```
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.distributions.empirical_distribution import ECDF
from scipy.stats import norm,binom,geom,t,ttest_ind,ttest_1samp,ttest_rel,chi
from scipy.stats import f, f_oneway
from scipy.stats import poisson,chi2_contingency
```

```
df = pd.read_csv("/content/drive/MyDrive/csv /yulu.csv")
df.head(10)
```

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

```
df.describe()
```

	season	holiday	workingday	weather	temp	at
count	10886.000000	10886.000000	10886.000000	10886.000000	10886.00000	10886.00
mean	2.506614	0.028569	0.680875	1.418427	20.23086	23.65
std	1.116174	0.166599	0.466159	0.633839	7.79159	8.47
min	1.000000	0.000000	0.000000	1.000000	0.82000	0.76
25%	2.000000	0.000000	0.000000	1.000000	13.94000	16.66
50%	3.000000	0.000000	1.000000	1.000000	20.50000	24.24
75%	4.000000	0.000000	1.000000	2.000000	26.24000	31.06
max	4.000000	1.000000	1.000000	4.000000	41.00000	45.45

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

```
4    2734
2    2733
```

```
3    2733
1    2686
Name: season, dtype: int64
```

```
df[["humidity","count"]].corr()
```

	humidity	count
humidity	1.000000	-0.317371
count	-0.317371	1.000000

```
df[["temp","count"]].corr()
```

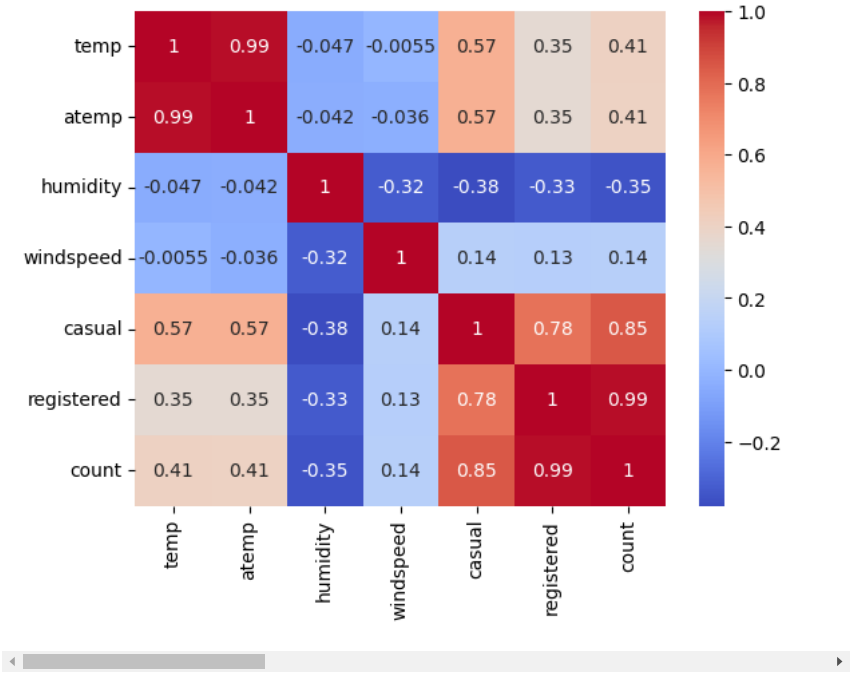
	temp	count
temp	1.000000	0.394454
count	0.394454	1.000000

```
df[["windspeed","count"]].corr()
```

	windspeed	count
windspeed	1.000000	0.101369
count	0.101369	1.000000

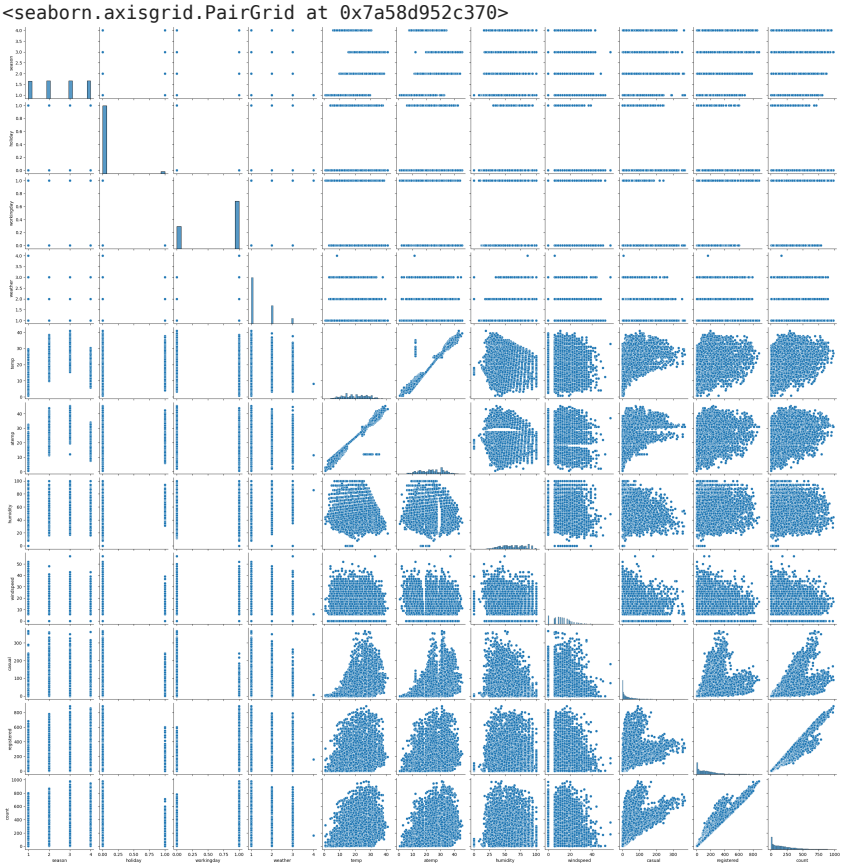
```
sns.heatmap(data=df.corr(method="spearman"),
 annot=True,
 cmap="coolwarm")
```

```
<ipython-input-40-32f9736fb7b7>:1: FutureWarning: The default value of nume
  sns.heatmap(data=df.corr(method="spearman"),
<Axes: >
```




```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7a58d952c370>
```



