Yulu : Hypothesis Testing

About Yulu

Yulu is India’s leading micro-mobility service provider, which offers unique vehicles for the daily commute. Starting off as a mission to eliminate traffic congestion in India, Yulu provides the safest commute solution through a user-friendly mobile app to enable shared, solo and sustainable commuting.

Yulu zones are located at all the appropriate locations (including metro stations, bus stands, office spaces, residential areas, corporate offices, etc) to make those first and last miles smooth, affordable, and convenient!

Yulu has recently suffered considerable dips in its revenues. They have contracted a consulting company to understand the factors on which the demand for these shared electric cycles depends. Specifically, they want to understand the factors affecting the demand for these shared electric cycles in the Indian market.

How you can help here?

The company wants to know:

Which variables are significant in predicting the demand for shared electric cycles in the Indian market? How well those variables describe the electric cycle demands Dataset:

Column Profiling:

datetime: datetime

season: season (1: spring, 2: summer, 3: fall, 4: winter)

holiday: whether day is a holiday or not

workingday: if day is neither weekend nor holiday is 1, otherwise is 0.

weather:

1: Clear, Few clouds, partly cloudy, partly cloudy

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

temp: temperature in Celsius

atemp: feeling temperature in Celsius

humidity: humidity

windspeed: wind speed

casual: count of casual users

registered: count of registered users

count: count of total rental bikes including both casual and registered

Defining Problem Statement:

Based on the count values provided for each hour w.r.t to biked rented we have to do the analysis from which variable it is getting affected.

And have to prove it.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from math import log

from scipy.stats import ttest_1samp, ttest_ind, ttest_ind_from_stats,f_oneway,chisquare,chi2_contingency
from scipy.stats import norm, t,f,chi2, kstest,shapiro,levene
```

```
from statsmodels.graphics.gofplots import qqplot
from scipy.special import boxcox
```

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?1642089089"
```

```
--2023-05-21 12:56:50--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/428/original/bike_sharing.csv?164208908
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.84.194.17, 99.84.194.96, 99.84.194.174, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99.84.194.17|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 648353 (633K) [text/plain]
Saving to: 'bike_sharing.csv?1642089089'

bike_sharing.csv?16 100%[===================>] 633.16K  --.-KB/s    in 0.03s

2023-05-21 12:56:50 (21.9 MB/s) - 'bike_sharing.csv?1642089089' saved [648353/648353]
```

```
yulu = pd.read_csv("bike_sharing.csv?1642089089")
yulu.head(10)
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000
4	2011-01-01	1	0	0	1	9.84	14.395	75	0.0000

Analysing Basic Metrics

```
yulu.shape
```

```
(10886, 12)
```

```
yulu.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
```

Changing the datatype of following attributes to appropriate data type

- datetime - datetime
- season - object
- holiday - object
- workingday - object
- weather - object

```
yulu['datetime'] = pd.to_datetime(yulu['datetime'])
cat_cols = ['season', 'holiday', 'workingday','weather']
num_cols= ['temp','atemp','humidity','windspeed','casual','registered','count']
```

```
for col in cat_cols:
  yulu[col] = yulu[col].astype('object')
```

```
yulu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   datetime        10886 non-null  datetime64[ns]
 1   season          10886 non-null  object
 2   holiday         10886 non-null  object
 3   workingday      10886 non-null  object
 4   weather         10886 non-null  object
 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
 12  hour            10886 non-null  int64
 13  rented_duration  10431 non-null  category
dtypes: category(1), datetime64[ns](1), float64(3), int64(5), object(4)
memory usage: 1.1+ MB
```

```
#Range of numerical attributes
for i in num_cols:
  print(f"Range of {i} attribute(numerical): min-{min(yulu[i])},max-{max(yulu[i])}")
  print()

#Date Range for given data
print(f"Date Range for given data is from {min(yulu.datetime)} to {max(yulu.datetime)} ")
print()
print(f"Number of days data given is {max(yulu.datetime)-min(yulu.datetime)} ")
```

```
Range of temp attribute(numerical): min-0.82,max-41.0

Range of atemp attribute(numerical): min-0.76,max-45.455

Range of humidity attribute(numerical): min-0,max-100

Range of windspeed attribute(numerical): min-0.0,max-56.9969

Range of casual attribute(numerical): min-0,max-367

Range of registered attribute(numerical): min-0,max-886

Range of count attribute(numerical): min-1,max-977

Date Range for given data is from 2011-01-01 00:00:00 to 2012-12-19 23:00:00

Number of days data given is 718 days 23:00:00
```

Descriptive Statistics of Dataset

```
yulu.describe()
```

	temp	atemp	humidity	windspeed	casual	registere
count	10886.00000	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000
mean	20.23086	23.655084	61.886460	12.799395	36.021955	155.55217
std	7.79159	8.474601	19.245033	8.164537	49.960477	151.03903
min	0.82000	0.760000	0.000000	0.000000	0.000000	0.00000
25%	13.94000	16.665000	47.000000	7.001500	4.000000	36.00000
50%	20.50000	24.240000	62.000000	12.998000	17.000000	118.00000
75%	26.24000	31.060000	77.000000	16.997900	49.000000	222.00000
max	41.00000	45.455000	100.000000	56.996900	367.000000	886.00000

Observations on Numerical columns:

- Columns casual, registered, count might have outliers as mean and median have very much difference.
- temp, atemp, humidity, windspeed columns have mean and median almost similar values. Standard Deviation is also less. So Less number of outliers are expected in these columns.

