Yulu Case study by Dinesh Prabhu DSML2022 Dec

```
#Importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy.stats import ttest_ind,f_oneway
from scipy.stats.contingency import chi2_contingency
```

```
! gdown 1AqzfGofasFBaIhsl0gSZw-vNbgzQesix
```

```
Downloading...
From: https://drive.google.com/uc?id=1AqzfGofasFBaIhsl0gSZw-vNbgzQesix
To: /content/yulu2.txt
100% 648k/648k [00:00<00:00, 5.64MB/s]
```

```
#Creating a data frame
yulu_df=pd.read_csv('yulu2.txt')
yulu_df
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
10881	2012-12-19 19:00:00	4	0	1	1	15.58	19.695	50	26.0027	7	329	336
10882	2012-12-19 20:00:00	4	0	1	1	14.76	17.425	57	15.0013	10	231	241
10883	2012-12-19 21:00:00	4	0	1	1	13.94	15.910	61	15.0013	4	164	168
10884	2012-12-19 22:00:00	4	0	1	1	13.94	17.425	61	6.0032	12	117	129
10885	2012-12-19 23:00:00	4	0	1	1	13.12	16.665	66	8.9981	4	84	88

10886 rows × 12 columns

1. Defining Problem Statement and Analysing the basic metrics

> 1. Perform Exploratory Data analysis
> 2. Apply hypothesis testing methods to understand the factors on which the demand for these shared electric cycles depends
> 3. understand the customers usage season wise and draw insights to improve business

> 1.1 Observations on the Data

```
#Different columns available in the data frame
yulu_df.columns
```

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

```
# Shape of the data frame
yulu_df.shape
```

```
(10886, 12)
```

```
#Data type of each column
yulu_df.dtypes
```

```
datetime       object
season          int64
holiday         int64
workingday      int64
weather         int64
temp          float64
atemp         float64
humidity        int64
windspeed     float64
casual          int64
registered      int64
count           int64
dtype: object
```

```
#information about the each column
yulu_df.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
```

> Conversion of categorical objects into category

```
#since the date time column is time frame,it is converted to date time format

yulu_df["datetime"]=pd.to_datetime(yulu_df["datetime"])

# columns like "Season","Holiday","Working day","Weather" are categotical in nature
yulu_df["season"]=yulu_df["season"].astype("category")
yulu_df["holiday"]=yulu_df["holiday"].astype("category")
yulu_df["workingday"]=yulu_df["workingday"].astype("category")
yulu_df["weather"]=yulu_df["weather"].astype("category")
yulu_df.dtypes
```

```
datetime      datetime64[ns]
season              category
holiday             category
workingday          category
weather             category
temp                 float64
atemp                float64
humidity               int64
windspeed            float64
casual                 int64
registered             int64
count                  int64
dtype: object
```

> Missing value detection

> > Observation: we can conclude that there are no missing values present in the data

```
#finding sum of null values in each column
yulu_df.isna().sum()
```

```
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
```

> Statistical summary of the data