Hypothesis Testing

```
df.columns
```

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

Working day and non-working day dependancy on count of rides by using T-test

```
non_working_day=df[df["workingday"]==0]
working_day=df[df["workingday"]==1]
```

```
t_sat,p_value=ttest_ind(non_working_day["count"],working_day["count"])
alpha=0.01
H0="Working day is not dependent on number of rides"
Ha="Working day is dependent on number of rides"
if p_value<alpha:
  print(Ha)
else:
```

```
 print(H0)
print(p_value)
```

```
    Working day is not dependent on number of rides
    0.22644804226361348
```

```
t_sat,p_value=ttest_ind(non_working_day["registered"],working_day["registered"],alternative='greater')
alpha=0.01
H0="Working day is not dependent on number of registered rides"
Ha="Working day has more number of registered rides"
if p_value<alpha:
 print(Ha)
else:
 print(H0)
print(p_value)
```

```
    Working day is not dependent on number of registered rides
    1.0
```

```
t_sat,p_value=ttest_ind(non_working_day["casual"],working_day["casual"],alternative='greater')
alpha=0.01
H0="Working day has more number of casual rides"
Ha="Non-working day has more number of casual rides"
if p_value<alpha:
 print(Ha)
else:
 print(H0)
print(p_value)
```

```
    Non-working day has more number of casual rides
    1.78098371180272e-256
```

```
holiday=df[df["holiday"]==1]
non_holiday=df[df["holiday"]==0]
```

```
t_sat,p_value=ttest_ind(holiday["casual"],non_holiday["casual"],alternative="greater")
alpha=0.01
H0="Non-holiday day has more number of casual rides"
Ha="Holiday day has more number of casual rides"
if p_value<alpha:
 print(Ha)
else:
 print(H0)
print(p_value)
```

```
    Holiday day has more number of casual rides
    2.421530012047593e-06
```

```
clear=df[df["weather"]==1]
mist=df[df["weather"]==2]
rain=df[df["weather"]==3]
heavy_rain=df[df["weather"]==4]
```

```
f_sat,p_value=f_oneway(heavy_rain["count"],rain["count"],mist["count"],clear["count"])
alpha=0.01
H0="Weather is not dependent on number of rides"
Ha="Weather is dependent on number of rides"
if p_value<alpha:
 print(Ha)
else:
 print(H0)
print(p_value)
```

```
    Weather is dependent on number of rides
    5.482069475935669e-42
```

```
spring=df[df["season"]==1]
summer=df[df["season"]==2]
fall=df[df["season"]==3]
winter=df[df["season"]==4]
```

```
f_sat,p_value=f_oneway(spring["count"],summer["count"],fall["count"],winter["count"])
alpha=0.01
H0="Season is not dependent on number of rides"
Ha="Season is dependent on number of rides"
if p_value<alpha:
 print(Ha)
else:
```

```
 print(H0)
print(p_value)
```

```
Season is dependent on number of rides
6.164843386499654e-149
```

```
ttest_ind(fall["count"],summer["count"],alternative="greater")
```

```
Ttest_indResult(statistic=3.6407918229052068, pvalue=0.00013715780586249322)
```