```
yulu.describe(include='object')
```

	season	holiday	workingday	weather
count	10886	10886	10886	10886
unique	4	2	2	4
top	4	0	1	1
freq	2734	10575	7412	7192

```
yulu["season"].value_counts()
```

```
4    2734
2    2733
3    2733
1    2686
Name: season, dtype: int64
```

```
yulu["holiday"].value_counts()
```

```
0    10575
1      311
Name: holiday, dtype: int64
```

```
yulu["workingday"].value_counts()
```

```
1    7412
0    3474
Name: workingday, dtype: int64
```

```
yulu["weather"].value_counts()
```

```
1    7192
2    2834
3     859
4       1
Name: weather, dtype: int64
```

```
yulu.groupby('season')['count'].describe()
```

```
yulu.groupby('workingday')['count'].describe()
```

workingday	count	mean	std	min	25%	50%	75%	max
0	3474.0	188.506621	173.724015	1.0	44.0	128.0	304.0	783.0
1	7412.0	193.011873	184.513659	1.0	41.0	151.0	277.0	977.0

```
yulu.groupby('weather')['count'].describe()
```

weather	count	mean	std	min	25%	50%	75%	max
1	7192.0	205.236791	187.959566	1.0	48.0	161.0	305.0	977.0
2	2834.0	178.955540	168.366413	1.0	41.0	134.0	264.0	890.0
3	859.0	118.846333	138.581297	1.0	23.0	71.0	161.0	891.0
4	1.0	164.000000	NaN	164.0	164.0	164.0	164.0	164.0

Observations on categorical columns:

- Standard deviation is very high w.r.t weather, workingday and season. so more no.of outliers can be found.
- Season 4 winter has high usage of yulu bikes.
- During holiday = 0 (no holiday) has high usage of yulu bikes.

```
yulu.isnull().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
```

Observations:

Interestingly, There are no missing values in any of the columns. So no need to handle the missing values.

(Imputation of null values is not required)

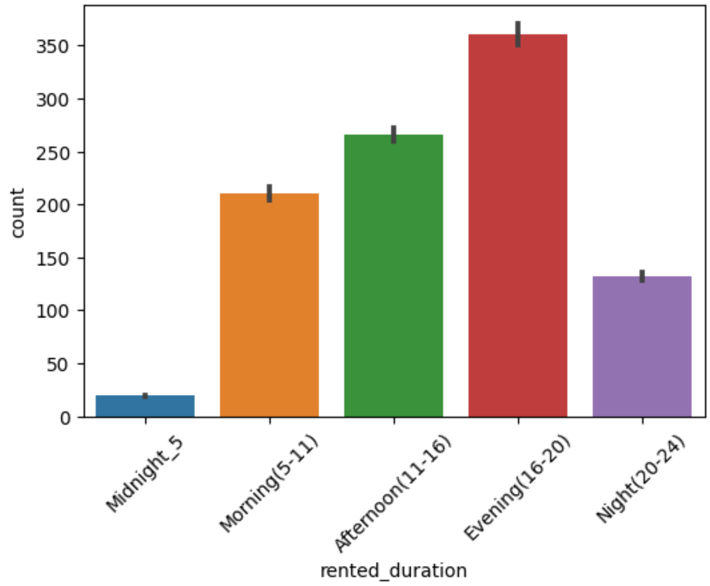
```
yulu['datetime'].min(), yulu['datetime'].max()
```

```
(Timestamp('2011-01-01 00:00:00'), Timestamp('2012-12-19 23:00:00'))
```

```
yulu["hour"] = yulu["datetime"].dt.hour
```

```
points = [0,5,11,16,20,24]
data_labels = ["Midnight_5","Morning(5-11)","Afternoon(11-16)","Evening(16-20)","Night(20-24)"]
yulu["rented_duration"]=pd.cut(yulu["hour"],bins=points,labels=data_labels)
```

```
plt.figure(figsize=(6,4))
sns.barplot(data=yulu,x="rented_duration",y="count")
plt.xticks(rotation= 45)
plt.show()
```

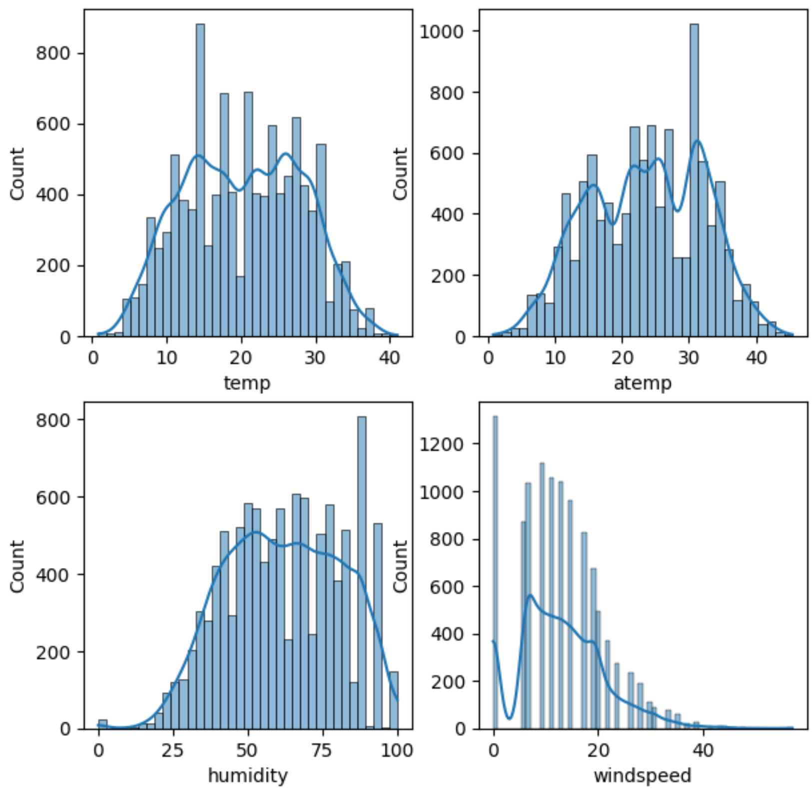


- More no.of Bikes are being rented between 4 PM to 8 PM compared to other timings

Uni-Variant Analysis

```
fig,ax = plt.subplots(2,2,figsize=(7,7))

sns.histplot(data = yulu, x="temp",ax = ax[0,0], kde=True )
sns.histplot(data = yulu, x="atemp",ax = ax[0,1], kde=True )
sns.histplot(data = yulu, x="humidity",ax = ax[1,0], kde=True )
sns.histplot(data = yulu, x="windspeed",ax = ax[1,1], kde=True )
```

