Yulu_Data

November 27, 2024

[ ]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chisquare
from scipy.stats import chi2_contingency
```

[ ]:
```python
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/
↪original/bike_sharing.csv?1642089089 -O yulu.csv
```

```
--2024-11-27 12:34:53--  https://d2beiqkhq929f0.cloudfront.net/public_assets/ass
ets/000/001/428/original/bike_sharing.csv?1642089089
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)…
3.167.84.9, 3.167.84.28, 3.167.84.148, …
Connecting to d2beiqkhq929f0.cloudfront.net
(d2beiqkhq929f0.cloudfront.net)|3.167.84.9|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 648353 (633K) [text/plain]
Saving to: 'yulu.csv'

 yulu.csv              0%[                    ]       0  --.-KB/s
yulu.csv            100%[===================>] 633.16K  --.-KB/s    in 0.02s

2024-11-27 12:34:54 (37.3 MB/s) - 'yulu.csv' saved [648353/648353]
```

[ ]:
```python
df=pd.read_csv("yulu.csv")
df
```

[ ]:
```
                  datetime  season  holiday  workingday  weather   temp  \
0      2011-01-01 00:00:00       1        0           0        1   9.84
1      2011-01-01 01:00:00       1        0           0        1   9.02
2      2011-01-01 02:00:00       1        0           0        1   9.02
3      2011-01-01 03:00:00       1        0           0        1   9.84
4      2011-01-01 04:00:00       1        0           0        1   9.84
…                      …     …       …           …        …     …
10881  2012-12-19 19:00:00       4        0           1        1  15.58
10882  2012-12-19 20:00:00       4        0           1        1  14.76
```

```
10883  2012-12-19 21:00:00       4        0           1        1  13.94
10884  2012-12-19 22:00:00       4        0           1        1  13.94
10885  2012-12-19 23:00:00       4        0           1        1  13.12

        atemp  humidity  windspeed  casual  registered  count
0      14.395        81     0.0000       3          13     16
1      13.635        80     0.0000       8          32     40
2      13.635        80     0.0000       5          27     32
3      14.395        75     0.0000       3          10     13
4      14.395        75     0.0000       0           1      1
…         …         …         …       …        …     …
10881  19.695        50    26.0027       7         329    336
10882  17.425        57    15.0013      10         231    241
10883  15.910        61    15.0013       4         164    168
10884  17.425        61     6.0032      12         117    129
10885  16.665        66     8.9981       4          84     88

[10886 rows x 12 columns]
```

[ ]:
```
df[df.holiday!=0]
```

[ ]:
```
                  datetime  season  holiday  workingday  weather   temp  \
372    2011-01-17 00:00:00       1        1           0        2   8.20
373    2011-01-17 01:00:00       1        1           0        2   8.20
374    2011-01-17 02:00:00       1        1           0        2   7.38
375    2011-01-17 03:00:00       1        1           0        2   7.38
376    2011-01-17 04:00:00       1        1           0        2   7.38
…                      …     …       …          …       …     …
10257  2012-11-12 19:00:00       4        1           0        1  22.14
10258  2012-11-12 20:00:00       4        1           0        2  21.32
10259  2012-11-12 21:00:00       4        1           0        3  22.14
10260  2012-11-12 22:00:00       4        1           0        1  21.32
10261  2012-11-12 23:00:00       4        1           0        2  22.14

        atemp  humidity  windspeed  casual  registered  count
372     9.850        47    15.0013       1          16     17
373     9.850        44    12.9980       1          15     16
374     8.335        43    16.9979       0           8      8
375     9.090        43    12.9980       0           2      2
376     9.850        43     8.9981       1           2      3
…         …         …         …       …        …     …
10257  25.760        73    19.0012      30         323    353
10258  25.000        77    19.0012      31         273    304
10259  25.760        73    15.0013      10         145    155
10260  25.000        77    16.9979      12         100    112
10261  25.760        77    15.0013       1          62     63
```

```
[311 rows x 12 columns]
```

[ ]:
```
df.shape
```

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

[ ]:
```
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
```

[ ]:
```
df.describe()
```

[ ]:
```
             season       holiday    workingday       weather         temp  \
count  10886.000000  10886.000000  10886.000000  10886.000000  10886.00000   
mean       2.506614      0.028569      0.680875      1.418427     20.23086   
std        1.116174      0.166599      0.466159      0.633839      7.79159   
min        1.000000      0.000000      0.000000      1.000000      0.82000   
25%        2.000000      0.000000      0.000000      1.000000     13.94000   
50%        3.000000      0.000000      1.000000      1.000000     20.50000   
75%        4.000000      0.000000      1.000000      2.000000     26.24000   
max        4.000000      1.000000      1.000000      4.000000     41.00000   

              atemp      humidity     windspeed        casual    registered  \
count  10886.000000  10886.000000  10886.000000  10886.000000  10886.000000   
mean      23.655084     61.886460     12.799395     36.021955    155.552177   
std        8.474601     19.245033      8.164537     49.960477    151.039033   
min        0.760000      0.000000      0.000000      0.000000      0.000000   
25%       16.665000     47.000000      7.001500      4.000000     36.000000   
50%       24.240000     62.000000     12.998000     17.000000    118.000000   
```