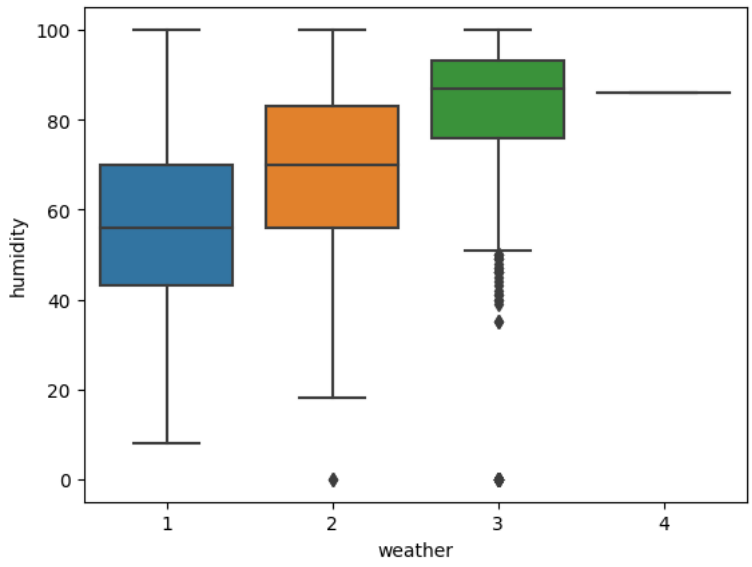
```
# Datatype of following attributes needs to changed to proper data type

# datetime - to datetime
# season - to categorical
# holiday - to categorical
# workingday - to categorical
# weather - to categorical

df['datetime'] = pd.to_datetime(df['datetime'])
cat_cols= ['season', 'holiday', 'workingday', 'weather']
for col in cat_cols:
    df[col] = df[col].astype('object')
```

```
df.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
```

```
sns.boxplot(data=df,x='weather',y='humidity')
```

```
<Axes: xlabel='weather', ylabel='humidity'>
```




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

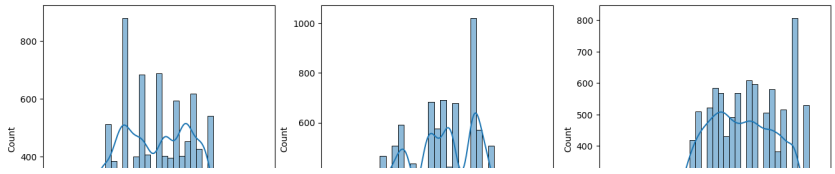
```
(Timestamp('2011-01-01 00:00:00'), Timestamp('2012-12-19 23:00:00'))
```

```
# number of unique values in each categorical columns
df[cat_cols].melt().groupby(['variable', 'value'])[['value']].count()
```

variable	value	value
holiday	0	10575
	1	311
season	1	2686
	2	2733
	3	2733
	4	2734
weather	1	7192
	2	2834
workingday	0	3474

Univariate Analysis

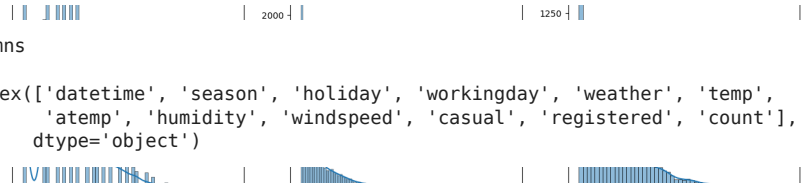
```
# understanding the distribution for numerical variables
num_cols = ['temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered','count']

fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(3):
        sns.histplot(df[num_cols[index]], ax=axis[row, col], kde=True)
        index += 1

plt.show()
sns.histplot(df[num_cols[-1]], kde=True)
plt.show()
```