```
<Axes: xlabel='windspeed', ylabel='Count'>
```



Insights:

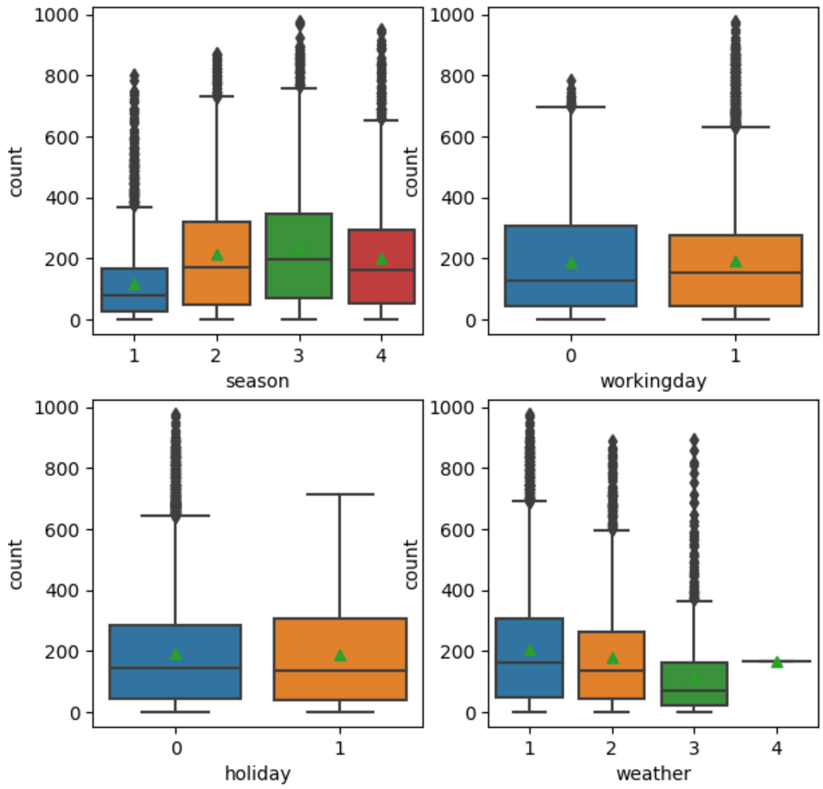
1) Temp, atemp and humidity looks like they follows the Normal Distribution

2) indspeed follows the binomial distribution

Bi-Variant Analysis

```
fig,ax = plt.subplots(2,2,figsize=(7,7))
sns.boxplot(data = yulu, x="season" ,y="count", ax = ax[0,0], showmeans=True)
sns.boxplot(data = yulu, x="workingday" ,y="count", ax = ax[0,1],showmeans=True)
sns.boxplot(data = yulu, x="holiday" ,y="count", ax = ax[1,0],showmeans=True)
sns.boxplot(data = yulu, x="weather" ,y="count", ax = ax[1,1],showmeans=True)
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Insights:

1) More no.of bikes are rented in season fall and summer compared to other seasons.

2) More bikes are rented when there is a holiday.

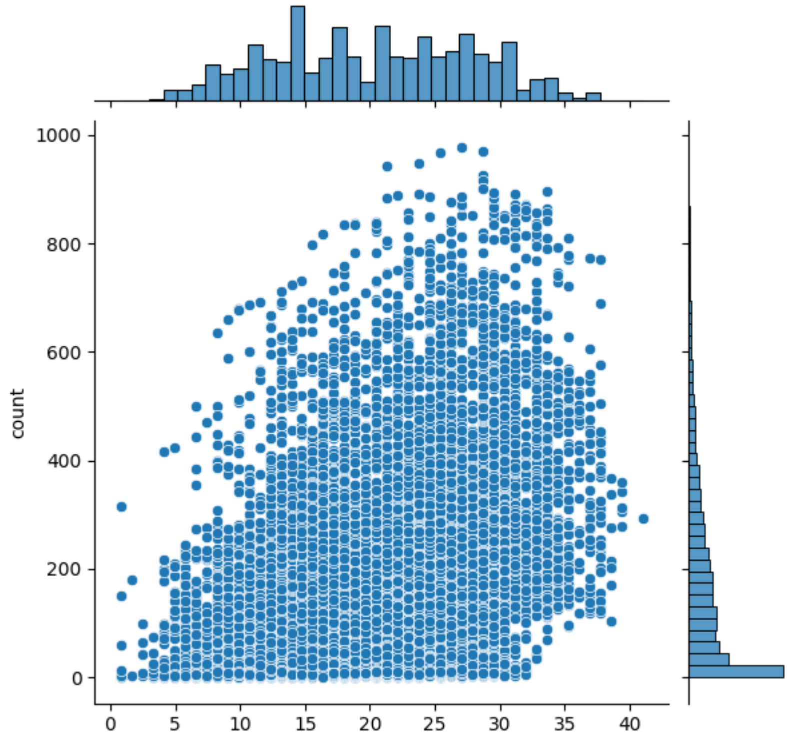
3) Bikes rented with respect to holiday is almost equal and very slight difference observed (i.e slightly more bikes rented when no holiday (0))

4) When it comes to weather, most of the bikes are rented when its 1: Clear, Few clouds, partly cloudy, partly cloudy

and very least no.of bikes are being rented when weather is 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

```
plt.figure(figsize=(5,5))
sns.jointplot(data=yulu,x='temp',y='count')
```

```
<seaborn.axisgrid.JointGrid at 0x7f3b2fc0ebc0>
<Figure size 500x500 with 0 Axes>
```



- When temparture is below 5 Celcius bikes rented is very low.