```
#statistical summary of the numerical columns
yulu_df.describe()[["temp","atemp","humidity","windspeed","casual","registered","count"]]
```

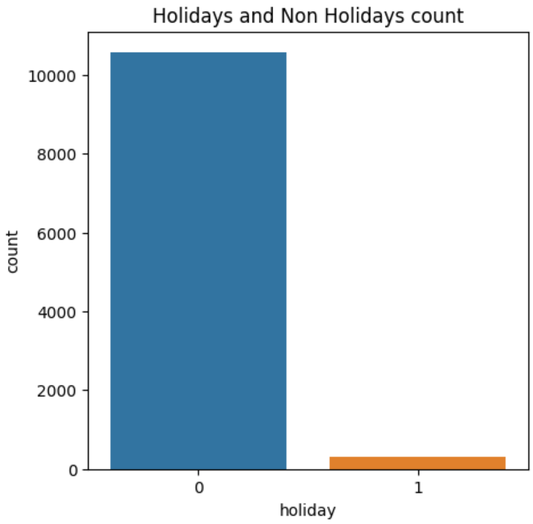
	temp	atemp	humidity	windspeed	casual	registered	count
count	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000	10886.000000
mean	20.23086	23.655084	61.886460	12.799395	36.021955	155.552177	191.574132
std	7.79159	8.474601	19.245033	8.164537	49.960477	151.039033	181.144454
min	0.82000	0.760000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	13.94000	16.665000	47.000000	7.001500	4.000000	36.000000	42.000000
50%	20.50000	24.240000	62.000000	12.998000	17.000000	118.000000	145.000000
75%	26.24000	31.060000	77.000000	16.997900	49.000000	222.000000	284.000000
max	41.00000	45.455000	100.000000	56.996900	367.000000	886.000000	977.000000

1.3 Visual Analysis

> 1. Univariate Analysis

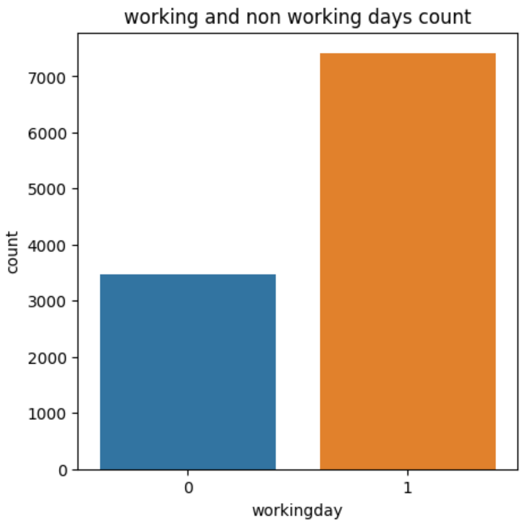
```
plt.figure(figsize=(5,5))
plt.title('Holidays and Non Holidays count')
```

```
sns.countplot(data=yulu_df,x='holiday')
plt.show()
```



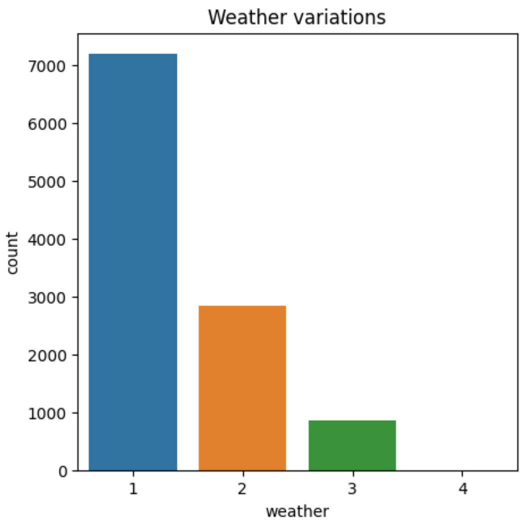
> **Observation**: we can see that During non holiday days the customer are more interested to rent cycles

```
plt.figure(figsize=(5,5))
plt.title('working and non working days count')
sns.countplot(data=yulu_df,x='workingday')
plt.show()
```



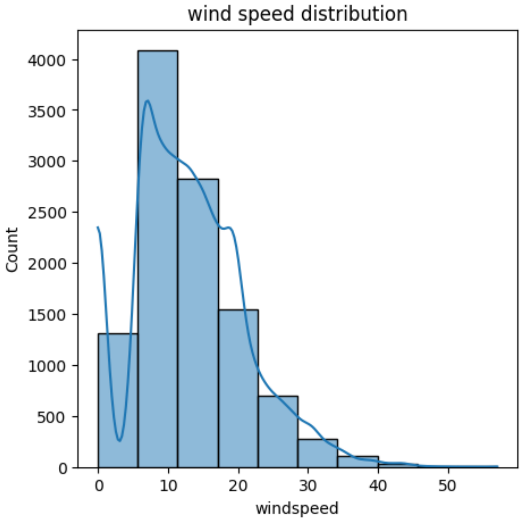
> **Observation** During working days there more usage of Yulu's vehicles

