yulu-business-case

April 4, 2024

1 Problem Statement:

Yulu, India's foremost micro-mobility service provider, is experiencing significant declines in its revenues. To address this concern, Yulu has engaged a consulting company to investigate the underlying factors affecting the demand for their shared electric cycles in the Indian market. To identify significant variables that predict the demand for shared electric cycles in the Indian market and understand their impact on cycle demand.

[11]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```python
df = pd.read_csv("bike_sharing.csv")
df.head()
```

[4]:
```
              datetime  season  holiday  workingday  weather  temp   atemp  \
0  2011-01-01 00:00:00       1        0           0        1  9.84  14.395
1  2011-01-01 01:00:00       1        0           0        1  9.02  13.635
2  2011-01-01 02:00:00       1        0           0        1  9.02  13.635
3  2011-01-01 03:00:00       1        0           0        1  9.84  14.395
4  2011-01-01 04:00:00       1        0           0        1  9.84  14.395

   humidity  windspeed  casual  registered  count
0        81        0.0       3          13     16
1        80        0.0       8          32     40
2        80        0.0       5          27     32
3        75        0.0       3          10     13
4        75        0.0       0           1      1
```

[5]:
```python
df.shape
```

[5]:
```
(10886, 12)
```

[6]:
```python
df.describe()
```

[6]:
```
             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
```

1.1 There are no missing values in the dataset

[10]:
```
df.isnull().sum()
```

[10]:
```
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
```

1.2 Converting the following attributes to proper data types