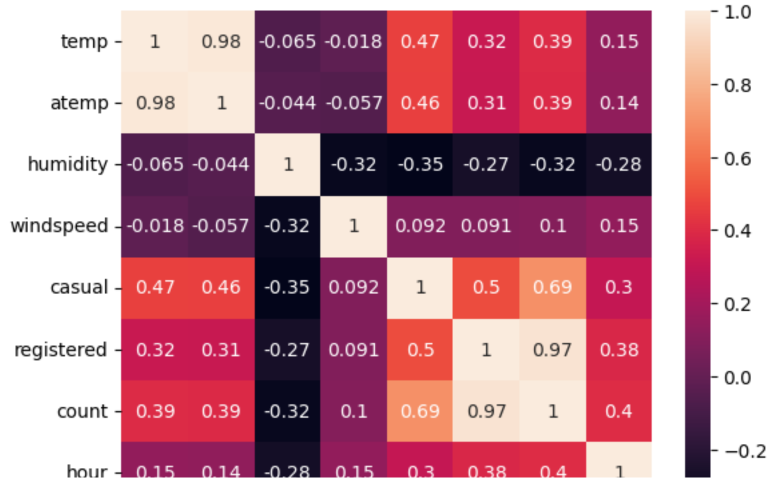
```
yulu.corr()[['casual','registered','count']]
```

```
<ipython-input-26-770405f32b1c>:1: FutureWarning: The default value of numeric_only i
  yulu.corr()[['casual','registered','count']]
```

	casual	registered	count
temp	0.467097	0.318571	0.394454
atemp	0.462067	0.314635	0.389784
humidity	-0.348187	-0.265458	-0.317371
windspeed	0.092276	0.091052	0.101369
casual	1.000000	0.497250	0.690414
registered	0.497250	1.000000	0.970948
count	0.690414	0.970948	1.000000
hour	0.302045	0.380540	0.400601

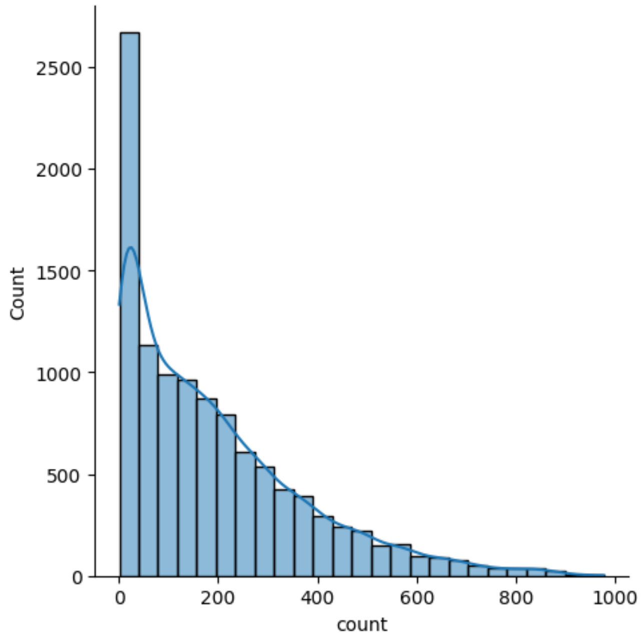
```
sns.heatmap(yulu.corr(),annot=True)
```

```
<ipython-input-27-d44051c2c58f>:1: FutureWarning: The default value of numeric_only i
  sns.heatmap(yulu.corr(),annot=True)
<Axes: >
```



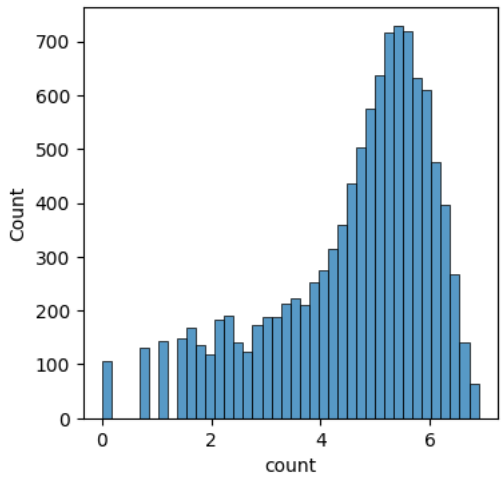

```
plt.figure(figsize=(2,2))
sns.displot(x='count', data=yulu, bins=25, kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f3b204eb760>
<Figure size 200x200 with 0 Axes>
```




```
plt.figure(figsize=(4,4))
log_dist = np.log(yulu["count"])
sns.histplot(log_dist)
```

```
<Axes: xlabel='count', ylabel='Count'>
```



- Data is right-skewed as seen from the figure.
- Bike rented count somewhat looks like Log Normal Distrinution

Hypothesis Testing - 1 (T test)

Q) Working Day has effect on number of electric cycles rented

Step 1: Define null and alternative hypothesis

Null Hypothesis (H0): Working Day has no effect on number of electric cycles rented

Alternate Hypothesis (H1): Working Day has effect on number of electric cycles rented.

$H0 : \mu1 = \mu2$

$Ha: \mu1 != \mu2$

Step 2: Set a significance level (alpha)

Significance level (alpha): 0.05

T-Test Assumption:

1) sample size should be grater than 30 (True)

2) data is collected from a representative, randomly selected portion of the total population(True)

We will use the 2-Sample T-Test to test the hypothesis defined above

Assumptions Test

```
yulu[yulu['workingday']==0]['count'].mean()
```

```
188.50662061024755
```

```
yulu[yulu['workingday']==1]['count'].mean()
```

```
193.01187263896384
```

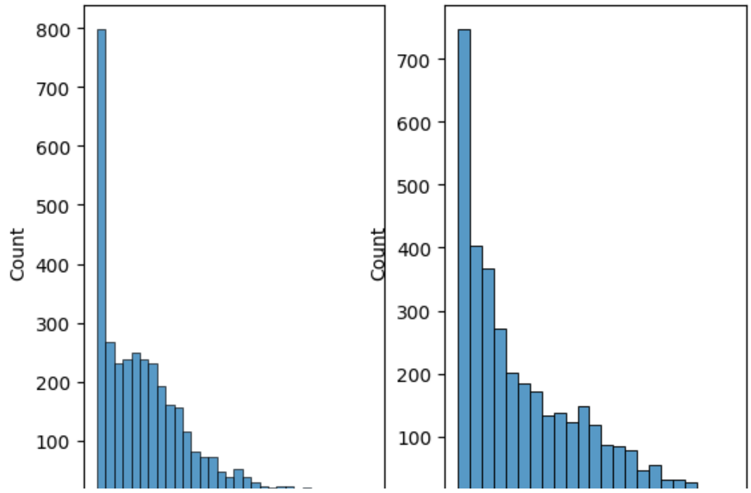
```
yulu.groupby('workingday')['count'].describe()
```

workingday	count	mean	std	min	25%	50%	75%	max
0	3474.0	188.506621	173.724015	1.0	44.0	128.0	304.0	783.0
1	7412.0	193.011873	184.513659	1.0	41.0	151.0	277.0	977.0

```
workingday = yulu[yulu['workingday']==1]['count'].sample(3474)
non_workingday = yulu[yulu['workingday']==0]['count'].sample(3474)
```

