Problem Statement

To find out :

- 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

We will be using different hypothesis tests to find the significance of different features that effect the demand.

In [ ]:

```

```

In [1]:

```
#importing the required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import statsmodels.api as sm
```

In [82]:

```
#loading the dataset
df = pd.read_csv("bike_sharing.csv")
df.head()
```

Out[82]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5
3	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0	3
4	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0	0

In [3]:

```
#No. of rows and columns
df.shape
```

Out[3]:

```
(10886, 12)
```

In [4]:

```
#checking Data types
df.dtypes
```

Out[4]:

```
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
```

In [5]:

```
# No of unique values
for i in df.columns:
    print(i, ':' , df[i].nunique())
```

```
datetime : 10886
season : 4
holiday : 2
workingday : 2
weather : 4
temp : 49
atemp : 60
humidity : 89
windspeed : 28
casual : 309
registered : 731
count : 822
```

In [6]:

```
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
```

From above observations, we can say that datetime contains the dates and time and hence can be converted to datetime dtype. Season, holiday, working day, weather are categorical features and rest are continuous.

count is the target variable (dependent) and all others are independent

In [7]:

```
#converting datetime feature to datetime dtyoe
df['datetime'] = pd.to_datetime(df['datetime'])
```

In [8]:

```
df.dtypes
```

Out[8]:

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

In [9]:

```
#checking null values in every column of our data
df.isnull().sum() / len(df) * 100
```

Out[9]:

```
datetime      0.0
season        0.0
holiday       0.0
workingday    0.0
weather       0.0
temp          0.0
atemp         0.0
humidity      0.0
windspeed     0.0
casual        0.0
registered    0.0
count         0.0
dtype: float64
```

There aren't any nulls in our dataframe

In [ ]:

```

```

Checking value counts for categorical columns

In [10]:

```
df['season'].value_counts()
```

Out[10]:

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

In [11]:

```
df['weather'].value_counts()
```

Out[11]:

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

In [12]:

```
df['workingday'].value_counts()
```

Out[12]:

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

In [13]:

```
df['holiday'].value_counts()
```

Out[13]:

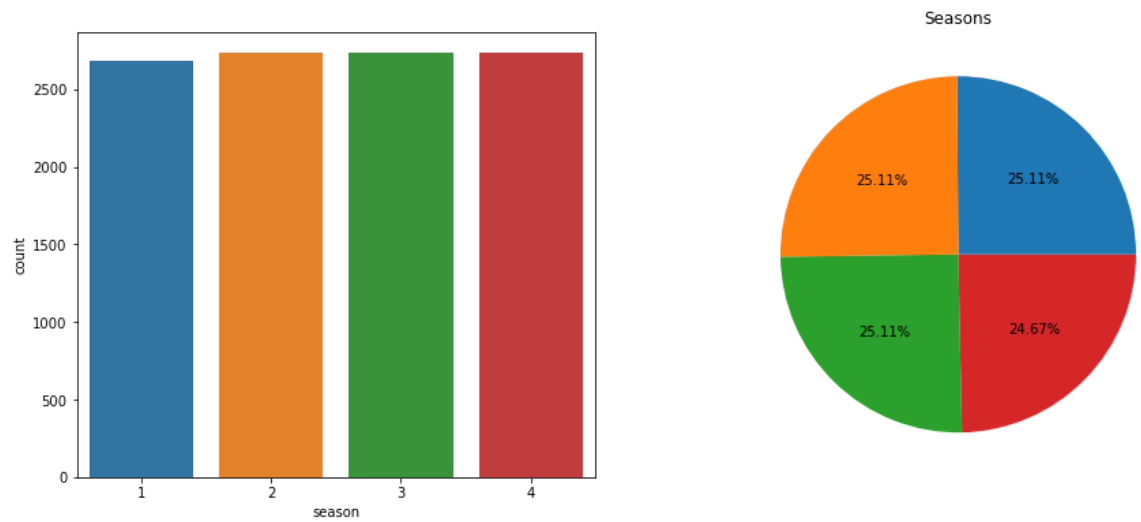
```
0    10575
1      311
Name: holiday, dtype: int64
```

Univariate Analysis

Categorical features

In [89]:

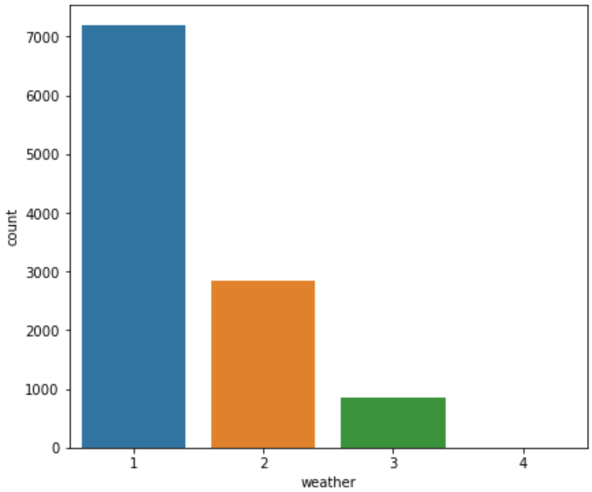
```
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.countplot(data = df , x= 'season')

plt.subplot(122)
plt.pie(df['season'].value_counts() , autopct = '%2.2f%%')
plt.title("Seasons")
plt.show()
```

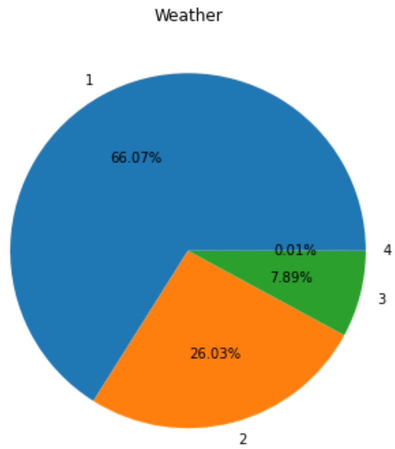


We can observe that almost all seasons have same data points in the dataset

In [93]:

```
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.countplot(data = df , x= 'weather')

plt.subplot(122)
plt.pie(df['weather'].value_counts() , autopct = '%2.2f%%', labels=df['weather'].unique())
plt.title("Weather")
plt.show()
```





In [16]:

```
df['weather'].value_counts(normalize=True)
```

Out[16]:

```
1    0.660665
2    0.260334
3    0.078909
4    0.000092
Name: weather, dtype: float64
```

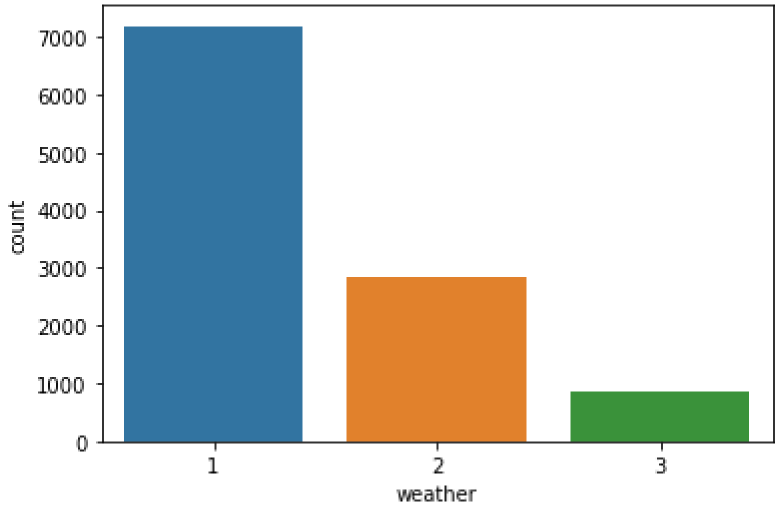
- We observe majority (66%) of sales are in weather category 1
- since we have only one data point for weather category 4 we can drop it