```
plt.figure(figsize=(5,5))
plt.title('Weather variations')
sns.countplot(data=yulu_df,x='weather')
plt.show()
```

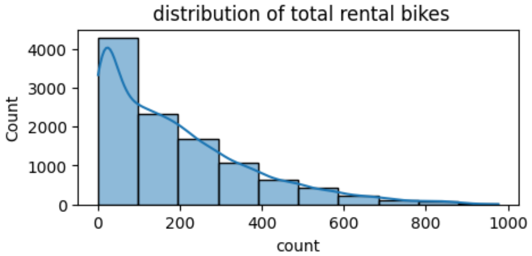


> **Observation**: the usage is more when the weather is Clear, Few clouds, partly cloudy, partly cloud and people tend to not to prefer YULU vehicles when the weather conditions are like Heavy Rain , Ice Pallets , Thunderstorm ,Mist, Snow , Fog

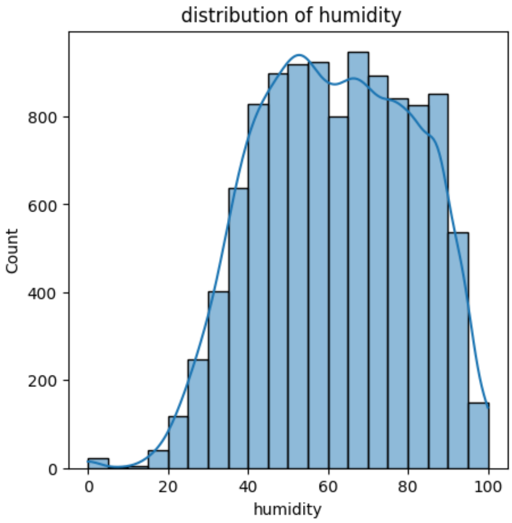
```
plt.figure(figsize=(5,5))
plt.title('wind speed distribution')
sns.histplot(data=yulu_df,x='windspeed',kde=True,bins=10)
plt.show()
```




```
plt.figure(figsize=(5,2))
plt.title(' distribution of total rental bikes')
sns.histplot(data=yulu_df,x='count',kde=True,bins=10)
plt.show()
```

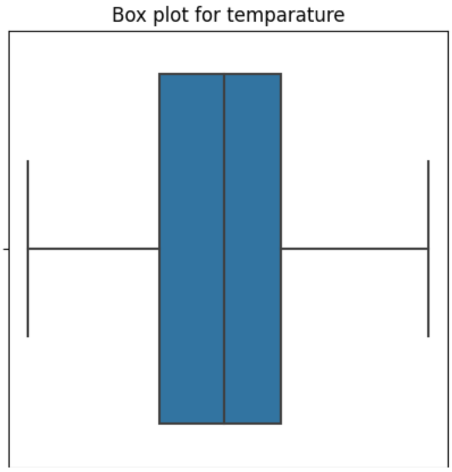



```
plt.figure(figsize=(5,5))
plt.title(' distribution of humidity')
sns.histplot(data=yulu_df,x='humidity',kde=True,bins=20)
plt.show()
```



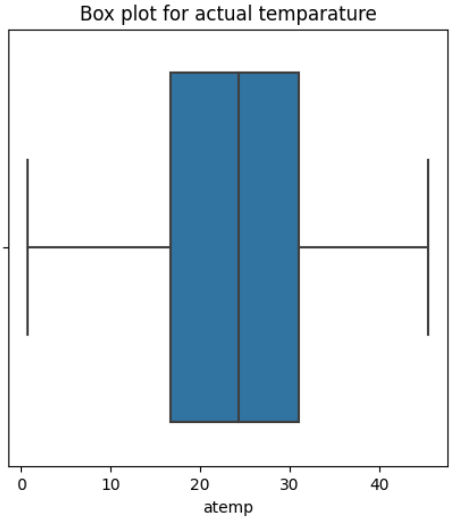
> Outlier detection