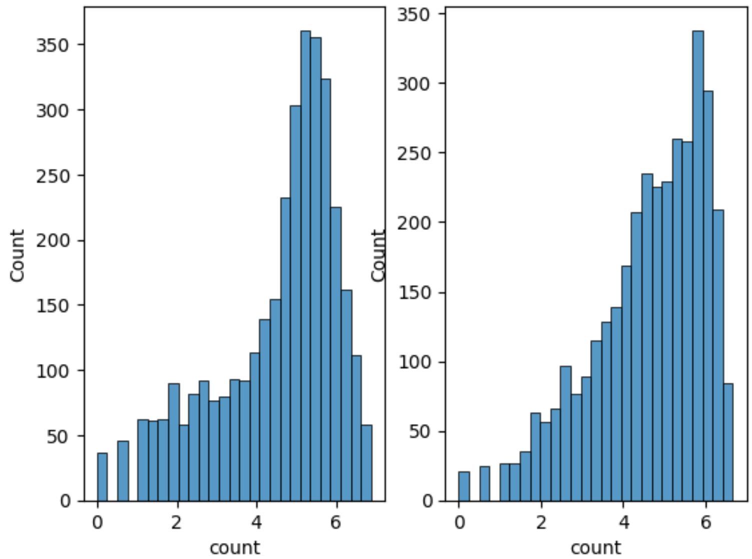
```
plt.subplot(121)
sns.histplot(workingday)

plt.subplot(122)
sns.histplot(non_workingday)
```

```
<Axes: xlabel='count', ylabel='Count'>
```




```
plt.subplot(121)
sns.histplot(np.log(workingday))

plt.subplot(122)
sns.histplot(np.log(non_workingday))
```

```
<Axes: xlabel='count', ylabel='Count'>
```



Normal Distribution:

KS Test to check if distribution are normal or not

```
log_workingday = np.log(workingday)
kstest(log_workingday,norm.cdf,args=(log_workingday.mean(),log_workingday.std()))
```

```
KstestResult(statistic=0.13963258596239647, pvalue=1.4928477298254112e-59, statistic_location=4.672828834461906,
statistic_sign=-1)
```

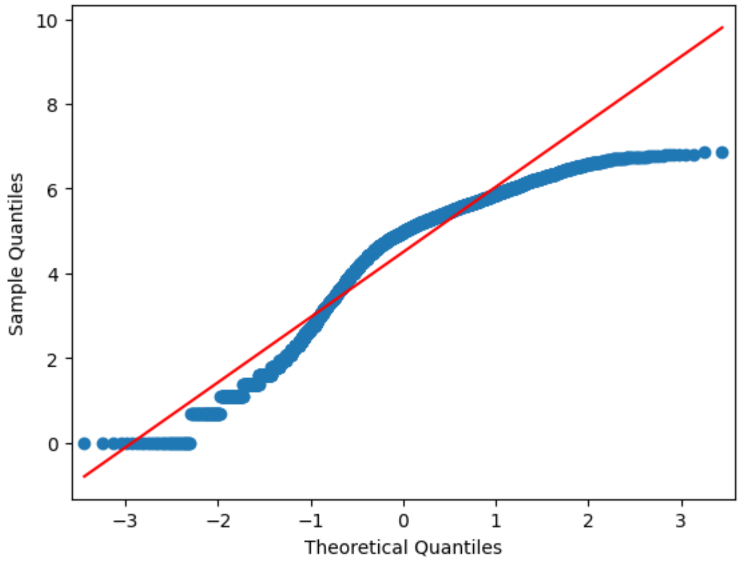
```
log_non_workingday = np.log(non_workingday)
kstest(log_non_workingday,norm.cdf,args=(log_non_workingday.mean(),log_non_workingday.std()))

##--> Reject HO however lets assume similar
```

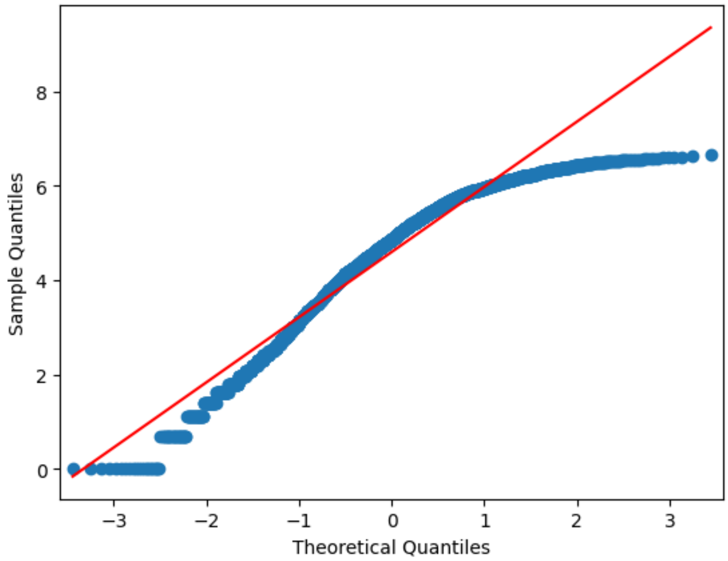
```
KstestResult(statistic=0.08314486002517152, pvalue=2.432481041131246e-21, statistic_location=4.997212273764115, statistic_sign=-1)
```

```
# OR
```

QQ-plot

```
qqplot(log_workingday,line='s')
plt.show()

qqplot(log_non_workingday,line='s')
plt.show()
```





T-Test Ind

```
tstat,pvalue= ttest_ind(np.log(workingday),np.log(non_workingday),alternative="two-sided")
print("tstat =", tstat)
print("pvalue =", pvalue)
```

```
    tstat = -2.4846834094900614
    pvalue = 0.012990042271292085
```

```
alpha = 0.05

if pvalue < alpha :
  print("Reject H0")
else:
  print("Failed to Reject H0")
```

```
    Reject H0
```

Insights:

pvalue is greater than alpha. Hence We have failed to reject H0

i.e We dont have enough evidence to say that Working Day has effect on number of electric cycles rented.