- casual, registered and count somewhat looks like Log Normal Distrinution

  temp, atemp and humidity looks like they follows the Normal Distribution

  windspeed follows the binomial distribution



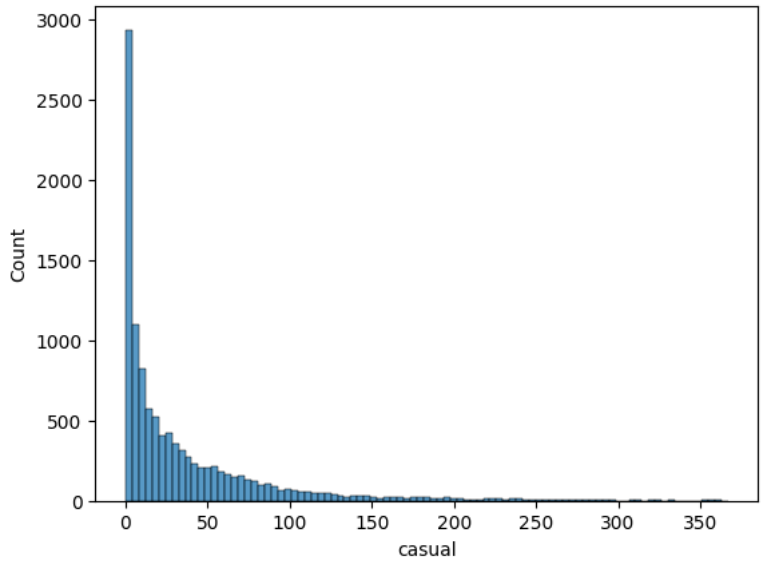
```
df.columns
```

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

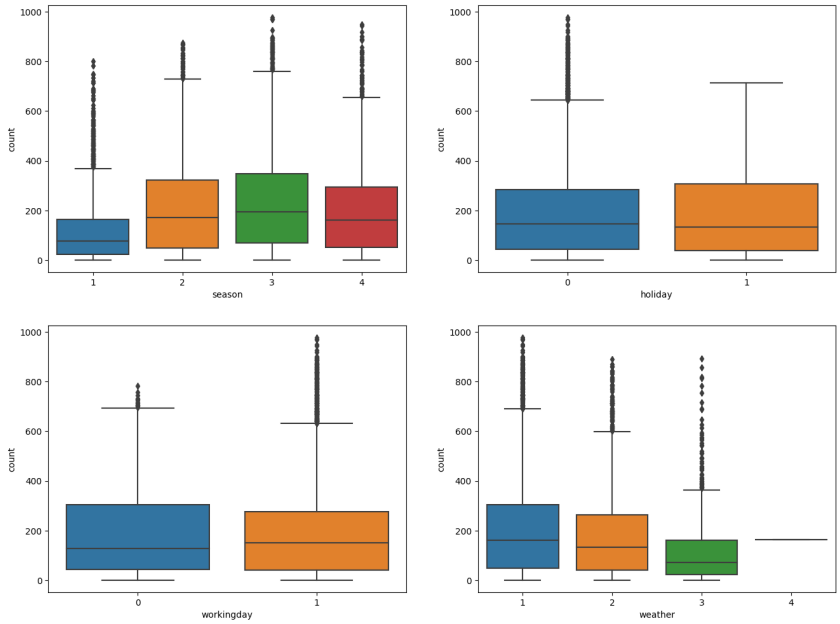


```
sns.histplot(df['casual'])
```

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




```
# plotting categorical variables againt count using boxplots
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))

index = 0
for row in range(2):
    for col in range(2):
        sns.boxplot(data=df, x=cat_cols[index], y='count', ax=axis[row, col])
        index += 1

plt.show()
```




```
# In summer and fall seasons more bikes are rented as compared to other seasons.
# Whenever its a holiday more bikes are rented.
# It is also clear from the workingday also that whenever day is holiday or weekend, slightly more bikes were rented.
# Whenever there is rain, thunderstorm, snow or fog, there were less bikes were rented.


# Null Hypothesis (H0): Weather is independent of the season
# Alternate Hypothesis (H1): Weather is not independent of the season
# Significance level (alpha): 0.05
```

Chi Square

```
data_table = pd.crosstab(df['season'], 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.chi2_contingency(data_table)
# expected_values = val[3]
# expected_values
```

```
Chi2ContingencyResult(statistic=3814.4941536976935, pvalue=0.0, dof=264, expected_freq=array([[ 5.42825648,
0.24673893,  0.24673893,  0.24673893,  0.24673893,
         0.49347786,  0.98695572,  1.97391145,  1.48043358,  1.72717252,
         3.70108396,  2.46738931,  3.94782289,  4.44130075,  9.12934044,
         9.12934044,  7.89564578,  9.6228183 , 12.09020761, 15.05107477,
        16.0380305 , 14.80433584, 19.73911446, 15.79129157, 25.6608488 ,
        22.94672056, 26.40106559, 31.82932207, 36.76410068, 31.3358442 ,
        31.08910527, 41.20540143, 45.39996326, 32.81627779, 47.62061363,
        37.25757854, 35.2836671 , 60.94451589, 36.02388389, 31.82932207,
        57.73690979, 46.88039684, 39.72496785, 53.7890869 , 45.89344112,
        40.46518464, 55.26952049, 51.32169759, 35.77714496, 41.45214036,
        43.91952967, 39.97170678, 50.5814808 , 49.84126401, 25.6608488 ,
        31.58258313, 62.42494948, 60.69777696, 27.14128238, 32.32279993,
        51.07495866, 63.90538306, 26.40106559, 33.55649458, 48.11409149,
        48.60756936, 27.88149917, 35.53040603, 60.2042991 , 47.3738747 ,
        36.76410068, 14.80433584, 42.93257395, 49.34778615, 77.96950211,
        18.50541981,  0.98695572,  9.86955723, 71.30755098, 90.79992651,
        37.01083961,  0.98695572,  0.24673893,  0.49347786, 50.5814808 ,
        79.94341356,  0.24673893,  0.24673893, 36.51736175],
       [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
```

```
               0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
               3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
               9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
              16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
              23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
              31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
              37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
              58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
              41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
              44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
              32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
              51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
              49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
              37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
              18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
              37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
              81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
             [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
               0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
               3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
               9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
              16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
              23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
              31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
              37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
              58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
              41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
              44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
              32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
              51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
              49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
              37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
              18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
              37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
              81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
             [ 5.5252618 ,  0.25114826,  0.25114826,  0.25114826,  0.25114826,
               0.50229653,  1.00459306,  2.00918611,  1.50688958,  1.75803785,
               3.76722306,  2.51148264,  4.01837222,  4.52066075,  9.2924857 ,
```

```
nrows, ncols = 4, 4
dof = (nrows-1)*(ncols-1)
print("degrees of freedom: ", dof)
alpha = 0.05


chi_sqr = sum([(o-e)**2/e for o, e in zip(data_table.values, expected_values)])
chi_sqr_statistic = chi_sqr[0] + chi_sqr[1]
print("chi-square test statistic: ", chi_sqr_statistic)

critical_val = stats.chi2.ppf(q=1-alpha, df=dof)
print(f"critical value: {critical_val}")

p_val = 1-stats.chi2.cdf(x=chi_sqr_statistic, df=dof)
print(f"p-value: {p_val}")

if p_val <= alpha:
    print("\nSince p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that\
    Weather is dependent on the season.")
else:
    print("Since p-value is greater than the alpha 0.05, We do not reject the Null Hypothesis")
```

```
 degrees of freedom:  9
 chi-square test statistic:  44.09441248632364
 critical value: 16.918977604620448
 p-value: 1.3560001579371317e-06

 Since p-value is less than the alpha 0.05, We reject the Null Hypothesis. Meaning that    Weather is dependent on the se
```