In [17]:

```
df = df[df['weather'] != 4]
```

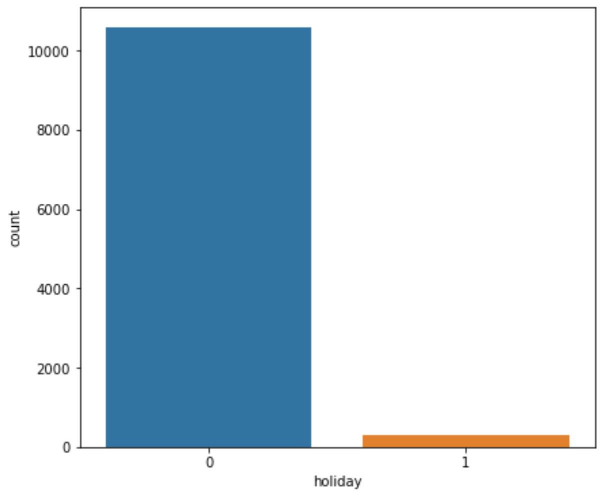
In [18]:

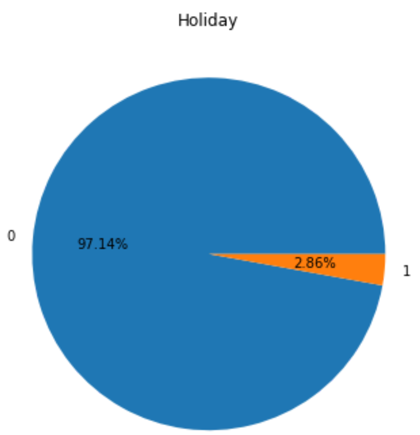
```
sns.countplot(data = df , x= 'weather')
plt.show()
```



In [95]:

```
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.countplot(data = df , x= 'holiday')

plt.subplot(122)
plt.pie(df['holiday'].value_counts() , autopct = '%2.2f%%' , labels= ['0','1'])
plt.title("Holiday")
plt.show()
```





In [20]:

```
df['holiday'].value_counts(normalize=True)
```

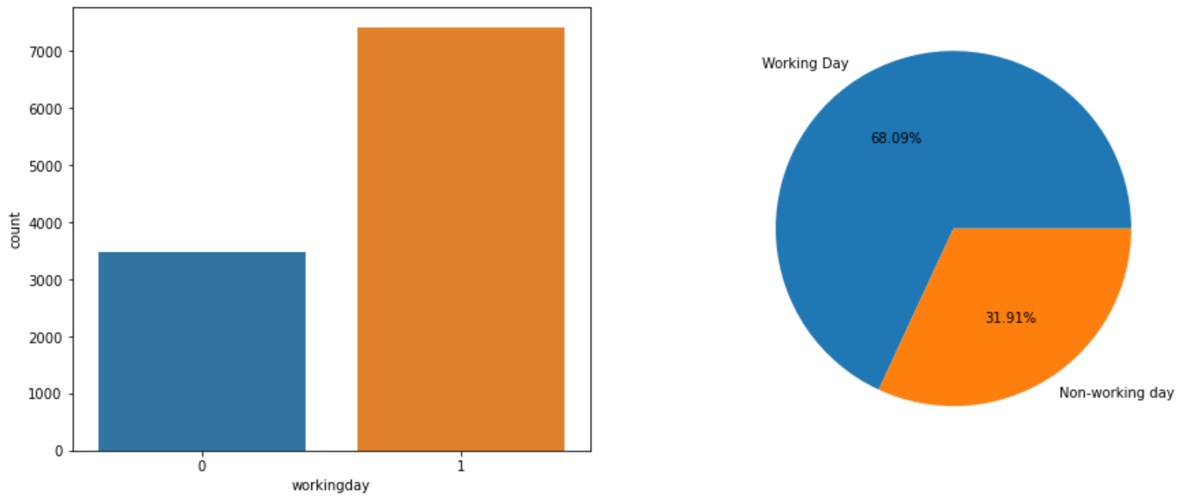
Out[20]:

```
0    0.971429
1    0.028571
Name: holiday, dtype: float64
```

we can see that 97 % of bookings are made on non holiday days

In [96]:

```
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.countplot(data = df , x= 'workingday')

plt.subplot(122)
plt.pie(df['workingday'].value_counts() , autopct = '%2.2f%%' , labels= ['Working Day' , 'N
plt.show()
```



In [22]:

```
df['workingday'].value_counts(normalize=True)
```

Out[22]:

```
1    0.680845
0    0.319155
Name: workingday, dtype: float64
```

we see that 68% of bokings are on a working day and since bokking on holidays is only 2.8% we can make a safe assumption that the majority bookings on non workingdays are on weekends

Insights

- We have almost same count of datapoints for all 4 seasons
- We observe majority (66%) of bookings are in weather category 1 - Clear, Few clouds, partly cloudy, partly cloudy
- 97 % of bookings are made on non holiday days.
- 68% of bokings are on a working day and 32% on non-working days

Continuous Features

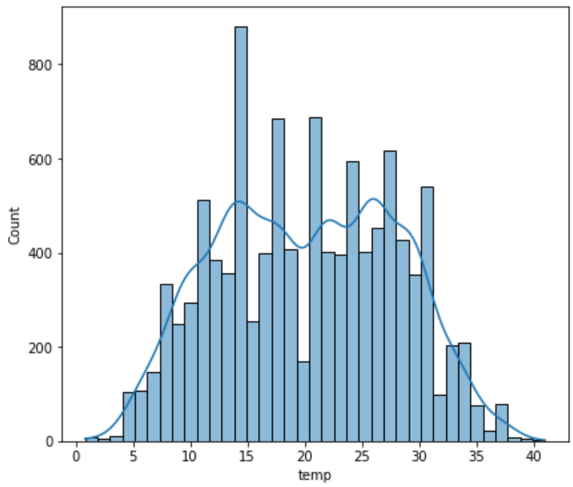
In [ ]:

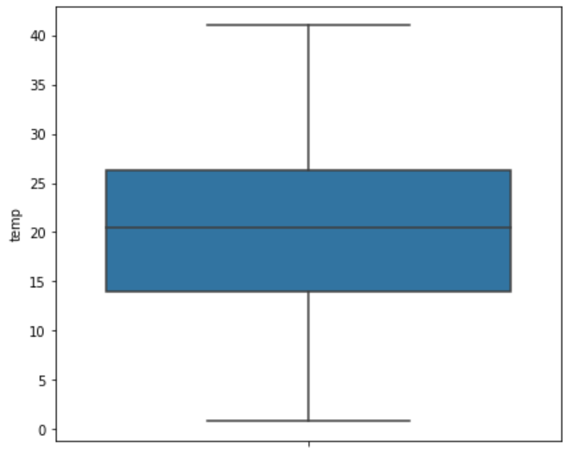
```

```

Type *Markdown* and LaTeX: α^2

In [23]:

```
#Temperature
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.histplot(data= df, x= 'temp' , kde=True)

plt.subplot(122)
sns.boxplot(data= df, y= 'temp')
plt.show()
```





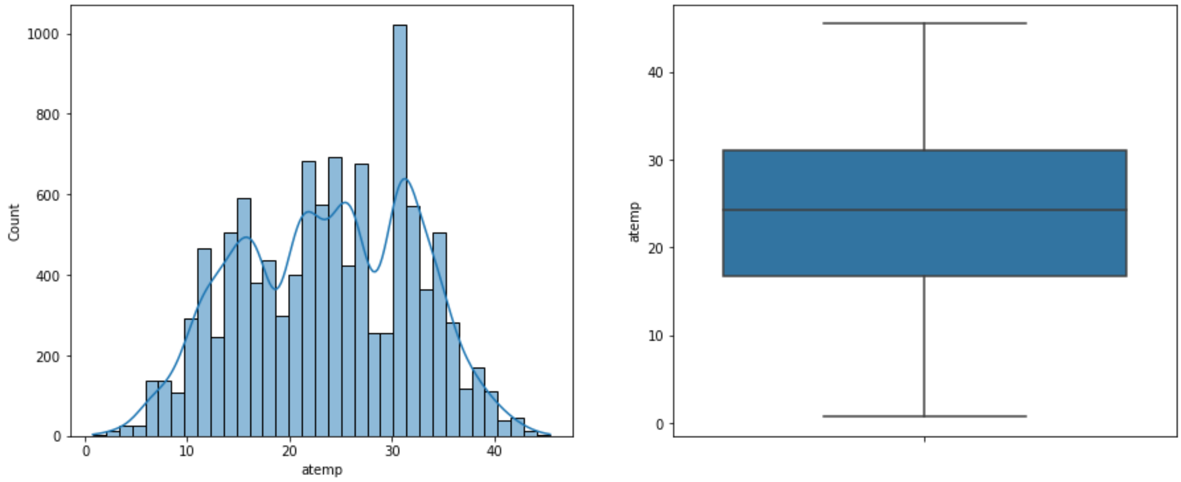
The most frequent temperature is around 14-15 degree celcius and median temp is around 20 degree celcius