```
75%        31.060000     77.000000      16.997900     49.000000     222.000000
max        45.455000    100.000000      56.996900    367.000000     886.000000

              count
count  10886.000000
mean     191.574132
std      181.144454
min        1.000000
25%       42.000000
50%      145.000000
75%      284.000000
max      977.000000
```

[ ]:
```
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
```

Identify and remove duplicate records.

[ ]:
```
df.duplicated().sum()
# No dupliactes found
```

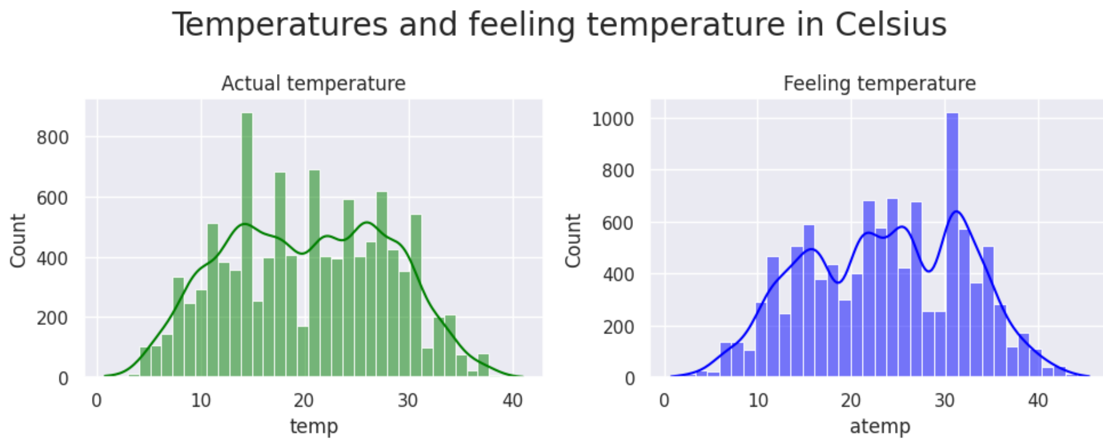
[ ]:
```
0
```

[ ]:
```
df['Date']=pd.to_datetime(df['datetime']).dt.date
df['Time']=pd.to_datetime(df['datetime']).dt.time
```

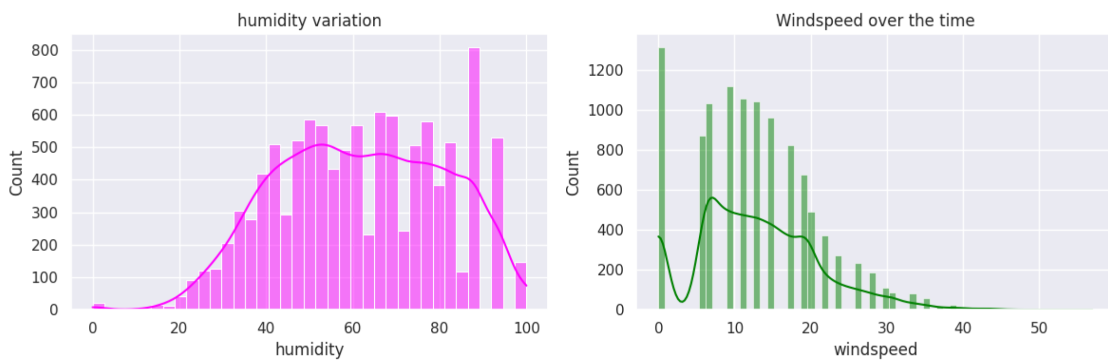
Analyze the distribution of Numerical & Categorical variables, separately

[ ]:
```
plt.figure(figsize=(10,4)).suptitle("Temperatures and feeling temperature in␣
  ↪Celsius",fontsize=20)
plt.subplot(1,2,1)
sns.histplot(df.temp,kde=True,color='green')
plt.title("Actual temperature")
plt.subplot(1,2,2)
sns.histplot(df.atemp,kde=True,color='blue')
```

```
plt.title("Feeling temperature")
plt.tight_layout()
plt.show()
```

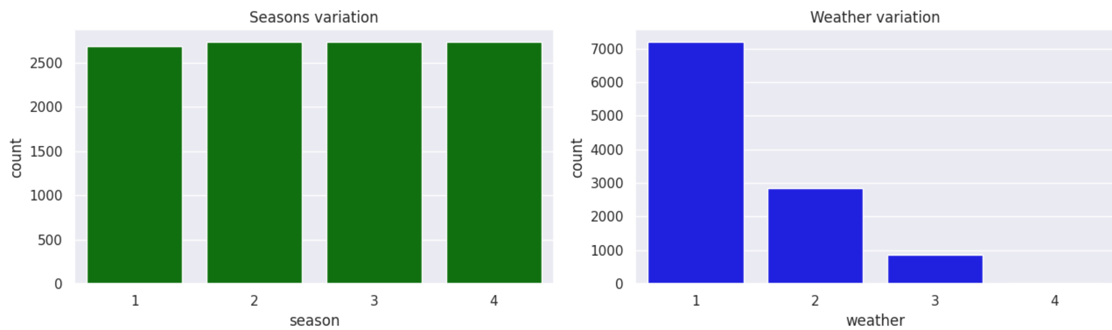


[ ]:
```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
sns.histplot(df.humidity,kde=True,color='magenta')
plt.title("humidity variation")
plt.subplot(1,2,2)
sns.histplot(df.windspeed,kde=True,color='green')
plt.title("Windspeed over the time")
plt.tight_layout()
plt.show()
```

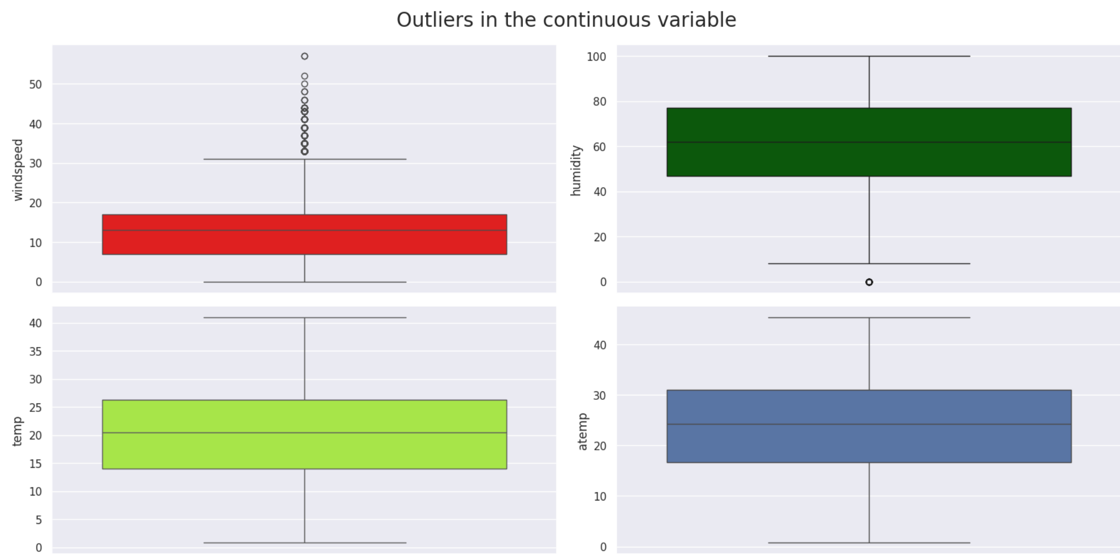


[ ]:
```
plt.figure(figsize=(13,4))
plt.subplot(1,2,1)
sns.countplot(df,x='season',color='green')
plt.title("Seasons variation")
```