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

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

```
data_table1 = pd.crosstab(df['season'], df['humidity'])
data_table1
```

humidity	0	8	10	12	13	14	15	16	17	18	...	88	89	90	91	92	93
season																	
1	22	1	1	1	1	2	4	6	3	2	...	25	0	1	0	2	114
2	0	0	0	0	0	0	0	1	2	2	...	162	18	1	0	0	29
3	0	0	0	0	0	0	0	0	1	0	...	63	123	0	1	0	1
4	0	0	0	0	0	0	0	1	0	3	...	118	9	2	0	0	61

4 rows × 89 columns

```
stats.chi2_contingency(data_table1)
```

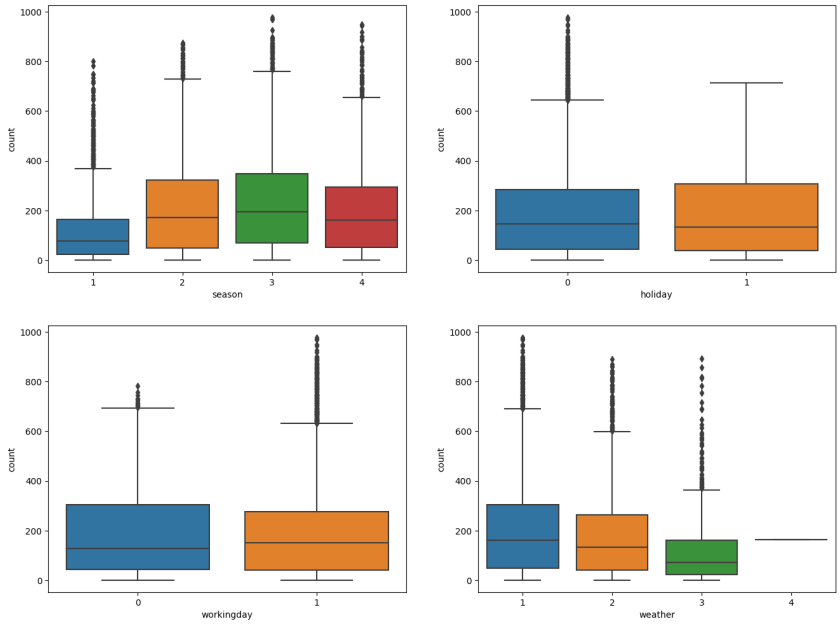
```
Chi2ContingencyResult(statistic=3814.4941536976935, pvalue=0.0, dof=264, expected_freq=array([[ 5.42825648,
0.24673893,  0.24673893,  0.24673893,  0.24673893,
         0.49347786,  0.98695572,  1.97391145,  1.48043358,  1.72717252,
         3.70108396,  2.46738931,  3.94782289,  4.44130075,  9.12934044,
         9.12934044,  7.89564578,  9.6228183 , 12.09020761, 15.05107477,
        16.0380305 , 14.80433584, 19.73911446, 15.79129157, 25.6608488 ,
        22.94672056, 26.40106559, 31.82932207, 36.76410068, 31.3358442 ,
        31.08910527, 41.20540143, 45.39996326, 32.81627779, 47.62061363,
        37.25757854, 35.2836671 , 60.94451589, 36.02388389, 31.82932207,
        57.73690979, 46.88039684, 39.72496785, 53.7890869 , 45.89344112,
        40.46518464, 55.26952049, 51.32169759, 35.77714496, 41.45214036,
        43.91952967, 39.97170678, 50.5814808 , 49.84126401, 25.6608488 ,
        31.58258313, 62.42494948, 60.69777696, 27.14128238, 32.32279993,
        51.07495866, 63.90538306, 26.40106559, 33.55649458, 48.11409149,
        48.60756936, 27.88149917, 35.53040603, 60.2042991 , 47.3738747 ,
        36.76410068, 14.80433584, 42.93257395, 49.34778615, 77.96950211,
        18.50541981,  0.98695572,  9.86955723, 71.30755098, 90.79992651,
        37.01083961,  0.98695572,  0.24673893,  0.49347786, 50.5814808 ,
        79.94341356,  0.24673893,  0.24673893, 36.51736175],
       [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
         0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
         3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
         9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
        16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
        23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
        31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
        37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
        58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
        41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
        44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
        32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
        51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
        49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
        37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
        18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
        37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
        81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
       [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
         0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
         3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
         9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
        16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
        23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
        31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
        37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
        58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
        41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
        44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
        32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
        51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
        49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
        37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
        18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
        37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
        81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
       [ 5.5252618 ,  0.25114826,  0.25114826,  0.25114826,  0.25114826,
         0.50229653,  1.00459306,  2.00918611,  1.50688958,  1.75803785,
         3.76722396,  2.51148264,  4.01837222,  4.52066875,  9.29248576,
```

```
df.columns
```