Hypothesis Testing - 2 (ANOVA test - season)

Q) No. of cycles rented similar or different in different seasons

Step 1: Define null and alternative hypothesis

Null Hypothesis (H0): No. of cycles rented similar in different seasons(mu1=mu2=mu3=mu4)

Alternate Hypothesis (H1): No. of cycles rented different in different seasons(Atleast one of mean of count is not same)

Step 2: Set a significance level (alpha)

Significance level (alpha): 0.05

Anova and its assumptions test-

1) Normal Distribution:- Sampled groups are assumed to be drawn from normally distributed populations.

2) Homogeneity of variance - We Assume variances is same across all groups.

3) Sample drawn is independent.

```
yulu.groupby('season')['count'].describe()
```

season	count	mean	std	min	25%	50%	75%	max
1	2686.0	116.343261	125.273974	1.0	24.0	78.0	164.0	801.0
2	2733.0	215.251372	192.007843	1.0	49.0	172.0	321.0	873.0
3	2733.0	234.417124	197.151001	1.0	68.0	195.0	347.0	977.0
4	2734.0	198.988296	177.622409	1.0	51.0	161.0	294.0	948.0

```
spring = yulu[yulu['season']==1]['count']
summer = yulu[yulu['season']==2]['count']
fall = yulu[yulu['season']==3]['count']
winter = yulu[yulu['season']==4]['count']
```

ANOVA Normality

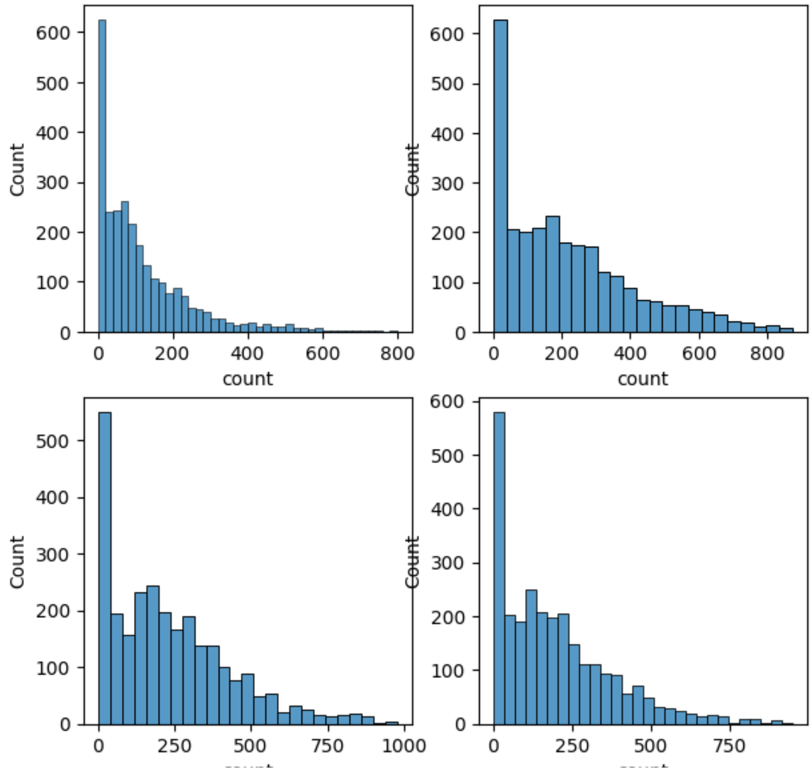
Shapiro test -

H0 - Data is Gaussian

Ha - Data is not gaussian

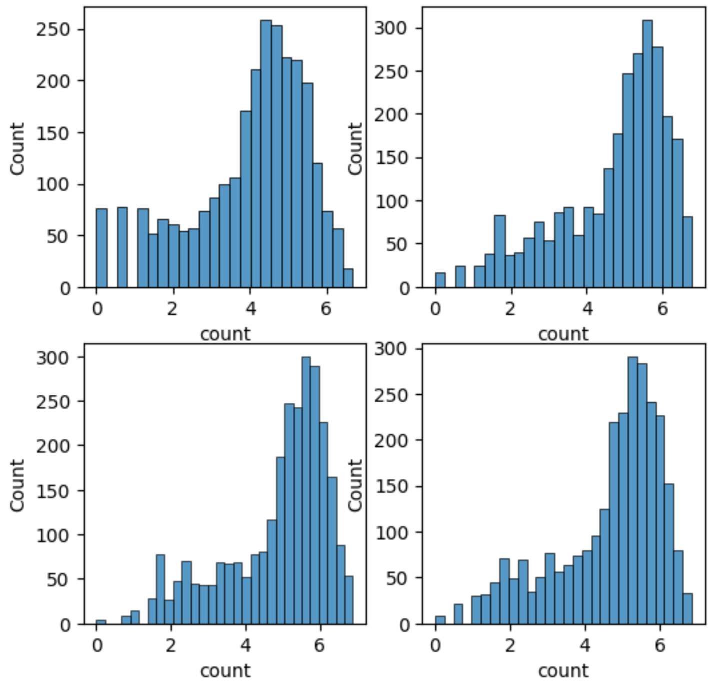
```
fig,ax = plt.subplots(2,2,figsize=(7,7))
sns.histplot(spring, ax=ax[0,0], palette = "bone_r")
sns.histplot(summer, ax=ax[0,1], palette = "PiYG_r")
sns.histplot(fall, ax=ax[1,0], palette = "PuBuGn_r")
sns.histplot(winter, ax=ax[1,1])

plt.show()
```

```
<ipython-input-128-181096a5301a>:2: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(spring, ax=ax[0,0], palette = "bone_r")
<ipython-input-128-181096a5301a>:3: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(summer, ax=ax[0,1], palette = "PiYG_r")
<ipython-input-128-181096a5301a>:4: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(fall, ax=ax[1,0], palette = "PuBuGn_r")
```




```
## Converting to Lognormal dist as it is right skewed
fig,ax = plt.subplots(2,2,figsize=(6,6))
sns.histplot(np.log(spring), ax=ax[0,0], palette = "bone_r")
sns.histplot(np.log(summer), ax=ax[0,1], palette = "PiYG_r")
sns.histplot(np.log(fall), ax=ax[1,0], palette = "PuBuGn_r")
sns.histplot(np.log(winter), ax=ax[1,1])
plt.show()
```

```
<ipython-input-149-ebdda436fd10>:3: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(np.log(spring), ax=ax[0,0], palette = "bone_r")
<ipython-input-149-ebdda436fd10>:4: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(np.log(summer), ax=ax[0,1], palette = "PiYG_r")
<ipython-input-149-ebdda436fd10>:5: UserWarning: Ignoring `palette` because no `hue`
  sns.histplot(np.log(fall), ax=ax[1,0], palette = "PuBuGn_r")
```




```
shapiro(np.log(spring))
```

```
ShapiroResult(statistic=0.9254210591316223, pvalue=1.354961114124157e-34)
```

```
shapiro(np.log(summer))
```

```
ShapiroResult(statistic=0.904330313205719, pvalue=2.2642132942412607e-38)
```

```
shapiro(np.log(fall))
```

```
ShapiroResult(statistic=0.891656756401062, pvalue=3.384359999098725e-40)
```

```
shapiro(np.log(winter))
```

```
ShapiroResult(statistic=0.90409255027771, pvalue=2.0568127137304018e-38)
```