In [ ]:

```

```

In [24]:

```
#Feeling Temperature
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.histplot(data= df, x= 'atemp' , kde=True)

plt.subplot(122)
sns.boxplot(data= df, y= 'atemp')
plt.show()
```

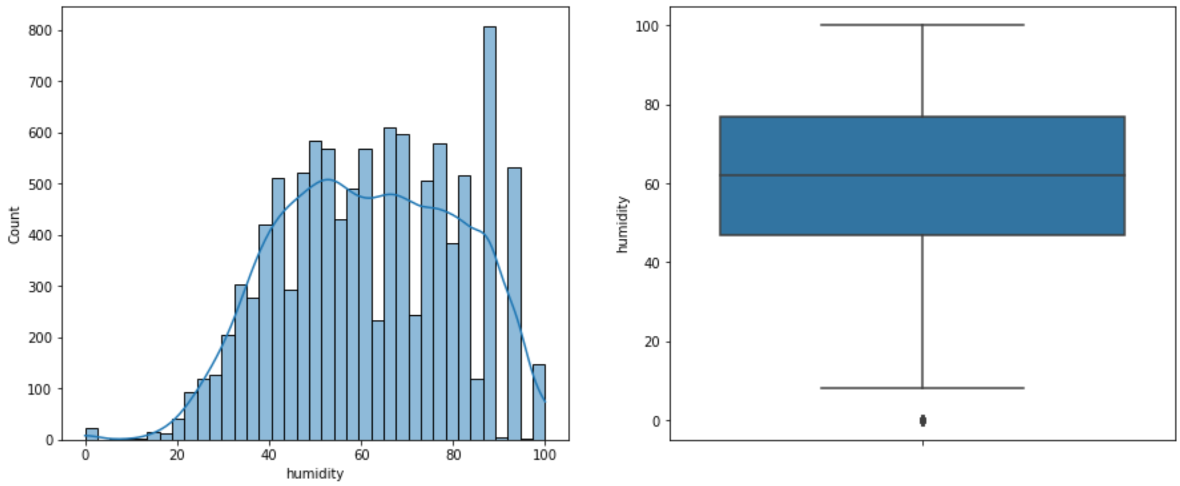


The most frequent feeling temperature is around 22-26 degree celcius with median at 25

In [ ]:

In [25]:

```
#Humidity
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.histplot(data= df, x= 'humidity' , kde=True)

plt.subplot(122)
sns.boxplot(data= df, y= 'humidity')
plt.show()
```



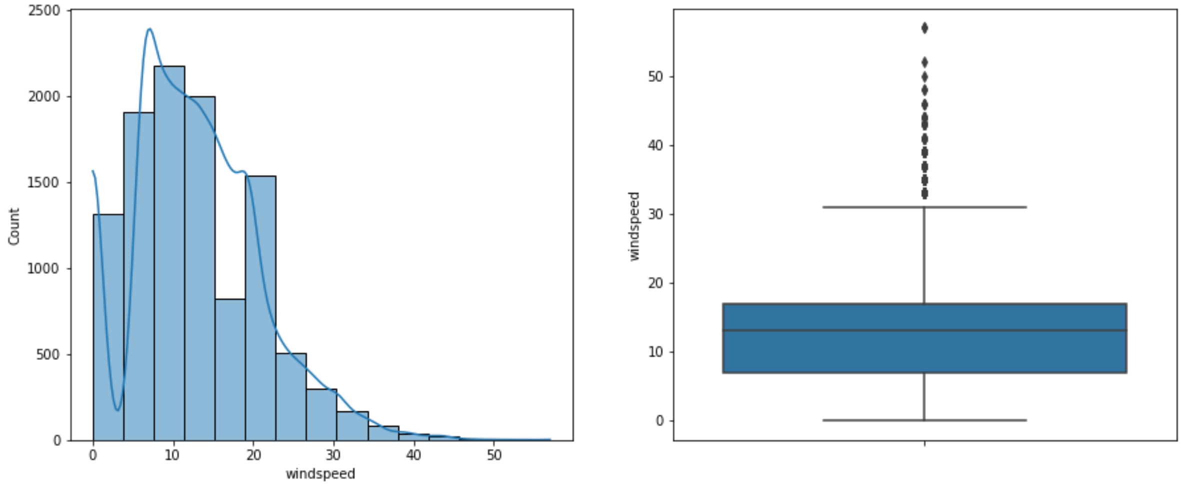
most frequent humidity levels are between 40-80 and median lvels are at 60

In [ ]:

```

```

In [26]:

```
#Windspeeds
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.histplot(data= df, x= 'windspeed' ,bins=15,  kde=True)

plt.subplot(122)
sns.boxplot(data= df, y= 'windspeed')
plt.show()
```



The most occuring windspeeds are between 5-15 with many outliers being present

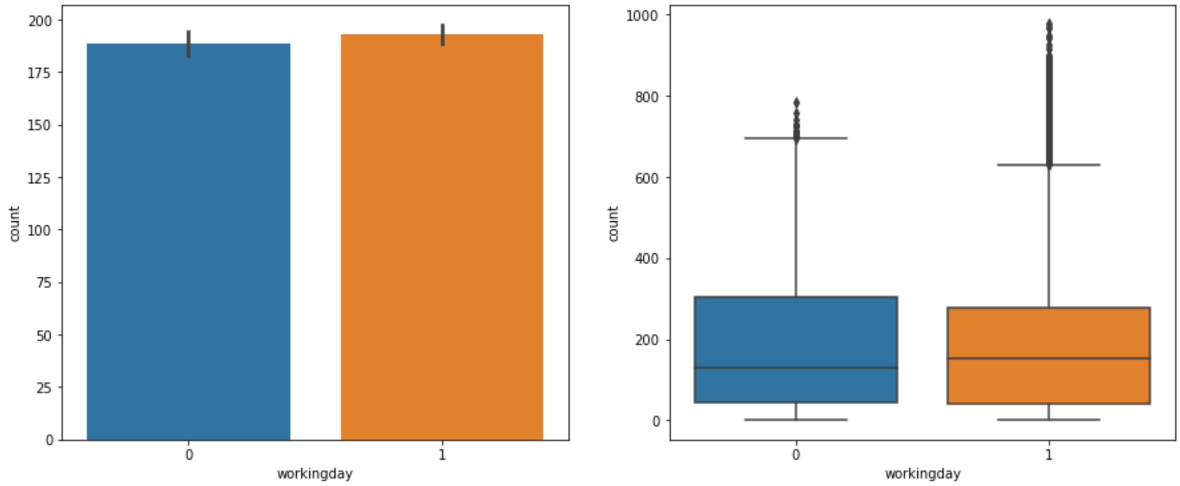
Insights

- The most frequent temperature is around 14-15 degree celcius and median temp is around 20 degree celcius
- The most frequent feeling temperature is around 22-26 degree celcius with median at 25
- The most occuring windspeeds are between 5-15 with many outliers being present

Bivariate Analysis

In [27]:

```
#relation between workday and count
plt.figure(figsize=(15,6))

plt.subplot(121)
sns.barplot(data= df, y= 'count', x='workingday')

plt.subplot(122)
sns.boxplot(data=df , y ='count' ,x= 'workingday')
plt.show()
```



