/index.html

https://python-graph-gallery.com/seaborn/

https://www.geeksforgeeks.org/data-visualization-with-python/