As we can see from above pvalues of 4 samples - pvalue is less than lambda.Hence Reject H0.

But lets assume Normal Distribution

Levene Test for Homogeneity of variance

HO - Variences are same Ha- Variences are diff

```
levene(np.log(spring),np.log(summer),np.log(fall),np.log(winter))
```

```
LeveneResult(statistic=9.640605587638781, pvalue=2.3678125658230693e-06)
```

As per pvalue we can see that We have to Reject HO and variences are not same. For now lets assume true

```
f_stat,season_pvalue=f_oneway(np.log(spring),np.log(summer),np.log(fall),np.log(winter))
print("f_stat =", f_stat)
print("season_pvalue =", season_pvalue)
```

```
f_stat = 192.44768979509686
season_pvalue = 1.3071364586238867e-121
```

```
alpha = 0.05

if season_pvalue < alpha :
  print("Reject H0")
else:
  print("Failed to Reject H0")
```

```
Reject H0
```

Insights:

pvalue is much lower that alpha

Hence Rejecting H0 i.e No. of cycles rented are different in different seasons

Hypothesis Testing - 3 (ANOVA test - weather)

Q) No. of cycles rented similar or different in different seasons

Step 1: Define null and alternative hypothesis

Null Hypothesis (H0): No. of cycles rented similar in different weathers(mu1=mu2=mu3=mu4)

Alternate Hypothesis (H1): No. of cycles rented different in different weathers(Atleast one of mean of count is not same)

Step 2: Set a significance level (alpha)

Significance level (alpha): 0.05

```
yulu.groupby('weather')['count'].describe()
```

count mean std min 25% 50% 75% max

```
weather1 = yulu[yulu['weather']==1]['count']
weather2 = yulu[yulu['weather']==2]['count']
weather3 = yulu[yulu['weather']==3]['count']
weather4 = yulu[yulu['weather']==4]['count']

f_stat_w,weather_pvalue=f_oneway(weather1,weather2,weather3,weather4)
print("f_stat weather =", f_stat_w)
print("weather_pvalue =", weather_pvalue)
```

```
f_stat weather = 65.53024112793271
weather_pvalue = 5.482069475935669e-42
```

```
alpha = 0.05

if weather_pvalue < alpha :
  print("Reject H0")
else:
  print("Failed to Reject H0")
```

```
Reject H0
```

Insights:

pvalue is much lower that alpha

Hence Rejecting H0 i.e No. of cycles rented are different in different weathers

Chi-Squared Test - Both are categorical variables

Weather relation with season

We are checking if weather and season has a relation.

Assumptions:

Test of independence.

Each cell should contain min value of 5

Each cell is Mutually exclusive

Step 1: Define null and alternative hypothesis

Null Hypothesis (H0): Weather is not dependent on season

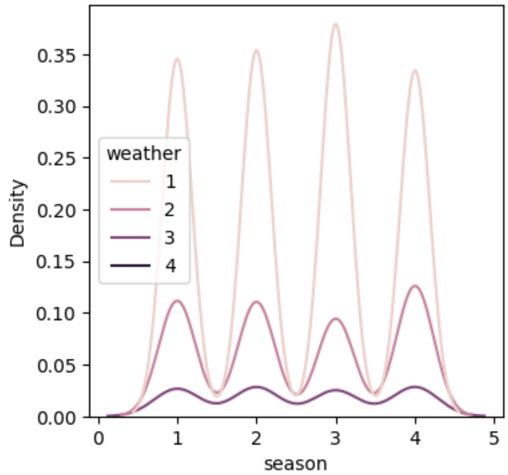
Alternate Hypothesis (H1): Weather is dependent on season

Step 2: Set a significance level (alpha)

Significance level (alpha): 0.05

```
plt.figure(figsize=(4,4))
sns.kdeplot(data=yulu,x='season',hue='weather')
```

```
<ipython-input-152-2d745a43a6a7>:2: UserWarning: Dataset has 0 variance; skipping density estimate. Pass `warn_singular=False` to d
  sns.kdeplot(data=yulu,x='season',hue='weather')
<Axes: xlabel='season', ylabel='Density'>
```




```
data_table= pd.crosstab(yulu['season'], yulu['weather'])
```

```
data_table
```

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

Removing Weather 4 as it doesnt satify the Assumptions (Each cell should contain min value of 5)

```
list1 = [4]
yulu_remove_4 = yulu[yulu.weather.isin(list1) == False]
```

```
data_table= pd.crosstab(yulu_remove_4['season'], yulu_remove_4['weather'])
data_table
```