```
plt.subplot(1,2,2)
sns.countplot(df,x='weather',color='blue')
plt.title("Weather variation")
plt.tight_layout()
plt.show()
```



[ ]:
```
plt.figure(figsize=(16,8)).suptitle("Outliers in the continuous␣
↪variable",fontsize=20)
plt.subplot(2,2,1)
sns.boxplot(data=df,y="windspeed",color='red')
plt.subplot(2,2,2)
sns.boxplot(data=df,y="humidity",color='darkgreen')
plt.subplot(2,2,3)
sns.boxplot(data=df,y="temp",color='greenyellow')
plt.subplot(2,2,4)
sns.boxplot(data=df,y="atemp")
plt.tight_layout()
plt.show()
```



Check for Outliers and deal with them accordingly.

[ ]:
```
upper_limit,lower_limit=np.percentile(df.windspeed,95),np.percentile(df.
 ↪windspeed,5)
Clip_windspeed=df['windspeed'].clip(lower_limit,upper_limit)
df=df[df.windspeed.isin(Clip_windspeed)]
```

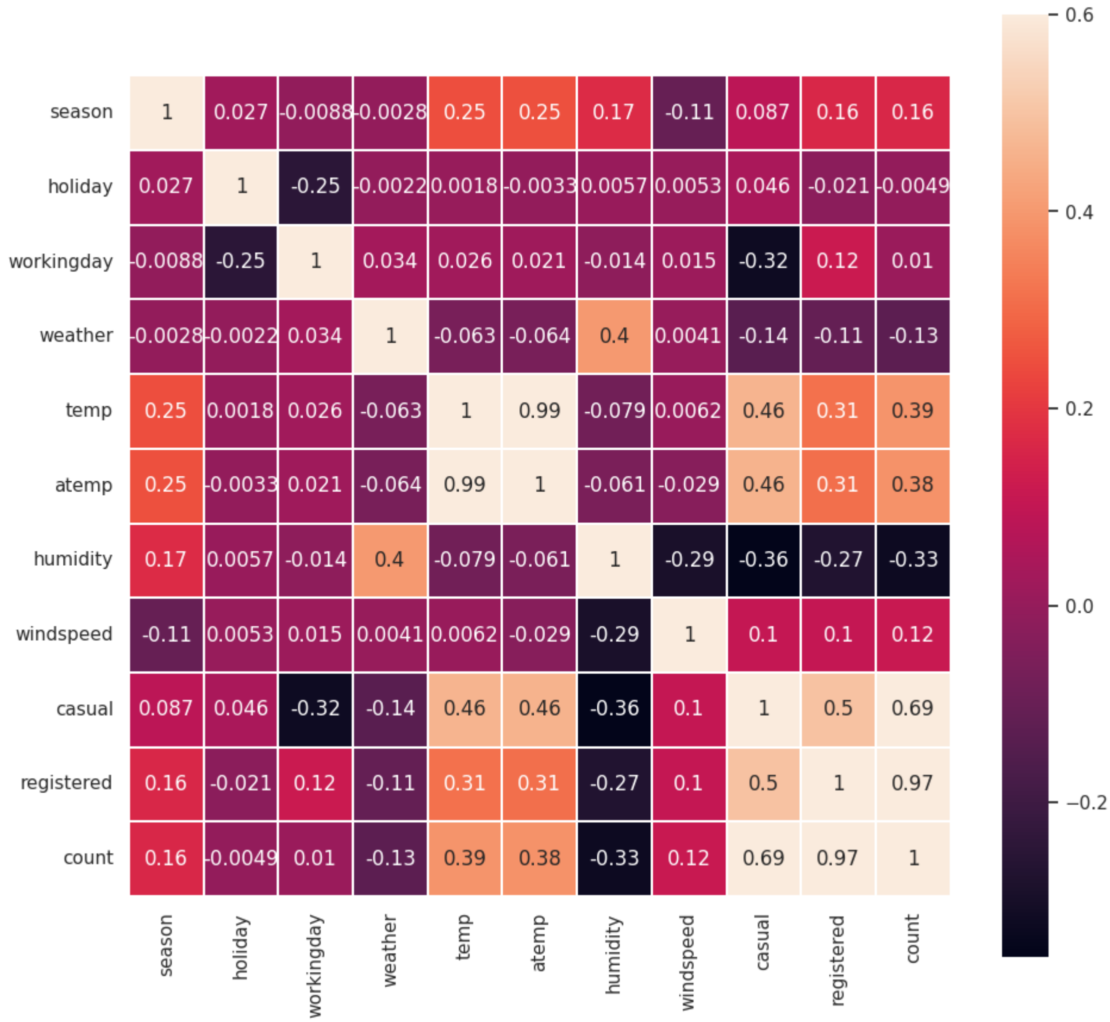
[ ]:
```
df.columns
```

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

#2. Try establishing a Relationship between the Dependent and Independent Variables.

[ ]:
```
df_check=df.loc[:,~df.columns.isin (['datetime','Date', 'Time'])]
```

[ ]:
```
df_check=df.loc[:,~df.columns.isin (['datetime','Date', 'Time'])]
sns.set(font_scale=1.0)
plt.figure(figsize=(10,9))
correlation_values = df_check.corr(method='pearson')
sns.heatmap(correlation_values,linewidths=0.01,annot=True,square=True,vmax=0.6)
plt.tight_layout()
```



#3.Check if there any significant difference between the no. of bike rides on Weekdays and Weekends?

[ ]:
```
print("Total number of rides including both types")
print(df.groupby('workingday')['count'].sum())
print('*'*40)
print("Total number of casual rides")
print(df.groupby('workingday')['casual'].sum())
print('*'*40)
print("Total number of registerd rides")
print(df.groupby('workingday')['registered'].sum())
```