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

```
# plotting categorical variables againt count using boxplots
fig, axis = plt.subplots(nrows=2, ncols=2, figsize=(16, 12))
```

```
index = 0
for row in range(2):
    for col in range(2):
        sns.boxplot(data=df, x=cat_cols[index], y='count', ax=axis[row, col])
        index += 1

plt.show()
```

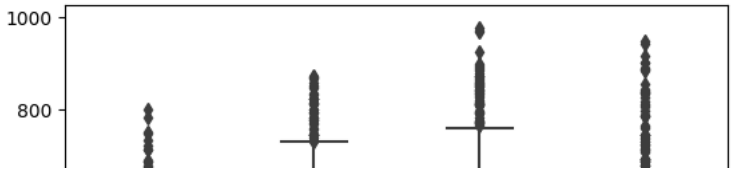



```
df['season'].unique()
```

```
array([1, 2, 3, 4], dtype=object)
```

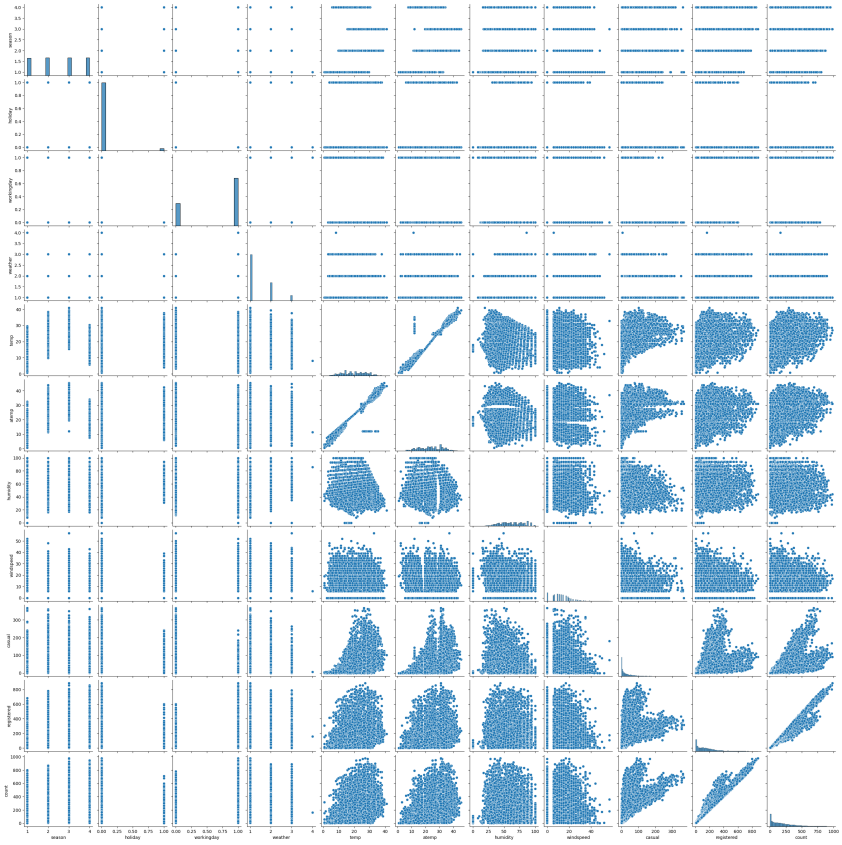
```
sns.boxplot(data=df,x='season',y='count')
```