```
plt.figure(figsize=(5,5))
plt.subplot()
plt.title("Box plot for temparature")
sns.boxplot(data=yulu_df,x='temp')
plt.show()
```



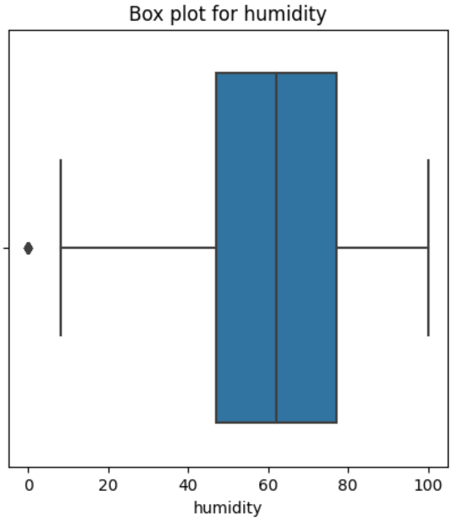
> #Observation: no outliers were detected in temparature column

```
plt.figure(figsize=(5,5))
plt.subplot()
plt.title("Box plot for actual temparature")
sns.boxplot(data=yulu_df,x='atemp')
plt.show()
```



> #Observation: no outliers were detected in actual temparature column

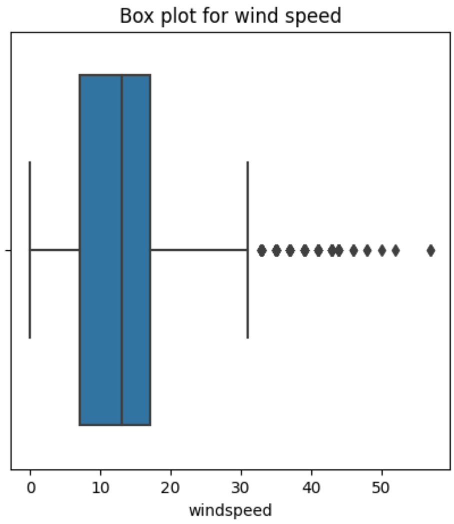
```
plt.figure(figsize=(5,5))
plt.subplot()
plt.title("Box plot for humidity")
sns.boxplot(data=yulu_df,x='humidity')
plt.show()
```



> #Observation: few outliers were detected in humidity column most of them were at the lower bound

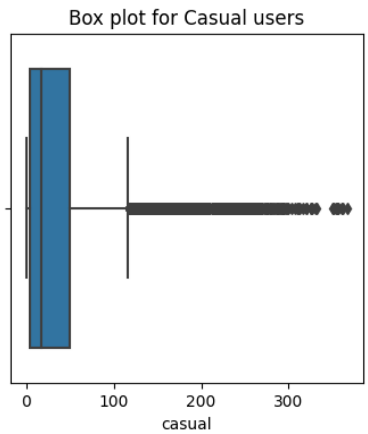
```
plt.figure(figsize=(5,5))
plt.subplot()
```

```
plt.title("Box plot for wind speed")
sns.boxplot(data=yulu_df,x='windspeed')
plt.show()
```



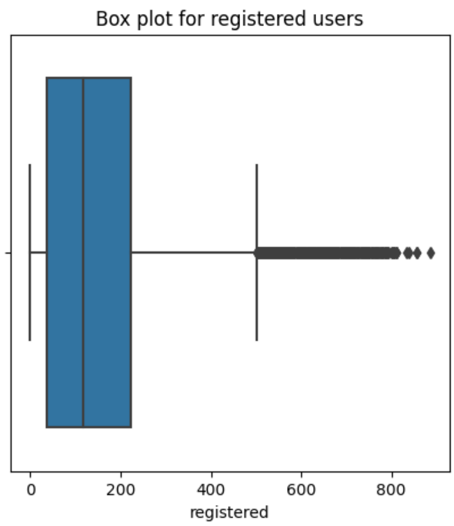
> #Observation: large number of outliers were observed in windspeed column