In [28]:

```
df.groupby(by ='workingday')['count'].mean()
```

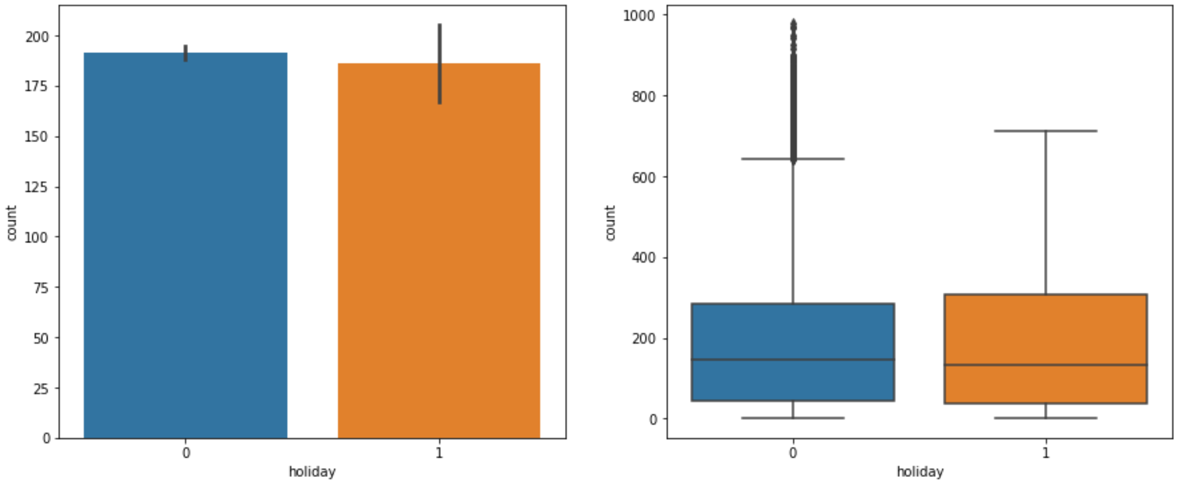
Out[28]:

```
workingday
0    188.506621
1    193.015787
Name: count, dtype: float64
```

- The mean count of rental bikes for working and non working days is almost same with slightly higer for working day
- The max and median count for working day is greater as compared to the non working day

In [98]:

```
#relation between holiday and count
plt.figure(figsize=(15,6))

plt.subplot(121)
sns.barplot(data= df, y= 'count', x='holiday')

plt.subplot(122)
sns.boxplot(data=df , y ='count' ,x= 'holiday')
plt.show()
```

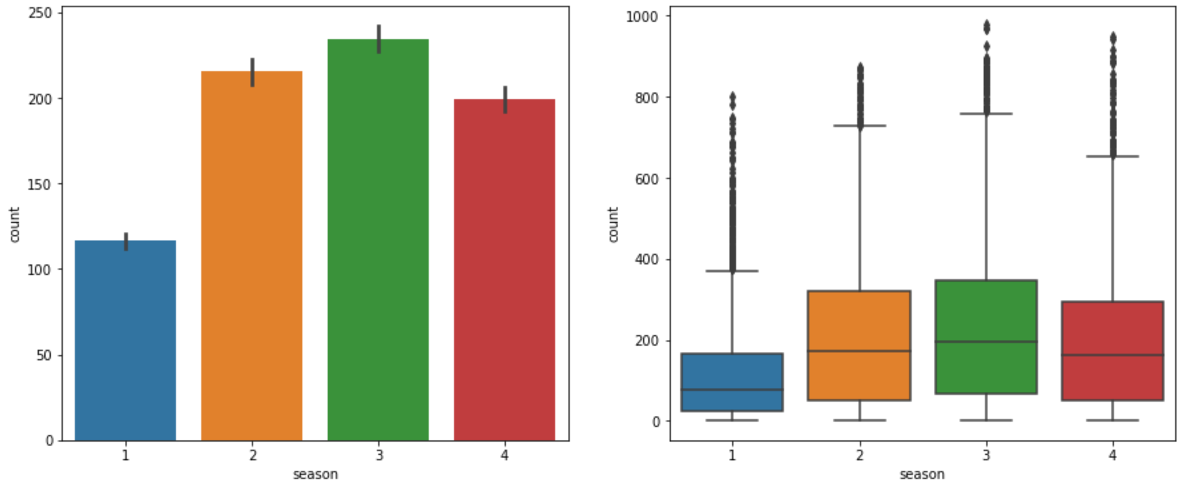


In [ ]:

- The mean and median count of rental bikes for holidya and non-holidays is almost same with slightly higher for non holiday.

In [29]:

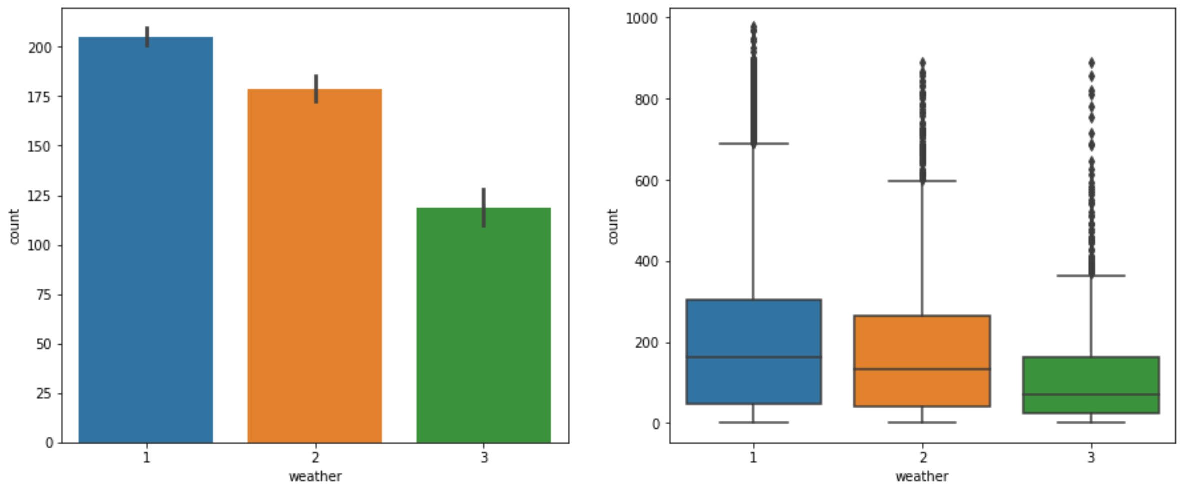
```
#relation between season and count
plt.figure(figsize=(15,6))

plt.subplot(121)
sns.barplot(data= df, y= 'count', x='season')

plt.subplot(122)
sns.boxplot(data=df , y ='count' ,x= 'season')
plt.show()
```



The mean and median of count of rentals is highest in season 3 - fall and lowest in season 1 - spring.

In [30]:

```
#relation between weather and count
plt.figure(figsize=(15,6))

plt.subplot(121)
sns.barplot(data= df, y= 'count', x='weather')

plt.subplot(122)
sns.boxplot(data=df , y ='count' ,x= 'weather')
plt.show()
```



In [31]:

```
df.groupby(by ='weather')['count'].mean()
```

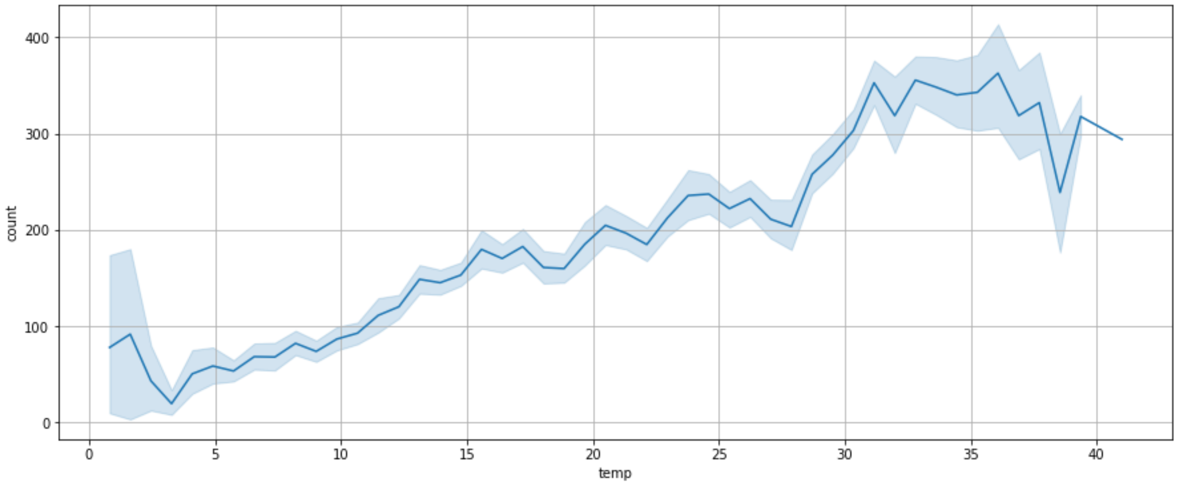
Out[31]:

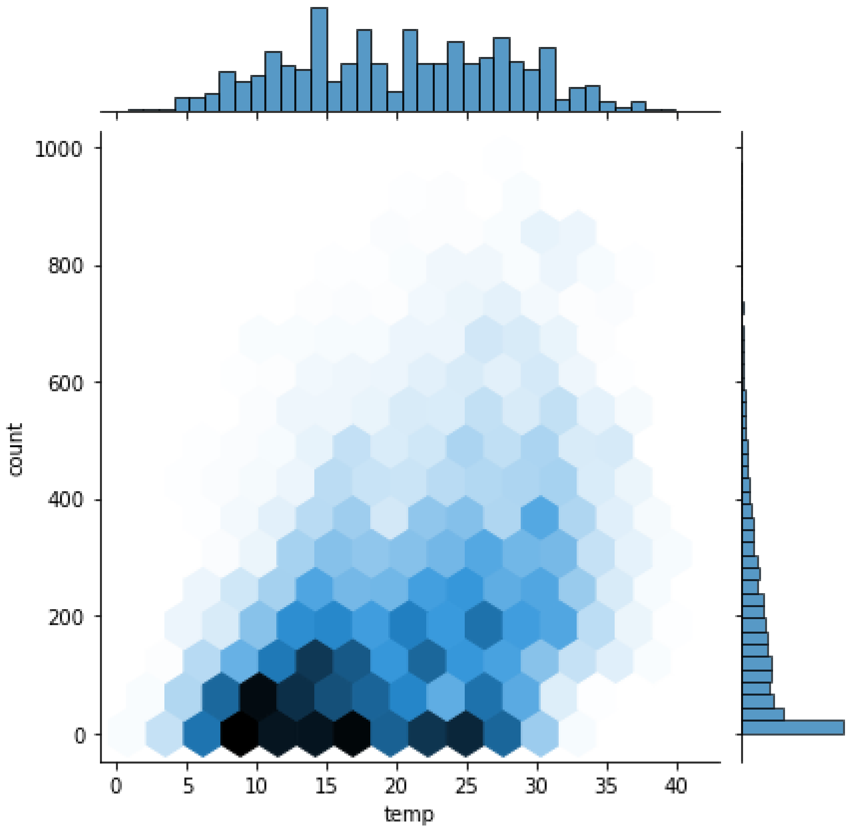
```
weather
1    205.236791
2    178.955540
3    118.846333
Name: count, dtype: float64
```

the mean and median of count of rentals is maximum in weather category 1 and least in category 3

In [88]:

```
#relation between temp and count
plt.figure(figsize=(15,6))

sns.lineplot(data= df, x= 'temp', y = 'count')
plt.grid()

sns.jointplot(data=df , x='temp', y= 'count' , kind = 'hex' , gridsize=15)
plt.show()
```

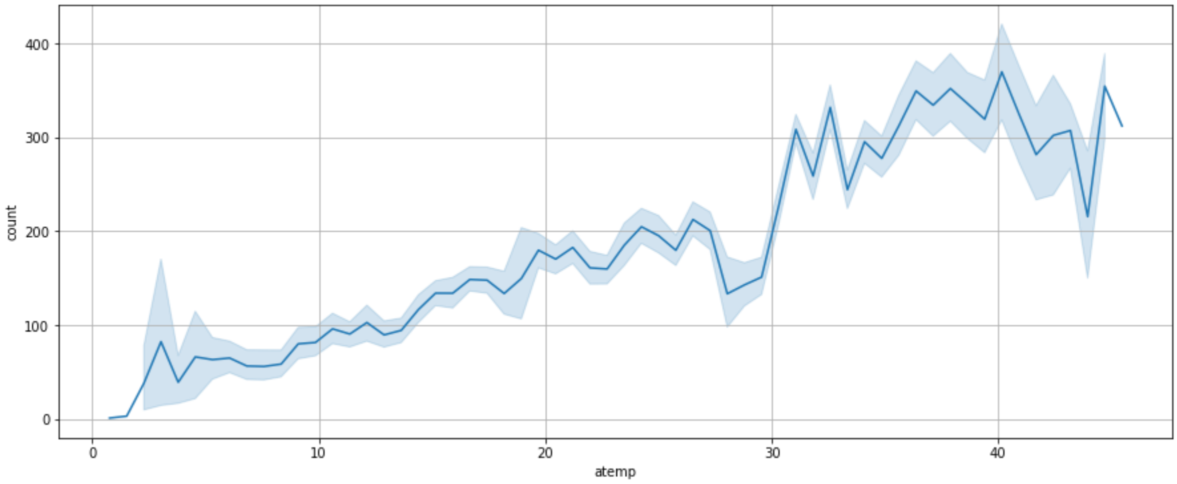


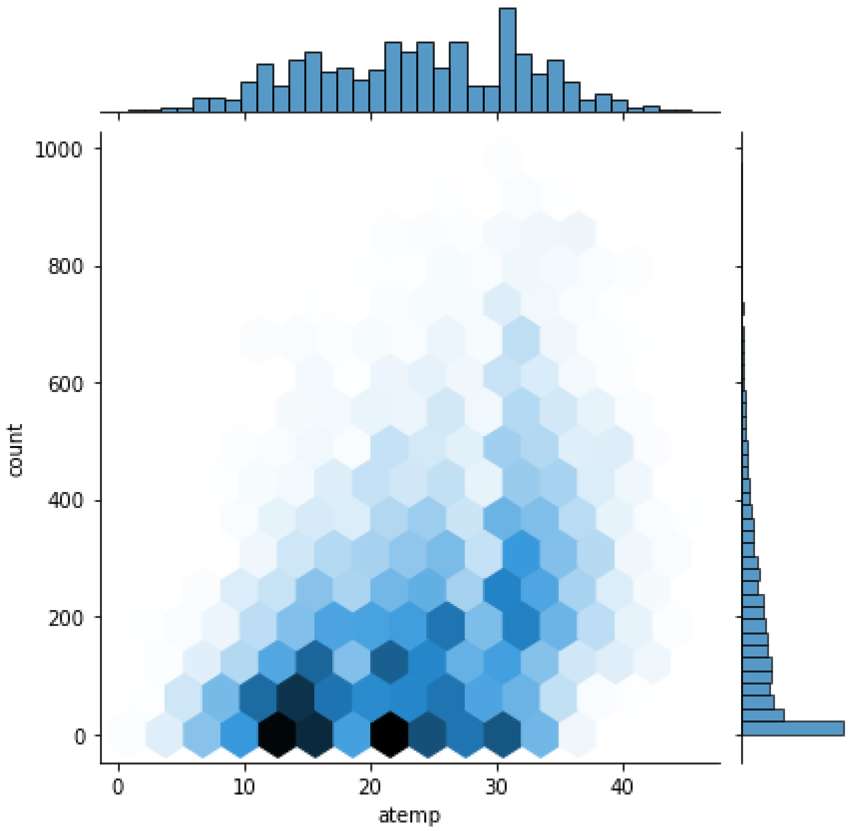


- The highest count of rentals is between 31-36 degree celcius while the highest frequency(density) is between 8-17 degrees and count of rentals around 100.
- we see as temp increases there is an increase in the count of rentals

In [99]:

```
#relation between atemp and count
plt.figure(figsize=(15,6))

sns.lineplot(data= df, x= 'atemp', y = 'count')
plt.grid()

sns.jointplot(data=df , x='atemp', y= 'count' , kind = 'hex' , gridsize=15)
plt.show()
```





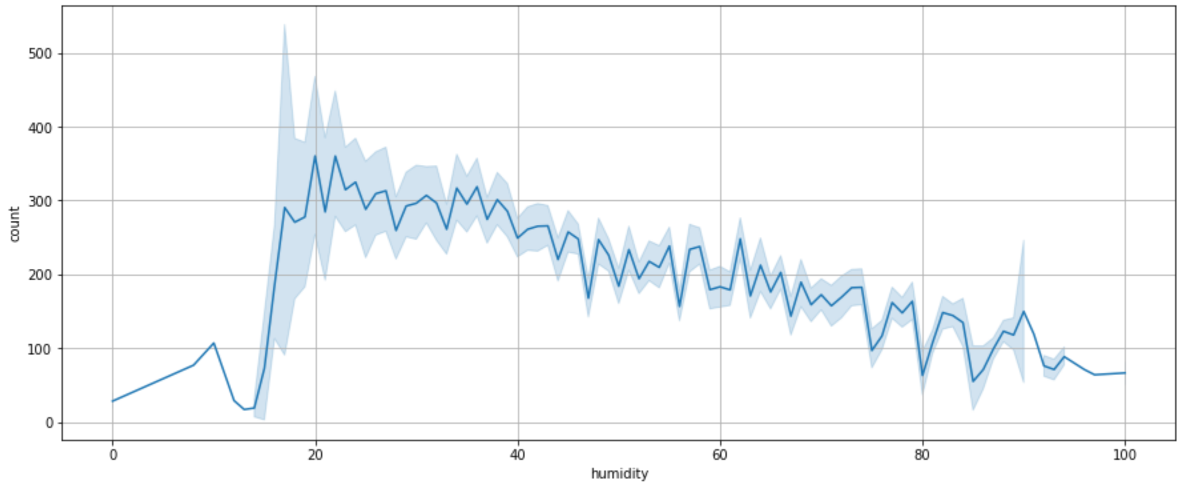
- The highest count of rentals is between 36-40 degree celcius feeling temperature while the highest frequency(density) is between 12-16 degrees and count of rentals around 100.
- we see as feeling temp increases there is an increase in the count of rentals.