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



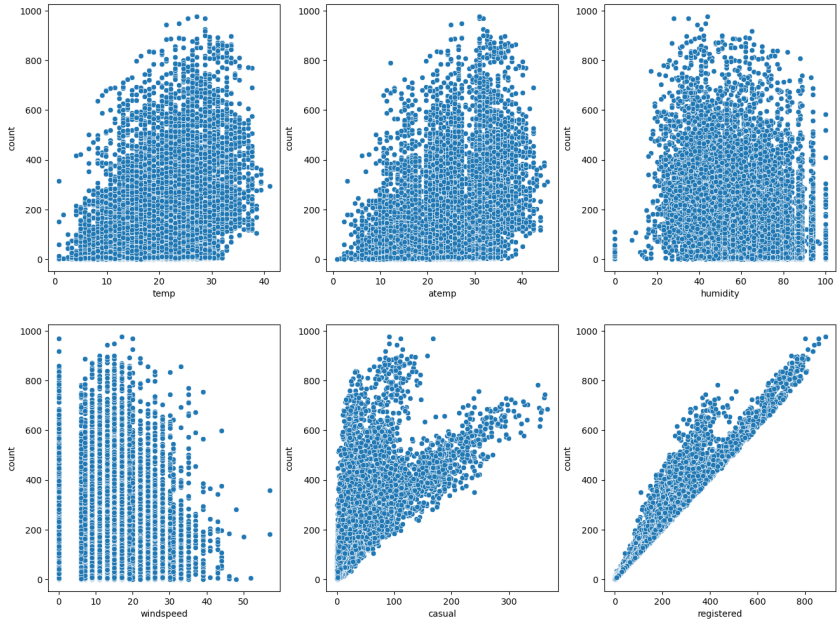

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7a58e47c0ca0>
```



```
# plotting numerical variables againt count using scatterplot
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(16, 12))
```

```
index = 0
for row in range(2):
    for col in range(3):
        sns.scatterplot(data=df, x=num_cols[index], y='count', ax=axis[row, col])
        index += 1

plt.show()
# Whenever the humidity is less than 20, number of bikes rented is very very low.
# Whenever the temperature is less than 10, number of bikes rented is less.
# Whenever the windspeed is greater than 35, number of bikes rented is less.
```

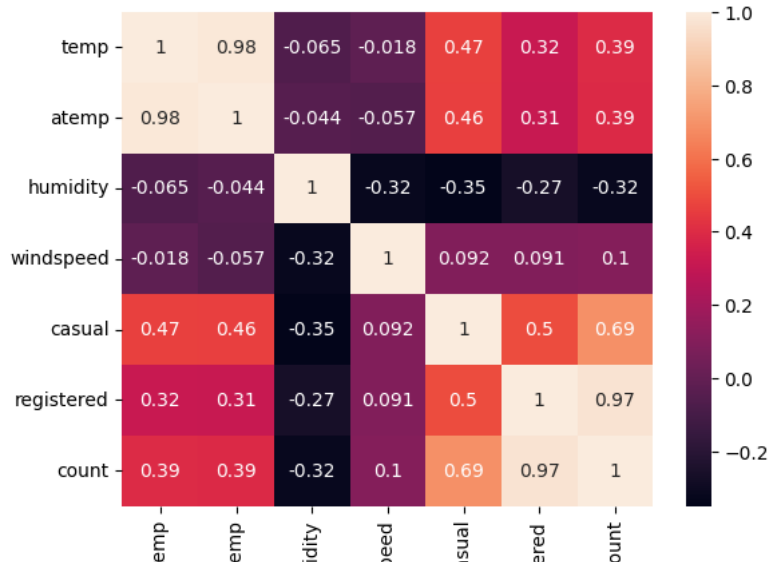



```
df.corr()['count']
```

```
<ipython-input-29-c6e37b628cdf>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In
  df.corr()['count']
temp          0.394454
atemp         0.389784
humidity     -0.317371
windspeed     0.101369
casual        0.690414
registered    0.970948
count         1.000000
Name: count, dtype: float64
```

```
sns.heatmap(df.corr(), annot=True)
plt.show()
```

```
<ipython-input-30-6522c2b4e5f9>:1: FutureWarning: The default value of nume
  sns.heatmap(df.corr(), annot=True)
```




```
df.columns

# data_table = pd.crosstab(df['season'], df['count'])
# data_table
```

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

```
data_table = pd.crosstab(df['season'], df['humidity'])
data_table
```

humidity	0	8	10	12	13	14	15	16	17	18	...	88	89	90	91	92	93
season																	
1	22	1	1	1	1	2	4	6	3	2	...	25	0	1	0	2	114
2	0	0	0	0	0	0	0	1	2	2	...	162	18	1	0	0	29
3	0	0	0	0	0	0	0	0	1	0	...	63	123	0	1	0	1
4	0	0	0	0	0	0	0	1	0	3	...	118	9	2	0	0	61

4 rows × 89 columns

```
from scipy.stats.contingency import chi2_contingency
chi2_contingency(data_table)

#session and humidity not depending
```

```
Chi2ContingencyResult(statistic=3814.4941536976935, pvalue=0.0, dof=264, expected_freq=array([[ 5.42825648,
0.24673893,  0.24673893,  0.24673893,  0.24673893,
         0.49347786,  0.98695572,  1.97391145,  1.48043358,  1.72717252,
         3.70108396,  2.46738931,  3.94782289,  4.44130075,  9.12934044,
         9.12934044,  7.89564578,  9.6228183 , 12.09020761, 15.05107477,
        16.0380305 , 14.80433584, 19.73911446, 15.79129157, 25.6608488 ,
        22.94672056, 26.40106559, 31.82932207, 36.76410068, 31.3358442 ,
        31.08910527, 41.20540143, 45.39996326, 32.81627779, 47.62061363,
        37.25757854, 35.2836671 , 60.94451589, 36.02388389, 31.82932207,
        57.73690979, 46.88039684, 39.72496785, 53.7890869 , 45.89344112,
        40.46518464, 55.26952049, 51.32169759, 35.77714496, 41.45214036,
        43.91952967, 39.97170678, 50.5814808 , 49.84126401, 25.6608488 ,
        31.58258313, 62.42494948, 60.69777696, 27.14128238, 32.32279993,
        51.07495866, 63.90538306, 26.40106559, 33.55649458, 48.11409149,
        48.60756936, 27.88149917, 35.53040603, 60.2042991 , 47.3738747 ,
        36.76410068, 14.80433584, 42.93257395, 49.34778615, 77.96950211,
        18.50541981,  0.98695572,  9.86955723, 71.30755098, 90.79992651,
        37.01083961,  0.98695572,  0.24673893,  0.49347786, 50.5814808 ,
        79.94341356,  0.24673893,  0.24673893, 36.51736175],
       [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
         0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
         3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
         9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
        16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
        23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
        31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
        37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
        58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
        41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
```