```
Total number of rides including both types
workingday
0      629581
1     1369531
```

```
Name: count, dtype: int64
****************************************
Total number of casual rides
workingday
0    197995
1    177451
Name: casual, dtype: int64
****************************************
Total number of registerd rides
workingday
0     431586
1    1192080
Name: registered, dtype: int64
```

[ ]:
```
#Null Hypothesis (HO):No significant difference between the no. of bike rides␣
↪on Weekdays and Weekends
#Alternate Hypothesis(Ha): Significant difference between the no. of bike rides␣
↪on Weekdays and Weekends
from scipy.stats import ttest_ind
for i in ['casual','registered','count']:
  t_stats,p_value=ttest_ind(df[df.workingday==0][i],df[df.
↪workingday==1][i],alternative='two-sided')
  alpha=0.05
  if p_value < alpha:
    print(f"Significant difference between the no. of bike rides on Weekdays␣
↪and Weekends for {i} rides ",p_value)
  else:
    print(f"No significant difference between the no. of bike rides on Weekdays␣
↪and Weekends for {i} rides",p_value)
```

```
Significant difference between the no. of bike rides on Weekdays and Weekends
for casual rides  8.054442556670831e-249
Significant difference between the no. of bike rides on Weekdays and Weekends
for registered rides  7.112909853962498e-34
No significant difference between the no. of bike rides on Weekdays and Weekends
for count rides 0.28732772342282314
```

RECOMMENDATIONS: 1. There is a signifiacant difference incase of registered rides for weekdays and weekends. It is advised to give discount offers on weekdays to registered customers to encourage registered rides. 2. There is a signifiacant difference incase of registered rides for weekdays and weekends.And it is more on weekends. So, to enhance profit yulu can charge extra during weekends to causal rider.

#Check if there any significant difference between the no. of bike rides on hoildays and not holidays?

[ ]:
```
print("Total number of rides including both types")
print(df.groupby('holiday')['count'].sum())
print('*'*40)
```

```python
print("Total number of casual rides")
print(df.groupby('holiday')['casual'].sum())
print('*'*40)
print("Total number of registerd rides")
print(df.groupby('holiday')['registered'].sum())
```

```
Total number of rides including both types
holiday
0    1944280
1      54832
Name: count, dtype: int64
****************************************
Total number of casual rides
holiday
0    360906
1     14540
Name: casual, dtype: int64
****************************************
Total number of registerd rides
holiday
0    1583374
1      40292
Name: registered, dtype: int64
```

[ ]:
```python
#Null Hypothesis (HO):No significant difference between the no. of bike rides␣
↪on holiday and not holidays
#Alternate Hypothesis(Ha): Significant difference between the no. of bike rides␣
↪on holiday and not holidays
from scipy.stats import ttest_ind
for i in ['casual','registered','count']:
  t_stats,p_value=ttest_ind(df[df.workingday==0][i],df[df.
↪workingday==1][i],alternative='two-sided')
  alpha=0.05
  if p_value < alpha:
    print(f"Significant difference between the no. of bike rides on holiday and␣
↪not holidays for {i} rides ",p_value)
  else:
    print(f"No significant difference between the no. of bike rides on holiday␣
↪and not holidays for {i} rides",p_value)
```

```
Significant difference between the no. of bike rides on holiday and not holidays
for casual rides  8.054442556670831e-249
Significant difference between the no. of bike rides on holiday and not holidays
for registered rides  7.112909853962498e-34
No significant difference between the no. of bike rides on holiday and not
holidays for count rides 0.28732772342282314
```

RECOMMENDATIONS: 1. There is a significant difference incase of registered rides for holiday and not holidays days. Its is advised to charge extra to registered riders on holidays to increases profit. 2.There is a significant difference incase of casual rides for holiday and not holidays days. Its is advised to charge extra to casual riders on holidays to increases profit.

#4. Check if the demand of bicycles on rent is the same for different Weather conditions?

Here we are going to use ANOVA test as we have one numerical column with more than two categories

Check assumptions of the ANOVA test 1. Normality 2. Equal variance among the groups

[ ]:
```
#Let's use Shapiro-Wilk test for normality check
# Significance level, alpha=.05
# H0: Data is Gaussian
#Ha: Data is not Gaussian

alpha= 0.05

from scipy.stats import shapiro
for i in ['casual','registered','count']:
  print(f"Let's check normality of {i} rides")
  for j in [1,2,3,4]:
    print(f"Let's check normality of {i} rides for weather_type{j}")
    i_weather_j=df[df['weather']==j][i]
    test_stats,p_value=shapiro(i_weather_j)
    if p_value> alpha:
      print(f"{i} rides distribution follows normal distribution for␣
↪weather_type {j} with p_value of {p_value}",)
    else:
      print(f"{i} rides distribution does not follow normal distribution  for␣
↪weather_type {j} with p_values of {p_value}")
      print(f"{i} rides distribution does not follow normal distribution")
      break
  print("*"*40)
```

```
Let's check normality of casual rides
Let's check normality of casual rides for weather_type1
casual rides distribution does not follow normal distribution  for weather_type
1 with p_values of 1.0471423862766511e-74
casual rides distribution does not follow normal distribution
****************************************
Let's check normality of registered rides
Let's check normality of registered rides for weather_type1
registered rides distribution does not follow normal distribution  for
weather_type 1 with p_values of 3.4564709026092123e-60
registered rides distribution does not follow normal distribution
****************************************
Let's check normality of count rides
```