datetime to datetime

season to categorical

holiday to categorical

workingday to categorical

weather to categorical

[13]:
```
df['datetime'] = pd.to_datetime(df['datetime'])
```

[17]:
```
categ_cols = ['season', 'holiday', 'workingday', 'weather']
for col in categ_cols:
    df[col] = df[col].astype('object')
```

[19]:
```
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  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
dtypes: datetime64[ns](1), float64(3), int64(4), object(4)
memory usage: 1020.7+ KB
```

1.3 Unique attributes for each attribute

[22]:
```
print(df.apply(lambda x : x.unique()))
```

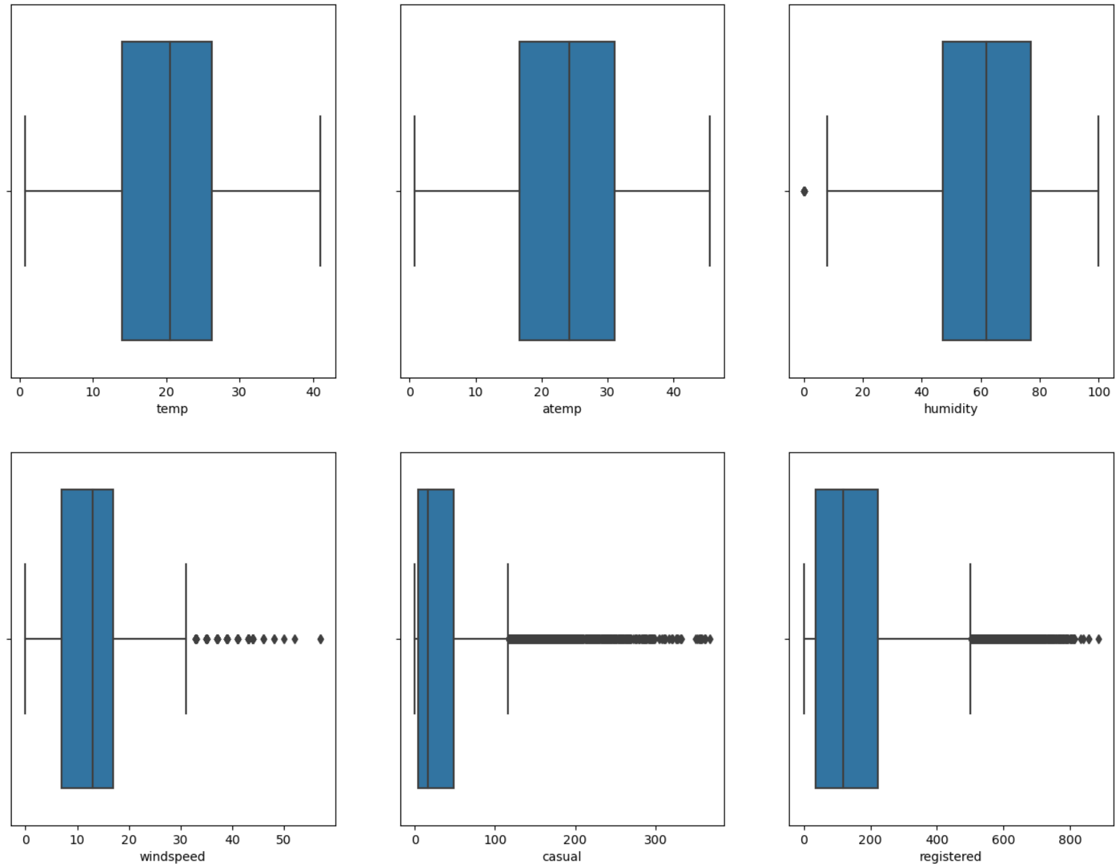
```
datetime      [2011-01-01T00:00:00.000000000, 2011-01-01T01:…
season                                              [1, 2, 3, 4]
holiday                                                   [0, 1]
workingday                                                [0, 1]
weather                                             [1, 2, 3, 4]
temp          [9.84, 9.02, 8.2, 13.12, 15.58, 14.76, 17.22, …
```

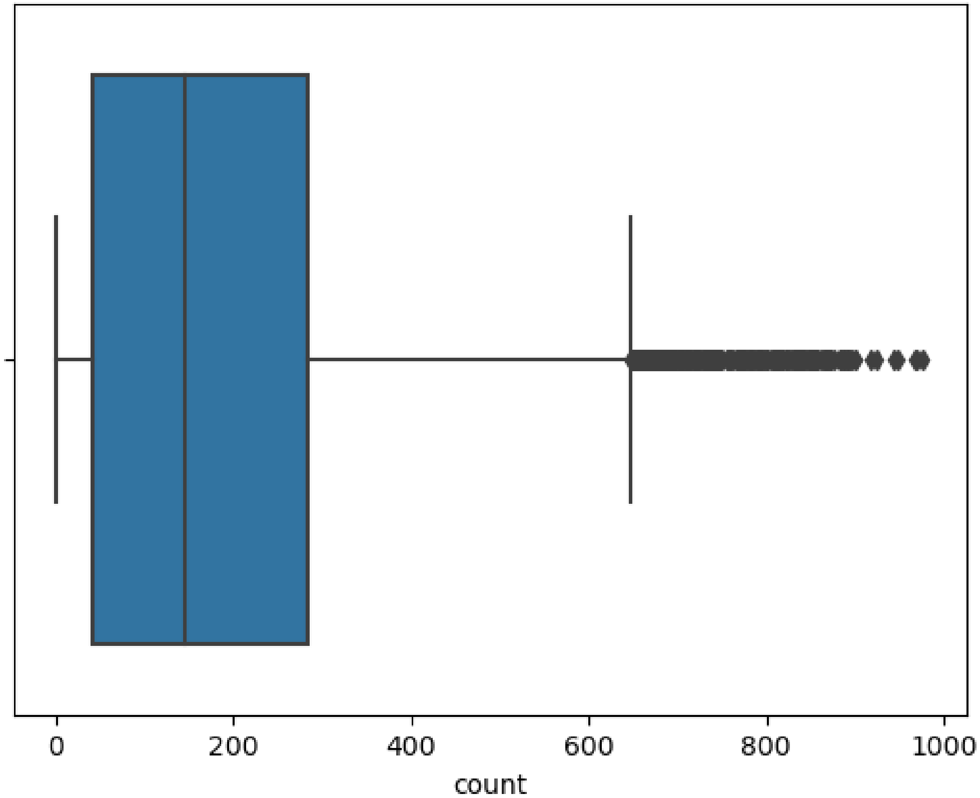
```
atemp          [14.395, 13.635, 12.88, 17.425, 19.695, 16.665…
humidity       [81, 80, 75, 86, 76, 77, 72, 82, 88, 87, 94, 1…
windspeed      [0.0, 6.0032, 16.9979, 19.0012, 19.9995, 12.99…
casual         [3, 8, 5, 0, 2, 1, 12, 26, 29, 47, 35, 40, 41,…
registered     [13, 32, 27, 10, 1, 0, 2, 7, 6, 24, 30, 55, 47…
count          [16, 40, 32, 13, 1, 2, 3, 8, 14, 36, 56, 84, 9…
dtype: object
```

1.4 Univariate Analysis

1.4.1 Plotting box plots to detect outliers in the data

[63]:
```
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.boxplot(x=df[num_cols[index]], ax=axis[row, col])
        index += 1