```
              44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
              32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
              51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
              49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
              37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
              18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
              37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
              81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
             [ 5.52324086,  0.2510564 ,  0.2510564 ,  0.2510564 ,  0.2510564 ,
               0.50211281,  1.00422561,  2.00845122,  1.50633842,  1.75739482,
               3.76584604,  2.51056403,  4.01690244,  4.51901525,  9.2890869 ,
               9.2890869 ,  8.03380489,  9.79119971, 12.30176373, 15.31444057,
              16.31866618, 15.06338416, 20.08451222, 16.06760977, 26.10986588,
              23.34824545, 26.86303509, 32.38627595, 37.40740401, 31.88416315,
              31.63310674, 41.92641925, 46.1943781 , 33.39050156, 48.45388572,
              37.90951681, 35.90106559, 62.01093147, 36.6542348 , 32.38627595,
              58.74719824, 47.70071652, 40.42008084, 54.73029579, 46.69649091,
              41.17325005, 56.23663421, 52.21973177, 36.40317839, 42.17747566,
              44.68803968, 40.67113724, 51.46656256, 50.71339335, 26.10986588,
              32.13521955, 63.51726989, 61.75987507, 27.6162043 , 32.88838876,
              51.96867536, 65.0236083 , 26.86303509, 34.14367077, 48.95599853,
              49.45811134, 28.36937351, 36.15212199, 61.25776226, 48.20282932,
              37.40740401, 15.06338416, 43.68381407, 50.21128054, 79.33382326,
              18.8292302 ,  1.00422561, 10.04225611, 72.55530039, 92.3887562 ,
              37.65846041,  1.00422561,  0.2510564 ,  0.50211281, 51.46656256,
              81.34227448,  0.2510564 ,  0.2510564 , 37.1563476 ],
             [ 5.5252618 ,  0.25114826,  0.25114826,  0.25114826,  0.25114826,
               0.50229653,  1.00459306,  2.00918611,  1.50688958,  1.75803785,
               3.76722396,  2.51148264,  4.01837222,  4.52066875,  9.29248576,
```

```
df.describe(include='all')
```

```
<ipython-input-60-174ba9bf1a5c>:1: FutureWarning: Treating datetime data as
  df.describe(include='all')
```

	datetime	season	holiday	workingday	weather	temp	atem
count	10886	10886.0	10886.0	10886.0	10886.0	10886.00000	10886.0000(
unique	10886	4.0	2.0	2.0	4.0	NaN	Na
top	2011-01-01 00:00:00	4.0	0.0	1.0	1.0	NaN	Na
freq	1	2734.0	10575.0	7412.0	7192.0	NaN	Na
first	2011-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	Na
last	2012-12-19 23:00:00	NaN	NaN	NaN	NaN	NaN	Na
mean	NaN	NaN	NaN	NaN	NaN	20.23086	23.6550{
std	NaN	NaN	NaN	NaN	NaN	7.79159	8.4746(
min	NaN	NaN	NaN	NaN	NaN	0.82000	0.7600(
25%	NaN	NaN	NaN	NaN	NaN	13.94000	16.6650(

```
sns.countplot(data=df,x="season")
```

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

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

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

Insights

```
# In summer and fall seasons more bikes are rented as compared to other seasons.
# Whenever its a holiday more bikes are rented.
# It is also clear from the workingday also that whenever day is holiday or weekend, slightly more bikes were rented.
# Whenever there is rain, thunderstorm, snow or fog, there were less bikes were rented.
# Whenever the humidity is less than 20, number of bikes rented is very very low.
# Whenever the temperature is less than 10, number of bikes rented is less.
# Whenever the windspeed is greater than 35, number of bikes rented is less
```

Recommendations

```
# In summer and fall seasons the company should have more bikes in stock to be rented.
# Because the demand in these seasons is higher as compared to other seasons.
# With a significance level of 0.01, workingday has no effect on the number of bikes being rented.
# In very low humid days, company should have less bikes in the stock to be rented.
# Whenever temprature is less than 10 or in very cold days, company should have less bikes.
# Whenever the windspeed is greater than 35 or in thunderstorms, company should have less bikes in stock to be rented.
```