```
plt.figure(figsize=(4,4))
plt.subplot()
plt.title("Box plot for Casual users")
sns.boxplot(data=yulu_df,x='casual')
plt.show()
```



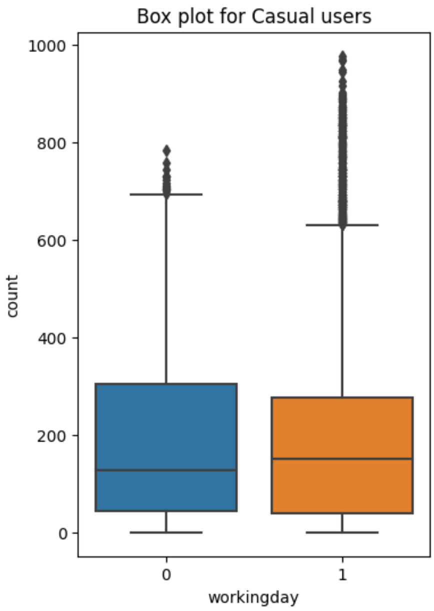
> Very large number of outliers were observed in casual users column

```
plt.figure(figsize=(5,5))
plt.subplot()
plt.title("Box plot for registered users")
sns.boxplot(data=yulu_df,x='registered')
plt.show()
```



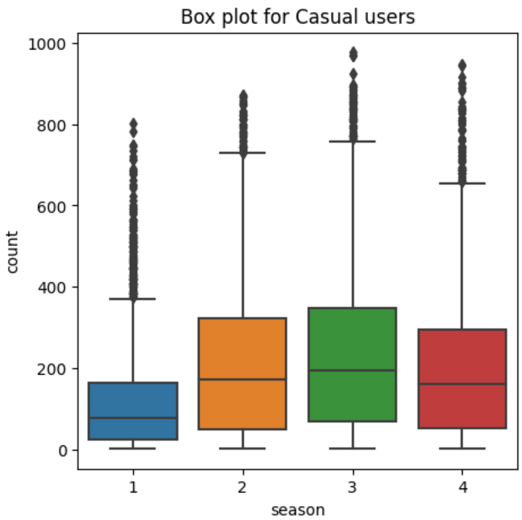
> Bivariate Analysis

```
plt.figure(figsize=(4,6))
plt.title("Box plot for Casual users")
sns.boxplot(data=yulu_df,x='workingday',y='count')
plt.show()
```



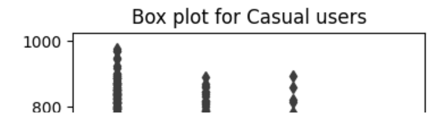
> **Observation** Working day column has more outliers the mean use of Cycles during the working day is more compared to non working

```
plt.figure(figsize=(5,5))
plt.title("Box plot for Casual users")
sns.boxplot(data=yulu_df,x='season',y='count')
plt.show()
```



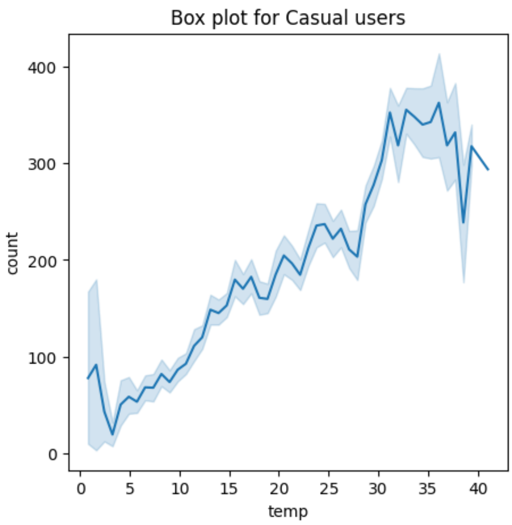
> **Observation** The seasons 2,3,4 have almost same mean count of cycles being rented

```
plt.figure(figsize=(4,4))
plt.title("Box plot for Casual users")
sns.boxplot(data=yulu_df,x='weather',y='count')
plt.show()
```



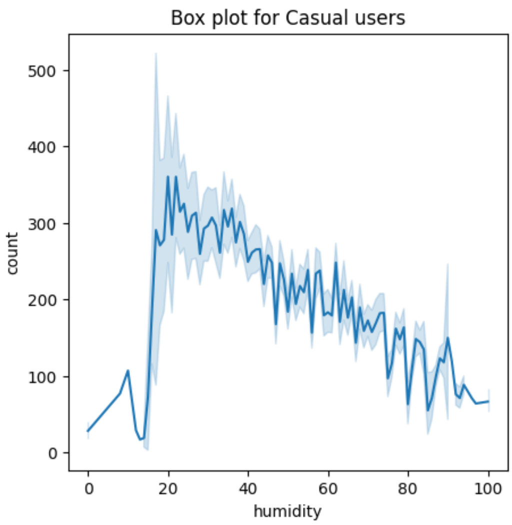
> Users are less likely to book a ride when the temparatures are low