```
Let's check normality of count rides for weather_type1
count rides distribution does not follow normal distribution  for weather_type 1
with p_values of 8.015369519865055e-57
count rides distribution does not follow normal distribution
****************************************

/usr/local/lib/python3.10/dist-packages/scipy/stats/_axis_nan_policy.py:531:
UserWarning: scipy.stats.shapiro: For N > 5000, computed p-value may not be
accurate. Current N is 6890.
  res = hypotest_fun_out(*samples, **kwds)
```

1 Let's use Levene test for variance assumption check

[ ]:
```python
from scipy.stats import levene
for i in ['casual','registered','count']:
  print(f"Let's check vaiance equality of {i} rides")
  ␣
 ↪L_stats,p_value=levene(df[df['weather']==1][i],df[df['weather']==2][i],df[df['weather']==3]
  if p_value < alpha:
    print("Variances significantly differ among groups with p_value ",p_value)
  else:
    print("Variance are relatively equal across th groups")
  print("*"*40)
```

```
Let's check vaiance equality of casual rides
Variances significantly differ among groups with p_value  6.591133827128098e-38
****************************************
Let's check vaiance equality of registered rides
Variances significantly differ among groups with p_value  4.40626563888364e-22
****************************************
Let's check vaiance equality of count rides
Variances significantly differ among groups with p_value  5.961769865041247e-35
****************************************
```

2 Now we are going to use Kruskal test as assumption are not satisfied

[ ]:
```python
# Null hypothesis (HO): All the groups have same median
# Alternate hypothesis (Ha): Atleast one of the group have differenet median
from scipy.stats import kruskal
for i in ['casual','registered','count']:
   print("*"*40)
   print(f"Here we are going to find effect of weather on {i} rides")
   ␣
 ↪stats,p_value=kruskal(df[df['weather']==1][i],df[df['weather']==2][i],df[df['weather']==3][
   if p_value < alpha:
```

```
    print("Reject NULL hypothesis")
    print(f"Atleast one of the weather condition for {i} rides has different␣
↪median with p_value of",p_value)
  else:
    print("Fail to reject null hypothesis")
    print("Demand of bicycles on rent is the same for different Weather␣
↪conditions")
```