In [ ]:

```

```

In [105]:

```
#relation between humidity and count
plt.figure(figsize=(15,6))

sns.lineplot(data= df, x= 'humidity', y = 'count')
plt.grid()
plt.show()
```



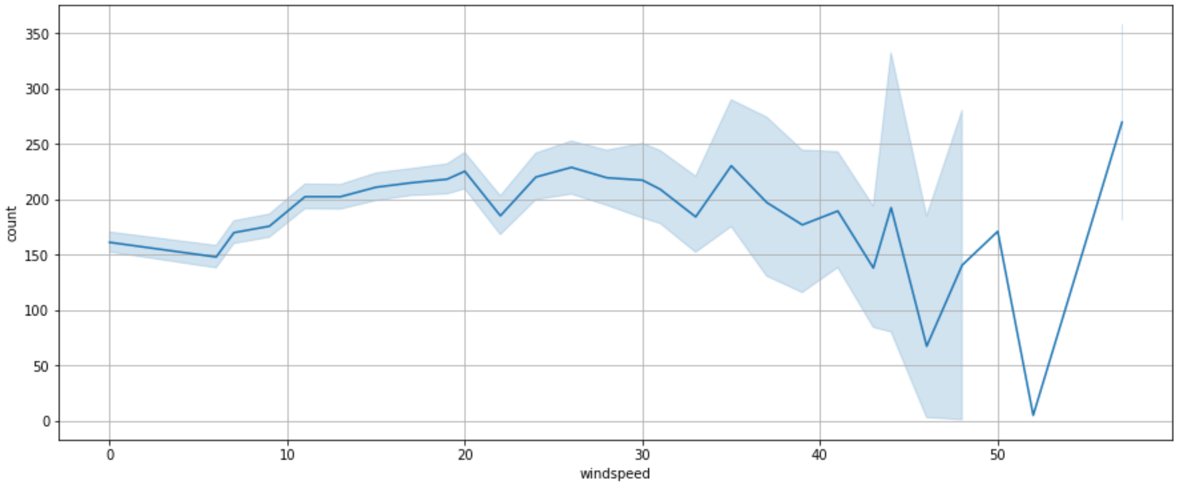
- The count of rentals is very low for humidity less than 18 but is maximum at 20.
- After humidity level 20 the count of rentals decreases as humidity increases.

In [ ]:

```

```

In [106]:

```
#relation between windspeed and count
plt.figure(figsize=(15,6))

sns.lineplot(data= df, x= 'windspeed', y = 'count')
plt.grid()


plt.show()
```



- the count is maximum between windspeed of 12-30 approx.
- above windspeeds of 35 we can observe a decrease in the count of rentals

Checking of relations between the dependent and independent features

1) Workday with count

- Null Hypothesis Ho : The means of count of non working day and working day are equal.
- Alternate Hypothesis Ha: The means of count of non working day and working day are not equal.
- Significance level = 5%

To test our null hypothesis we will use the **2-sample t test** as population standard deviations are not present.

To perform t test the means of samples must be normally distributed

In [34]:

```
nonworking = df[df['workingday'] == 0]['count']
working = df[df['workingday'] == 1]['count']
```

In [35]:

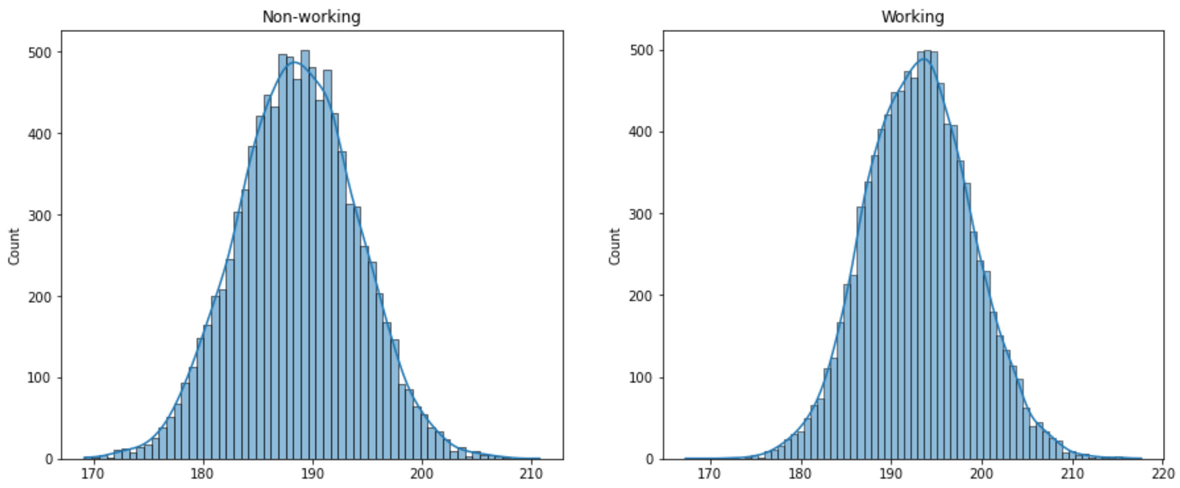
```
alpha = 0.05
```

In [36]:

```
#testing if means of samples  of count for working and non working are Gaussian
# Let's create r=10000 bootstrap samples, and let each bootstrap sample be of size=1000
# bs_means_m is a list of 'r' bootstrap sample means of purchase totals of males
r = 10000
size = 1000

bs_means_nw = np.empty(r)
bs_means_w = np.empty(r)

for i in range(r):
    bs_sample_nw = np.random.choice(nonworking, size =size)
    bs_means_nw[i] = np.mean(bs_sample_nw)

for i in range(r):
    bs_sample_w = np.random.choice(working, size =size)
    bs_means_w[i] = np.mean(bs_sample_w)
```

In [37]:

```
#Distribution of sample means
plt.figure(figsize=(15,6))
plt.subplot(121)
sns.histplot(bs_means_nw, kde=True)
plt.title("Non-working")

plt.subplot(122)
sns.histplot(bs_means_w, kde=True)
plt.title("Working")
plt.show()
```

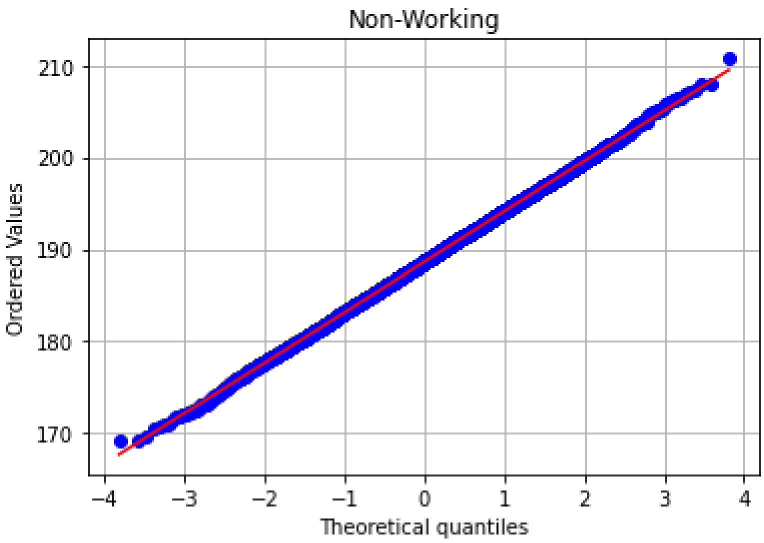


In [38]:

```
#QQ-plot to check if distribution is normal
fig, ax1 = plt.subplots()
plt.grid()
prob = stats.probplot(bs_means_nw, dist=stats.norm,  fit=True, plot=ax1)
plt.title("Non-Working")
```