```
plt.figure(figsize=(5,5))
plt.title("Box plot for Casual users")
sns.lineplot(data=yulu_df,x='temp',y='count')
plt.show()
```



> at extreme humid weather conditions user are not likely to book a ride

```
plt.figure(figsize=(5,5))
plt.title("Box plot for Casual users")
sns.lineplot(data=yulu_df,x='humidity',y='count')
plt.show()
```

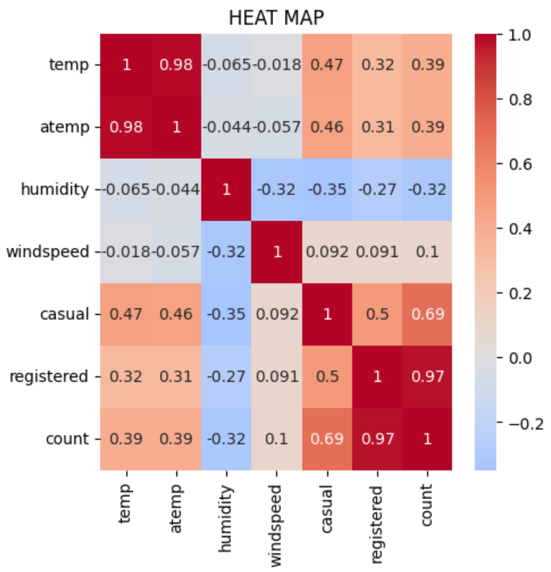


> HEAT MAP

```
df1=yulu_df[["temp","atemp","humidity","windspeed","casual","registered","count"]]
correlation=df1.corr()
correlation
```

	temp	atemp	humidity	windspeed	casual	registered	count
temp	1.000000	0.984948	-0.064949	-0.017852	0.467097	0.318571	0.394454
atemp	0.984948	1.000000	-0.043536	-0.057473	0.462067	0.314635	0.389784
humidity	-0.064949	-0.043536	1.000000	-0.318607	-0.348187	-0.265458	-0.317371
windspeed	-0.017852	-0.057473	-0.318607	1.000000	0.092276	0.091052	0.101369
casual	0.467097	0.462067	-0.348187	0.092276	1.000000	0.497250	0.690414
registered	0.318571	0.314635	-0.265458	0.091052	0.497250	1.000000	0.970948
count	0.394454	0.389784	-0.317371	0.101369	0.690414	0.970948	1.000000

```
plt.figure(figsize=(5,5))
plt.title("HEAT MAP")
sns.heatmap(correlation,cbar=True,annot=True,center=0,cmap="coolwarm")
plt.show()
```



2.Hypothesis Testing

> 1. **2- Sample T-Test** to check if Working Day has an effect on the number of electric cycles rented

```
df_working_day=yulu_df[yulu_df["workingday"]==1]["count"]  # count of rented bikes on working day

df_non_working_day=yulu_df[yulu_df["workingday"]==0]["count"] #count of rented bikes on non working day

mu1=df_working_day.mean()

mu2=df_non_working_day.mean()

print("mean count of rented bikes on working day             mu1:",mu1)

print("mean count of rented bikes on non working day         mu2:",mu2)
```

```
mean count of rented bikes on working day             mu1: 193.01187263896384
mean count of rented bikes on non working day         mu2: 188.50662061024755
```

```
#checking the variances
var1=np.var(df_working_day)
var2=np.var(df_non_working_day)
print("varaince 1 :{} \n varience 2 : {}".format(var1,var2))
print("ratio of variances ",(var1/var2))
```

```
varaince 1 :34040.69710674686
 varience 2 : 30171.346098942427
ratio of variances  1.1282458858519429
```

> **Observation** Since variances are almost same we can consider for 2 sample T test