```
****************************************
Here we are going to find effect of weather on casual rides
Reject NULL hypothesis
Atleast one of the weather condition for casual rides has different median with
p_value of 2.6346014529419155e-58
****************************************
Here we are going to find effect of weather on registered rides
Reject NULL hypothesis
Atleast one of the weather condition for registered rides has different median
with p_value of 1.8002017918706419e-34
****************************************
Here we are going to find effect of weather on count rides
Reject NULL hypothesis
Atleast one of the weather condition for count rides has different median with
p_value of 8.491841818937477e-41
```

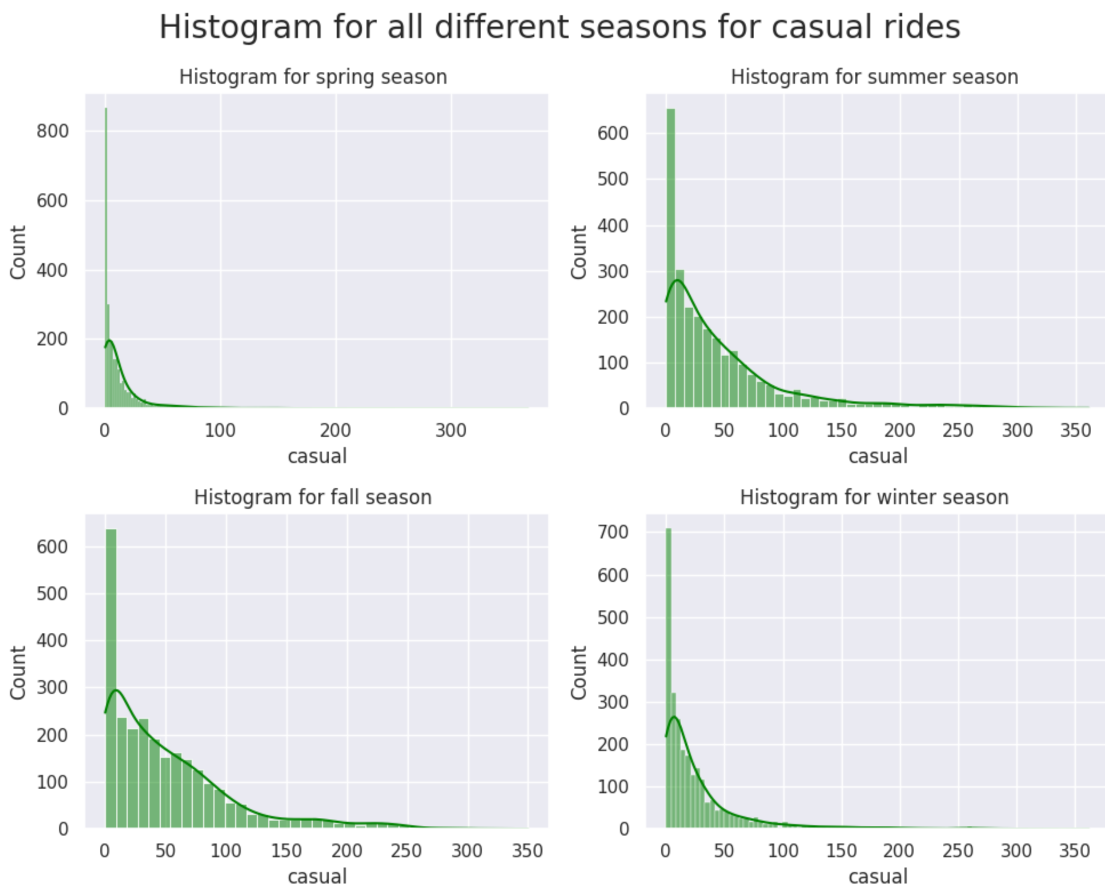
#5. Check if the demand of bicycles on rent is the same for different Seasons?

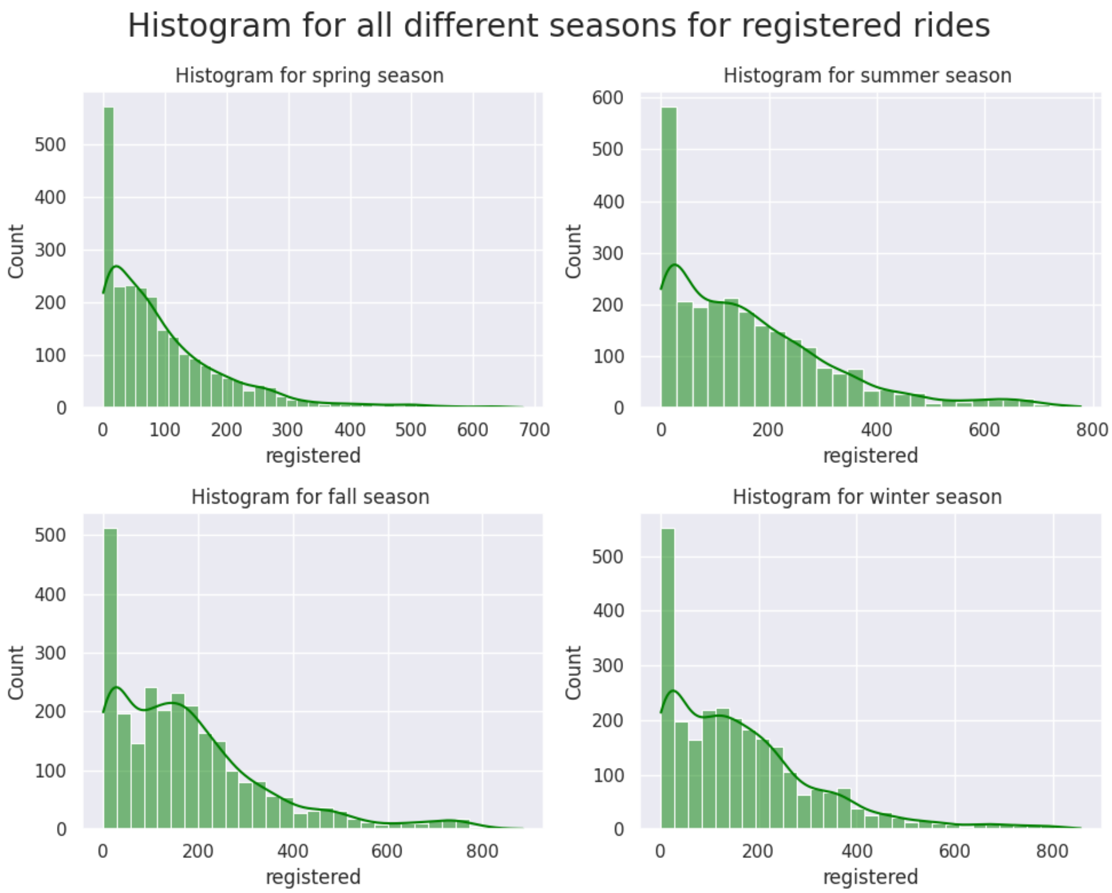
Let's check assumptions of ANOVA are met first before using ANOVA test

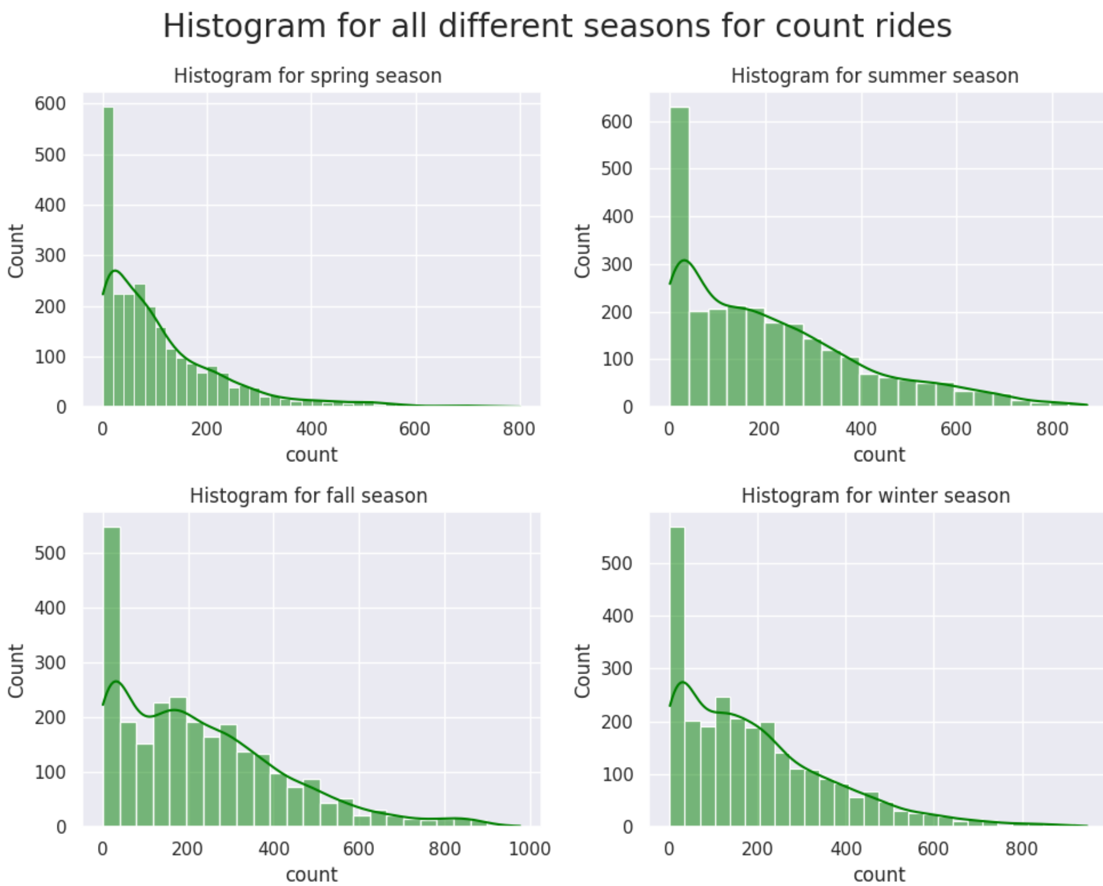
[ ]:
```
#Let's use histogram for normality check
# Significance level, alpha=.05
# H0: Data is Gaussian
#Ha: Data is not Gaussian