Out[38]:

```
Text(0.5, 1.0, 'Non-Working')
```

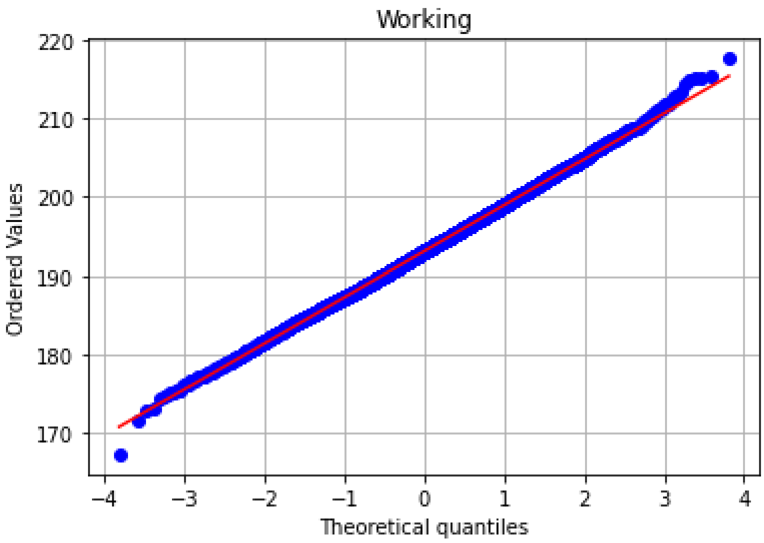


In [39]:

```
#QQ-plot to check if distribution is normal
fig, ax1 = plt.subplots()
plt.grid()
prob = stats.probplot(bs_means_w, dist=stats.norm,  fit=True, plot=ax1)
plt.title("Working")
```

Out[39]:

```
Text(0.5, 1.0, 'Working')
```



Therefore, we can conclude that the means of working and non working are normally distributed and hence our assumptions for t-test are satisfied.

In [ ]:

In [40]:

```
#2 sample t test
ttest_stat, p_val_t = stats.ttest_ind(nonworking,working)
```

In [41]:

```
p_val_t
```

Out[41]:

```
0.22607559007082925
```

In [42]:

```
p_val_t > alpha
```

Out[42]:

```
True
```

Because p value is greater than the significance level we fail to reject the null hypothesis.

Therefore , we conclude that the means of count of rentals on working and non-working days are equal and thus working day feature doesnot affect the count "

2) Weather with count of rentals

- Ho = There is no difference between means of count of rentals of different weather groups.
- Ha = There is some difference between means of count of rentals of different weather groups

To compare the means of multiple groups we can use **ANOVA**. Requirements for ANOVA:

- Each group is normally distributed
- Variance of each group is same

In [43]:

```
df.groupby(by='weather')['count'].describe()
```

Out[43]:

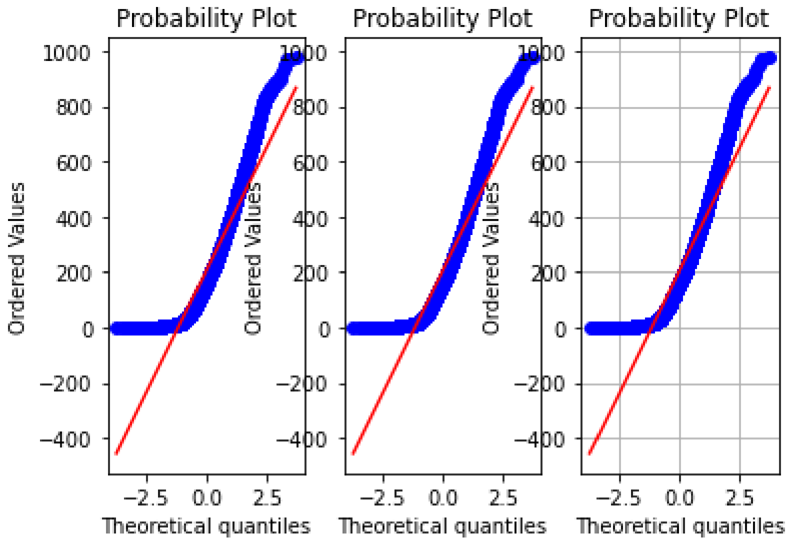
weather	count	mean	std	min	25%	50%	75%	max
1	7192.0	205.236791	187.959566	1.0	48.0	161.0	305.0	977.0
2	2834.0	178.955540	168.366413	1.0	41.0	134.0	264.0	890.0
3	859.0	118.846333	138.581297	1.0	23.0	71.0	161.0	891.0

In [44]:

```
w1 = df[df['weather'] == 1]['count']
w2 = df[df['weather'] == 2]['count']
w3 = df[df['weather'] == 3]['count']
```

In [45]:

```
fig, (ax1 ,ax2, ax3) = plt.subplots(1,3)
plt.grid()
prob1 = stats.probplot(w1, dist=stats.norm,  fit=True, plot=ax1)
prob2 = stats.probplot(w1, dist=stats.norm,  fit=True, plot=ax2)
prob3 = stats.probplot(w1, dist=stats.norm,  fit=True, plot=ax3)
```



We can observe that neither of the groups are normally distributed. Also, the variance is not same.

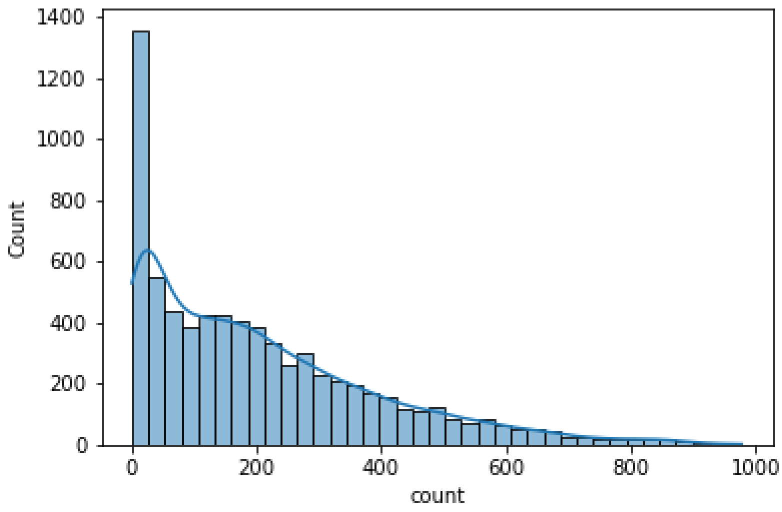
Since both requirements are not fulfilled we cannont use Anova.

In [46]:

```
sns.histplot(w1, kde=True)
```

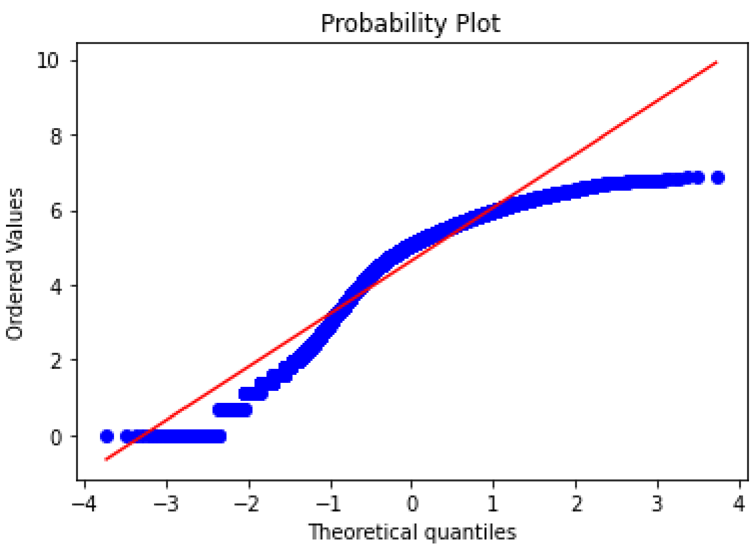
Out[46]:

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



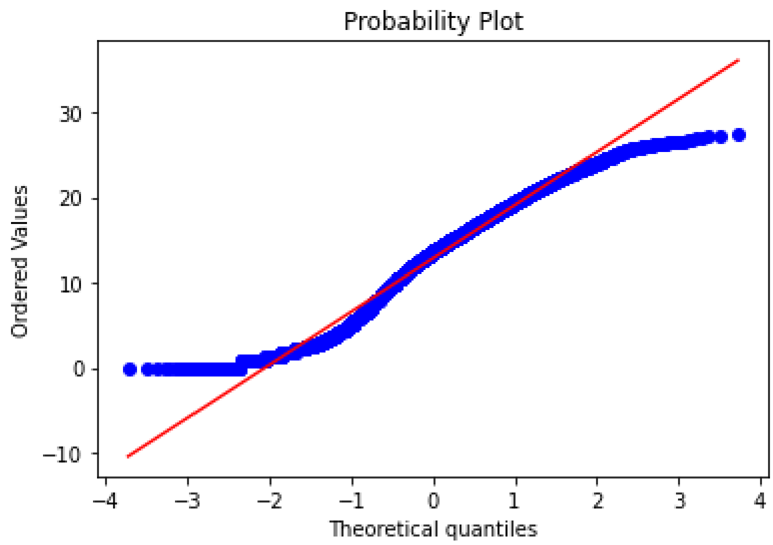
In [47]:

```
#log transform
xl = np.log(w1)
fig, ax4 = plt.subplots()
prob = stats.probplot(xl, dist='norm', plot=ax4)
```



In [48]:

```
#box-cox transform
xt , l = stats.boxcox(w1)
fig, ax4 = plt.subplots()
prob = stats.probplot(xt, dist='norm', plot=ax4)
```



Both logtransform and boxcox transform failed to tranform weather group to Gaussian.

Therefore we cannot test our null hypothesis

3) Season with count of rentals

- Ho = There is no difference between means of count of rentals of different seasons.
- Ha = There is some difference between means of count of rentals of different seasons.

To compare the means of multiple groups we can use **ANOVA**. Requirements for ANOVA:

- Each group is normally distributed
- Variance of each group is same

In [49]:

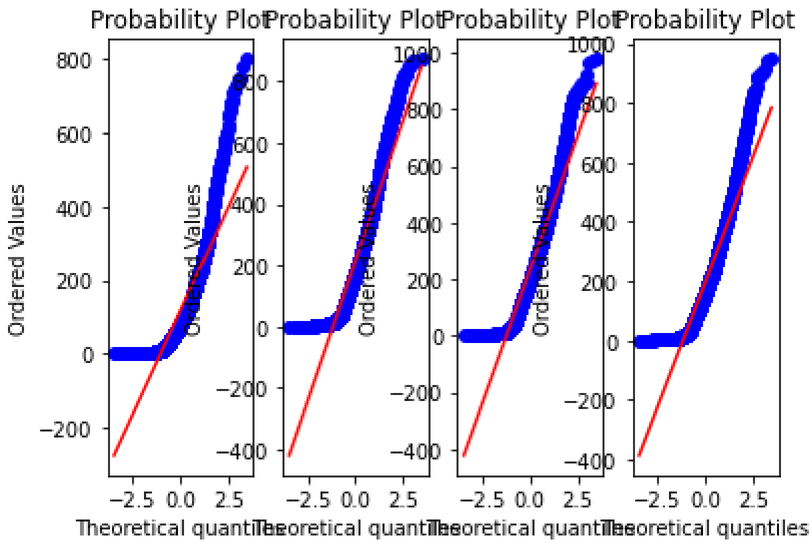
```
df.groupby(by='season')['count'].describe()
```

Out[49]:

	count	mean	std	min	25%	50%	75%	max
season								
1	2685.0	116.325512	125.293931	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

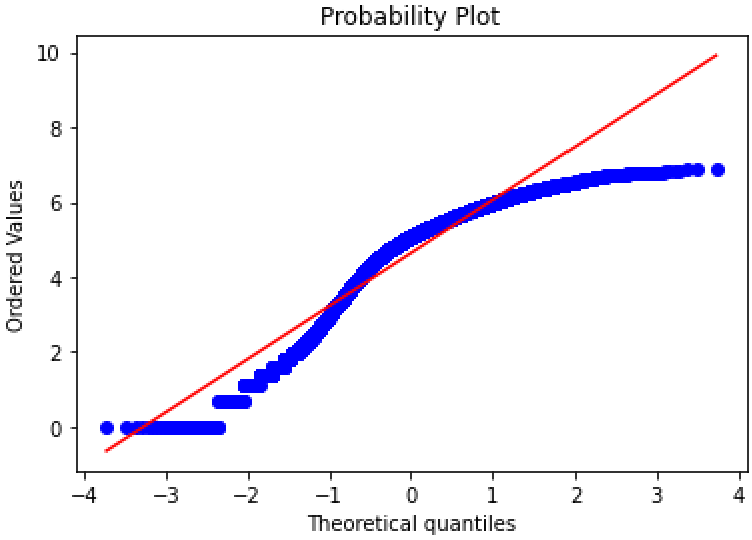
In [50]:

```
fig, ax = plt.subplots(1,4)
for i in range(1,5):
    x= df[df['season'] == i]['count']
    prob = stats.probplot(x, dist=stats.norm,  fit=True, plot=ax[i-1])
```



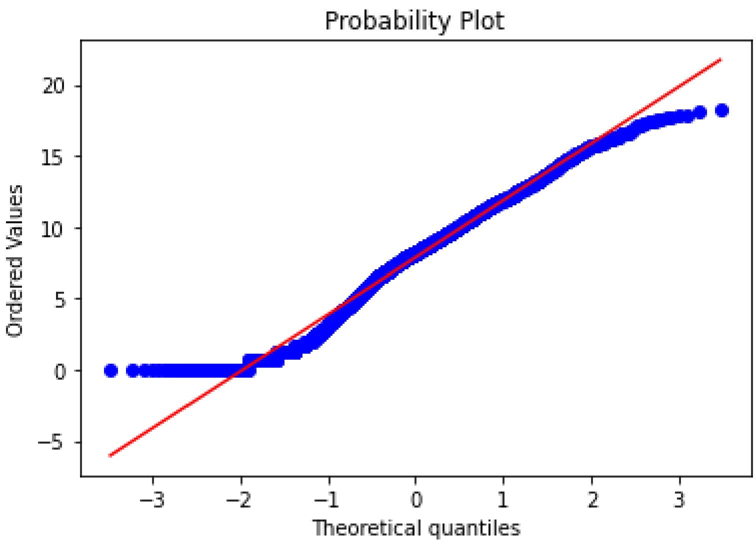
In [51]:

```
#log transform
l = np.log(df[df['season'] == 1]['count'])
fig, ax4 = plt.subplots()
prob = stats.probplot(xl, dist='norm', plot=ax4)
```



In [52]:

```
#box-cox transform
xt , l = stats.boxcox(df[df['season'] == 1]['count'])
fig, ax4 = plt.subplots()
prob = stats.probplot(xt, dist='norm', plot=ax4)
```



In [ ]:

```

```

We can observe that neither of the groups are normally distributed. Also, the variance is not same.

Since both requirements are not fulfilled we cannont use Anova.

Both logtransform and boxcox transform failed to tranform weather group to Gaussian.

Therefore we cannot test our null hypothesis

In [ ]:

```

```

4) Dependence of Weather on Season

- Ho : Season has no effect on Weather
- Ha : Season has some effect on Weather

To check the dependence we can use **chi-squared test** since both feature are categorical

In [53]:

```
cross= pd.crosstab(df['weather'],df['season'])
cross
```

Out[53]:

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

In [54]:

```
chi_t_stat, p_val_chi , dof, expected = stats.chi2_contingency(cross)
```

In [55]:

```
chi_t_stat
```

Out[55]:

46.101457310732485

In [56]:

```
p_val_chi
```

Out[56]:

2.8260014509929403e-08

In [57]:

```
p_val_chi > alpha
```

Out[57]:

False

Because our p value from chi-squared test is less than the significance level we can **reject the null**

hypothesis.

Therefore we can conclude that Weather is dependent on Season.

In [ ]:

```

```