```
#H0:: mu1=mu2  (there is no difference between the mean count of rented bikes whether its a working day or non working day)
#Ha:: mu1>mu2  (yes there is difference betweeen mean count of rented bikes besed on the working day)

t_statistic,p_value=ttest_ind(df_working_day,df_non_working_day,alternative="greater")
print("p_value  ",p_value)

aplha=0.05  #Significance level

if p_value<0.05:
  print("reject H0: working days has effect on cycles being rented")
else:
  print("Fail to reject H0 : working day has no effect on the cycles being rented")
```

```
p_value   0.11322402113180674
Fail to reject H0 : working day has no effect on the cycles being rented
```

> 2.2. **ANNOVA** to check if No. of cycles rented is similar or different in different 1. weather 2. season

```
#H0:: Number of cycles rented is similar in different weather and season
#Ha:: Number of cycles rented is not similar in different in different weather and season
w1=yulu_df[yulu_df['weather']==1]['count'].values
w2=yulu_df[yulu_df['weather']==2]['count'].values
w3=yulu_df[yulu_df['weather']==3]['count'].values
w4=yulu_df[yulu_df['weather']==4]['count'].values


s1=w1=yulu_df[yulu_df['season']==1]['count'].values
```

```
s2=w1=yulu_df[yulu_df['season']==2]['count'].values
s3=w1=yulu_df[yulu_df['season']==3]['count'].values
s4=w1=yulu_df[yulu_df['season']==4]['count'].values

stats,p_val=f_oneway(w1,w2,w3,w4,s1,s2,s3,s4)
print('stats ',stats)
print('p_value',p_val)

aplha=0.05 #Significance level

if p_val<0.05:
  print("reject H0")
else:
  print("Fail to reject H0 ")
```

```
stats  127.34076932175545
p_value 2.0293852335555225e-183
reject H0
```

> **Observation::** the number of cycles rented is not similar in different weather and season

> 2.3 **Chi-square** test to check if Weather is dependent on the season

```
#H0:: Weather is independent of the season
#Ha:: Weather is not independent of the season
data_table=pd.crosstab(yulu_df['season'],yulu_df['weather'])
data_table
```

weather	1	2	3	4
season				
1	1759	715	211	1
2	1801	708	224	0
3	1930	604	199	0
4	1702	807	225	0

```
stats,pval,dof,expected_freq=chi2_contingency(data_table)
print('stats ',stats)
print('P_value ',pval)
print('degrees of freedom ',dof)
print('Expected frequency ',expected_freq)
```

```
stats  49.158655596893624
P_value  1.549925073686492e-07
degrees of freedom  9
Expected frequency  [[1.77454639e+03 6.99258130e+02 2.11948742e+02 2.46738931e-01]
 [1.80559765e+03 7.11493845e+02 2.15657450e+02 2.51056403e-01]
 [1.80559765e+03 7.11493845e+02 2.15657450e+02 2.51056403e-01]
 [1.80625831e+03 7.11754180e+02 2.15736359e+02 2.51148264e-01]]
```

```
alpha=0.05
if p_val<0.05:
  print("reject H0")
else:
  print("Fail to reject H0 ")
```

```
reject H0
```

> **Observation::** on rejecting null hypothesis we conclude weather is dependent on the season

3. Which Variables are significant in predicting the demand for shared electric cycles in the indian market

1. **Weather and Season:** Using Analysis of variances ANOVA it is concluded that the number of cycles rented is not similar in different weather and season
2. **Working day:** to check if Working Day has an effect on the number of electric cycles rented a 2 sample T test is conducted and the results shows that We don't have the sufficient evidence to say that working day has effect on the number of cycles being rented.
3. **Weather**: With the help of Chi_square test it is observed that the weather is dependent on the season
4. from the line plot for Humidity vs the number of cycles being rented we can recommend that yulu can place less bikes in the stock to be rented.
5. Whenever the windspeed is greater than 35 or in thunderstorms, company should have less bikes in stock to be rented.
6. Whenever temprature is less than 10 or in very cold days, company should have less bikes.

0s completed at 9:57 PM