alpha= 0.05
for i in ['casual','registered','count']:
  plt.figure(figsize=(10,8)).suptitle(f"Histogram for all different seasons for␣
↪{i} rides",fontsize=20)
  plt.subplot(2,2,1)
  sns.histplot(df[df['season']==1][i],kde=True,color='green')
  plt.title("Histogram for spring season ")
  plt.subplot(2,2,2)
  sns.histplot(df[df['season']==2][i],kde=True,color='green')
  plt.title("Histogram for summer season ")
  plt.subplot(2,2,3)
  sns.histplot(df[df['season']==3][i],kde=True,color='green')
  plt.title("Histogram for fall season ")
  plt.subplot(2,2,4)
```

```
sns.histplot(df[df['season']==4][i],kde=True,color='green')
plt.title("Histogram for winter season ")
plt.tight_layout()
plt.show()
```




Histogram for all different seasons for registered rides
Histogram for spring season
Histogram for summer season
Histogram for fall season
Histogram for winter season
Count
registered
0
100
200
300
400
500
600
700
800



Conclusion: *Distribution for type of season for any type of rider i.e casual, registered or both is not normal distribution.*

Let's use Levene test for variance assumption check

```
[ ]: from scipy.stats import levene
for i in ['casual','registered','count']:
  print(f"Let's check vaiance equality of {i} rides")
 ␣
 ↪L_stats,p_value=levene(df[df['season']==1][i],df[df['season']==2][i],df[df['season']==3][i]
  if p_value < alpha:
    print("Variances significantly differ among groups with p_value ",p_value)
  else:
    print("Variance are relatively equal across th groups")
  print("*"*40)
```

```
Let's check vaiance equality of casual rides
Variances significantly differ among groups with p_value
1.0741420840015213e-146
****************************************
```

```
Let's check vaiance equality of registered rides
Variances significantly differ among groups with p_value  2.7418470796882453e-70
****************************************
Let's check vaiance equality of count rides
Variances significantly differ among groups with p_value
3.6972971685846006e-111
****************************************
```

#Now we are going to use Kruskal test as assumption are not satisfied

[ ]:
```
# Null hypothesis (H0): All the groups have same median
# Alternate hypothesis (Ha): Atleast one of the group have differenet median
from scipy.stats import kruskal
for i in ['casual','registered','count']:
   print("*"*40)
   print(f"Here we are going to find effect of season on {i} rides")
  ␣
 ↪stats,p_value=kruskal(df[df['season']==1][i],df[df['season']==2][i],df[df['season']==3][i],d
   if p_value < alpha:
    print("Reject NULL hypothesis")
    print(f"Atleast one of the season condition for {i} rides has different␣
 ↪median with p_value of",p_value)
   else:
    print("Fail to reject null hypothesis")
    print("Demand of bicycles on rent is the same for different season␣
 ↪conditions")
```

```
****************************************
Here we are going to find effect of season on casual rides
Reject NULL hypothesis
Atleast one of the season condition for casual rides has different median with
p_value of 0.0
****************************************
Here we are going to find effect of season on registered rides
Reject NULL hypothesis
Atleast one of the season condition for registered rides has different median
with p_value of 1.3168302005736572e-110
****************************************
Here we are going to find effect of season on count rides
Reject NULL hypothesis
Atleast one of the season condition for count rides has different median with
p_value of 2.504949726738821e-142
```

[ ]:
```
# Let's have T_Test for getting more insight
from scipy.stats import ttest_ind
alpha=0.05
for i in ['casual','registered','count']:
  print(f"Checking significant statistical difference for {i} rides")
```

```
for j in [1,2,3]:
  for k in range(j+1,5):
    print(f"Checking significant statistical difference for {i} rides incase␣
↪of season type {j} and {k}")
    t_test,p_value=ttest_ind(df[df['season']==j][i],df[df['season']==k][i])
    if p_value < alpha:
      print(f"Significant statistical difference for {i} rides incase of␣
↪season type {j} and {k} with p_values",p_value)
    else:
       print(f"No Significant statistical difference for {i} rides incase of␣
↪season type {j} and {k} with p_values",p_value)
    print("*"*40)
print("#"*60)
```

```
Checking significant statistical difference for casual rides
Checking significant statistical difference for casual rides incase of season
type 1 and 2
Significant statistical difference for casual rides incase of season type 1 and
2 with p_values 8.306625680390875e-121
****************************************
Checking significant statistical difference for casual rides incase of season
type 1 and 3
Significant statistical difference for casual rides incase of season type 1 and
3 with p_values 2.3531458018991817e-172
****************************************
Checking significant statistical difference for casual rides incase of season
type 1 and 4
Significant statistical difference for casual rides incase of season type 1 and
4 with p_values 6.66248786496448e-33
****************************************
Checking significant statistical difference for casual rides incase of season
type 2 and 3
Significant statistical difference for casual rides incase of season type 2 and
3 with p_values 0.0027330532679905256
****************************************
Checking significant statistical difference for casual rides incase of season
type 2 and 4
Significant statistical difference for casual rides incase of season type 2 and
4 with p_values 1.6407361002377453e-43
****************************************
Checking significant statistical difference for casual rides incase of season
type 3 and 4
Significant statistical difference for casual rides incase of season type 3 and
4 with p_values 1.3529660226163975e-71
****************************************
############################################################
Checking significant statistical difference for registered rides
```