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

```
chi2, pval, dof, exp_freq = chi2_contingency(data_table)
print("chi-square statistic: {} ,\nPvalue: {} , \nDegree of freedom: {} ,\nexpected frequency:\n{} ".format(chi2, pval, dof, exp_freq))
```

```
chi-square statistic: 46.10145731073249 ,
Pvalue: 2.8260014509929343e-08 ,
Degree of freedom: 6 ,
expected frequency:
[[1774.04869086  699.06201194  211.8892972 ]
 [1805.76352779  711.55920992  215.67726229]
 [1805.76352779  711.55920992  215.67726229]
 [1806.42425356  711.81956821  215.75617823]]
```

```
alpha = 0.05

if pval < alpha :
  print("Reject H0")
else:
  print("Failed to Reject H0")
```

```
Reject H0
```

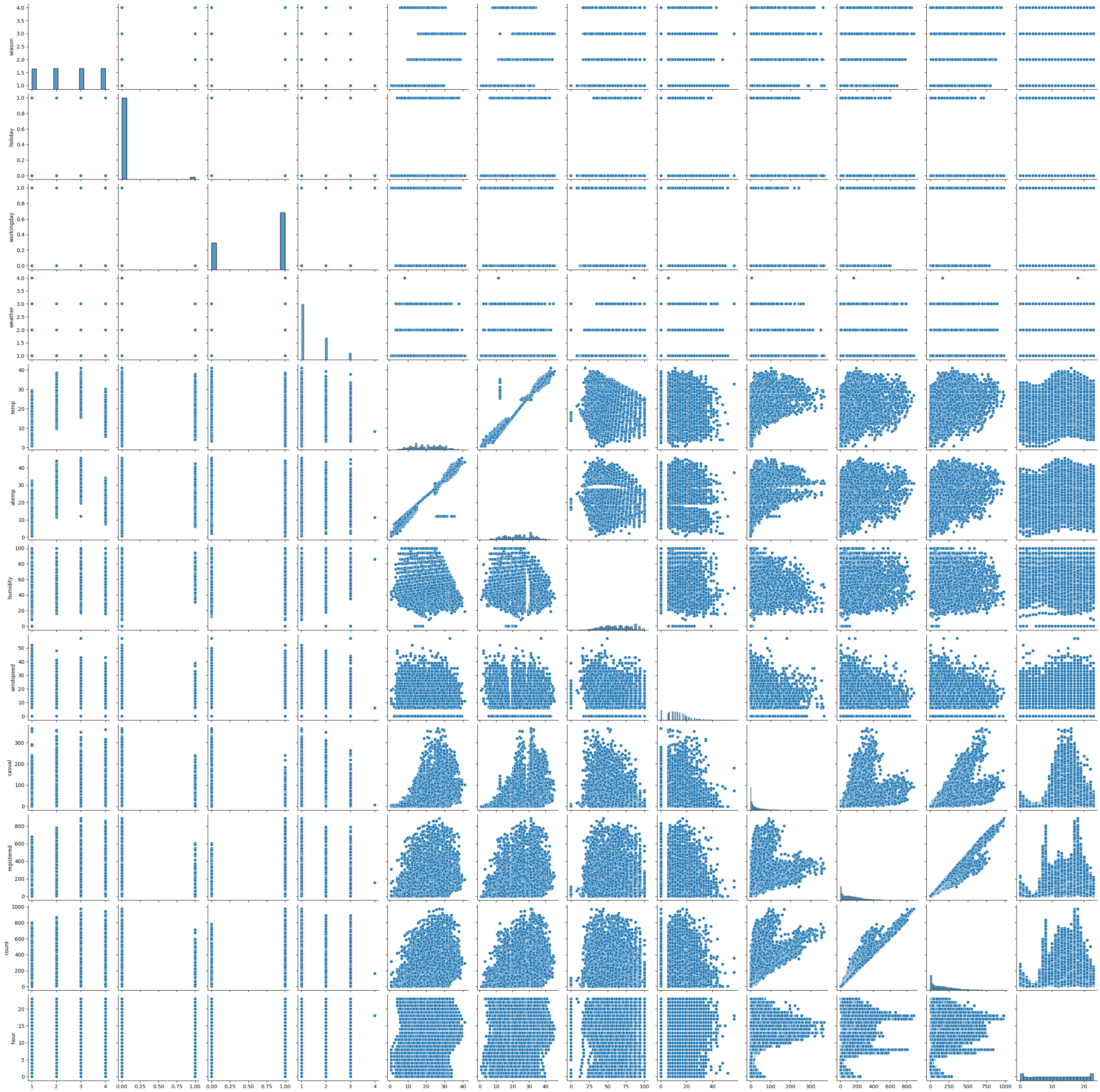
Insights:

P-Value is low. so Null hypotheis is rejected.

i.e. Weather is dependent and has effect on season

```
sns.pairplot(yulu)
```

```
<seaborn.axisgrid.PairGrid at 0x7f3b291ac520>
```



Insights:

1) More number of bikes are being rented in season fall and summer compared to other seasons.

2) More number of bikes are being rented when there is a holiday.

3) When it comes to weather, most of the bikes are rented when its 1: Clear, Few clouds, partly cloudy, partly cloudy

and very least no.of bikes are being rented when weather is 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

4) When temparture is below 5 Celcius bikes rented is very low.

5) Sum of cycles rented are different in different seasons.

6) No. of cycles being rented are different in different weathers.

7) Weather is dependent and has effect on season.

8) More no.of Bikes are being rented between 4 PM to 8 PM compared to other timings and less no.of bikes rented between 12 AM - 5 AM.

Recommendations:

1) In summer and fall seasons the yulu company should have more bikes in stock in order to be rented. Because the demand in these 2 seasons is higher as compared to other seasons.

2) With 95% confidence level, Working Day has no effect on number of electric cycles rented.

3) Whenever temprature is less than 5 (cold days) , company should have less bikes to be stocked.

As More no.of Bikes are being rented between 4 PM to 8 PM compared to other timings, Company should arrange more no.of Bikes in that period of time.

✓ 1s completed at 20:52