```
Checking significant statistical difference for registered rides incase of
season type 1 and 2
Significant statistical difference for registered rides incase of season type 1
and 2 with p_values 1.591392930367561e-68
****************************************
Checking significant statistical difference for registered rides incase of
season type 1 and 3
Significant statistical difference for registered rides incase of season type 1
and 3 with p_values 4.84796554804436e-93
****************************************
Checking significant statistical difference for registered rides incase of
season type 1 and 4
Significant statistical difference for registered rides incase of season type 1
and 4 with p_values 1.6524564889538648e-75
****************************************
Checking significant statistical difference for registered rides incase of
season type 2 and 3
Significant statistical difference for registered rides incase of season type 2
and 3 with p_values 0.0008865124709351012
****************************************
Checking significant statistical difference for registered rides incase of
season type 2 and 4
No Significant statistical difference for registered rides incase of season type
2 and 4 with p_values 0.4529102094908416
****************************************
Checking significant statistical difference for registered rides incase of
season type 3 and 4
Significant statistical difference for registered rides incase of season type 3
and 4 with p_values 0.008976872817910072
****************************************
############################################################
Checking significant statistical difference for count rides
Checking significant statistical difference for count rides incase of season
type 1 and 2
Significant statistical difference for count rides incase of season type 1 and 2
with p_values 4.976246947423306e-99
****************************************
Checking significant statistical difference for count rides incase of season
type 1 and 3
Significant statistical difference for count rides incase of season type 1 and 3
with p_values 2.4514592313735486e-134
****************************************
Checking significant statistical difference for count rides incase of season
type 1 and 4
Significant statistical difference for count rides incase of season type 1 and 4
with p_values 5.823111494957243e-79
****************************************
Checking significant statistical difference for count rides incase of season
```

```
type 2 and 3
Significant statistical difference for count rides incase of season type 2 and 3
with p_values 0.00031438034706589795
****************************************
Checking significant statistical difference for count rides incase of season
type 2 and 4
Significant statistical difference for count rides incase of season type 2 and 4
with p_values 0.001513117859035398
****************************************
Checking significant statistical difference for count rides incase of season
type 3 and 4
Significant statistical difference for count rides incase of season type 3 and 4
with p_values 5.522034904970142e-12
****************************************
############################################################
```

[ ]:
```
for i in ['casual','registered','count']:
  print(f"Number of rides rides for {i} rides")
  x=df.groupby('season')[i].sum()
  print(f"Number of rides for {i} rides incase of different season type",x)
```

```
Number of rides rides for casual rides
Number of rides for casual rides incase of different season type season
1     38309
2    123492
3    139129
4     74516
Name: casual, dtype: int64
Number of rides rides for registered rides
Number of rides for registered rides incase of different season type season
1    247176
2    435940
3    486544
4    454006
Name: registered, dtype: int64
Number of rides rides for count rides
Number of rides for count rides incase of different season type season
1    285485
2    559432
3    625673
4    528522
Name: count, dtype: int64
```

#Recommendation: 1. Since only there is one case where we have no significant statistical difference for registered rides incase of season type 2 and 4 with p_values 0.4529102094908416, for rest all types of season have significant impact on the rides. 2. With help of T_test for pairs we can say that causal rides are more for season type 2 and 3.Hence to increase profit we can charges extra in season type 2 and 3.

Hence we can say distribution for causal, registered

[ ]:
```
from scipy.stats import f_oneway
for i in ['casual','registered','count']:
  i_weather_1=df[df['weather']==1][i]
  i_weather_2=df[df['weather']==2][i]
  i_weather_3=df[df['weather']==3][i]
  i_weather_4=df[df['weather']==4][i]
  f_stats,p_value=f_oneway(i_weather_1,i_weather_2,i_weather_3,i_weather_4)
  print("*"*40)
  print(f"Here we are going to find effect of weather on {i} rides")
  if p_value < alpha:
    print("Reject null hypothesis")
    print(f"Atleast one of the weather condition for {i} rides has different␣
↪mean with p_value of",p_value)
  else:
    print("Fail to reject null hypothesis")
    print("Demand of bicycles on rent is the same for different Weather␣
↪conditions?")
```

```
****************************************
Here we are going to find effect of weather on casual rides
Reject null hypothesis
Atleast one of the weather condition for casual rides has different mean with
p_value of 5.2747318871773536e-42
****************************************
Here we are going to find effect of weather on registered rides
Reject null hypothesis
Atleast one of the weather condition for registered rides has different mean
with p_value of 1.4323637632653132e-29
****************************************
Here we are going to find effect of weather on count rides
Reject null hypothesis
Atleast one of the weather condition for count rides has different mean with
p_value of 1.2035642095025712e-39
```

#6. Check if the Weather conditions are significantly different during different Seasons?

[ ]:
```
weather_season=pd.crosstab(index=df.weather,columns=df.season)
weather_season
```

[ ]:

season	1	2	3	4
weather				
1	1590	1721	1910	1669
2	688	688	595	792
3	201	209	181	214
4	1	0	0	0

[ ]:
```python
from scipy.stats import chi2_contingency
chi_stats,p_value,df,expected_freq=chi2_contingency(weather_season)
print("chi_stats",chi_stats)
print("p_value",p_value)
print("degree of freedom",df)
print("expected frequency",expected_freq)
```

```
chi_stats 55.29873806102515
p_value 1.0682692671386608e-08
degree of freedom 9
expected frequency [[1.63373171e+03 1.72464098e+03 1.76943685e+03
1.76219046e+03]
 [6.55152500e+02 6.91608567e+02 7.09572426e+02 7.06666507e+02]
 [1.90878669e+02 2.01500143e+02 2.06733913e+02 2.05887274e+02]
 [2.37116359e-01 2.50310737e-01 2.56812315e-01 2.55760589e-01]]
```

[ ]:
```python
alpha = 0.05
if p_value < alpha:
  print("Reject null hypothesis")
  print("Weather and season are  not significanlty different")
else:
  print("Fail to reject null hypothesis")
  print("Weather and season are independent")
```

```
Reject null hypothesis
Weather and season are  not significanlty different
```