plt.show()
sns.boxplot(x=df[num_cols[-1]])
plt.show()
```


0 10 20 30 40
temp
0 10 20 30 40
atemp
0 20 40 60 80 100
humidity
0 10 20 30 40 50
windspeed
0 100 200 300
casual
0 200 400 600 800
registered

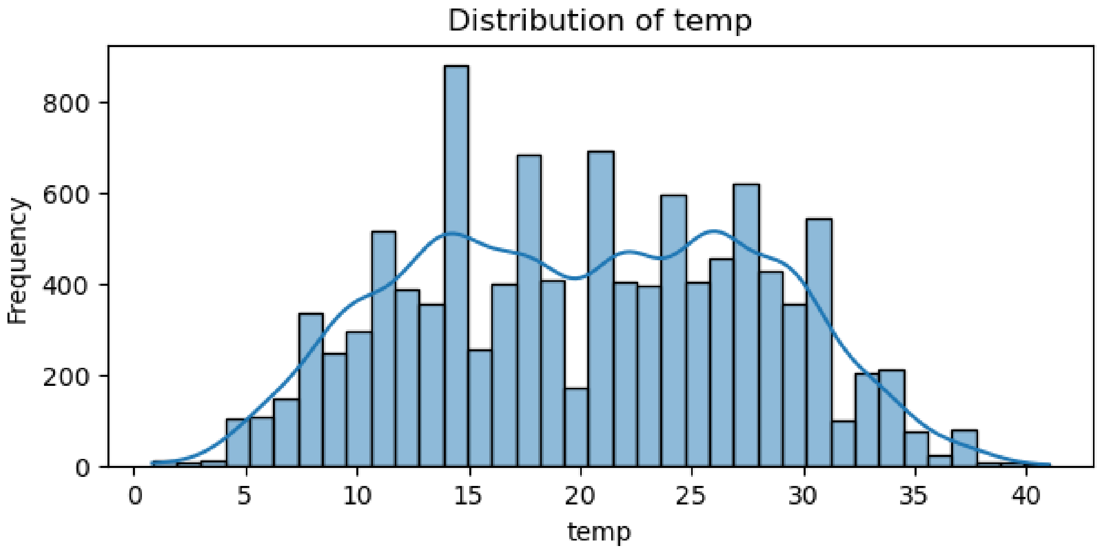


HUMIDITY CASUAL COUNT & REGISTRED have outliers

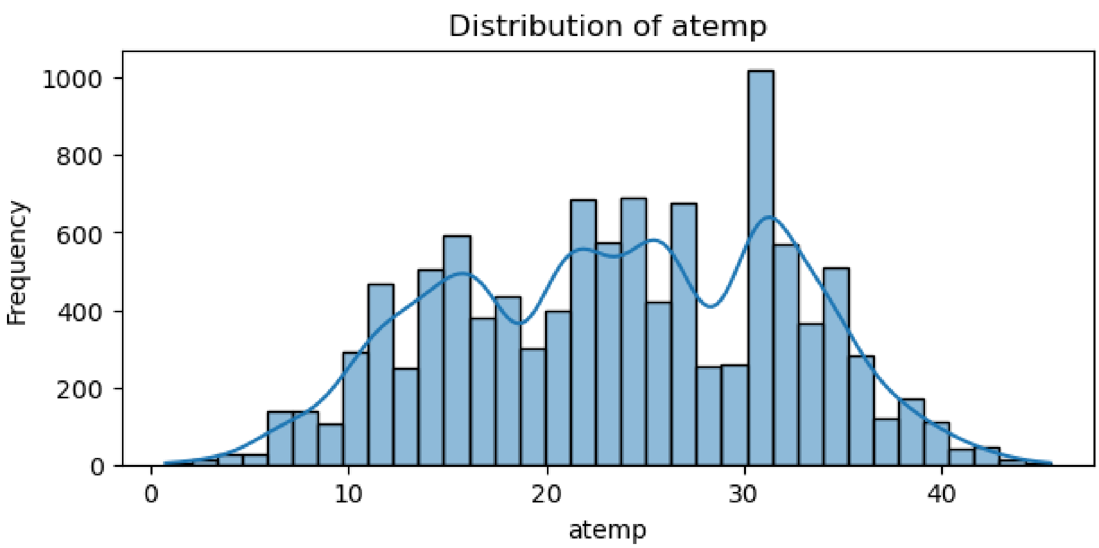
1.4.2 Let's plot distribution plots for all continuous variables and barplots/countplots for all categorical variables in the Yulu dataset

1.4.3 Univariate analysis - Distribution plots for Continuous variables

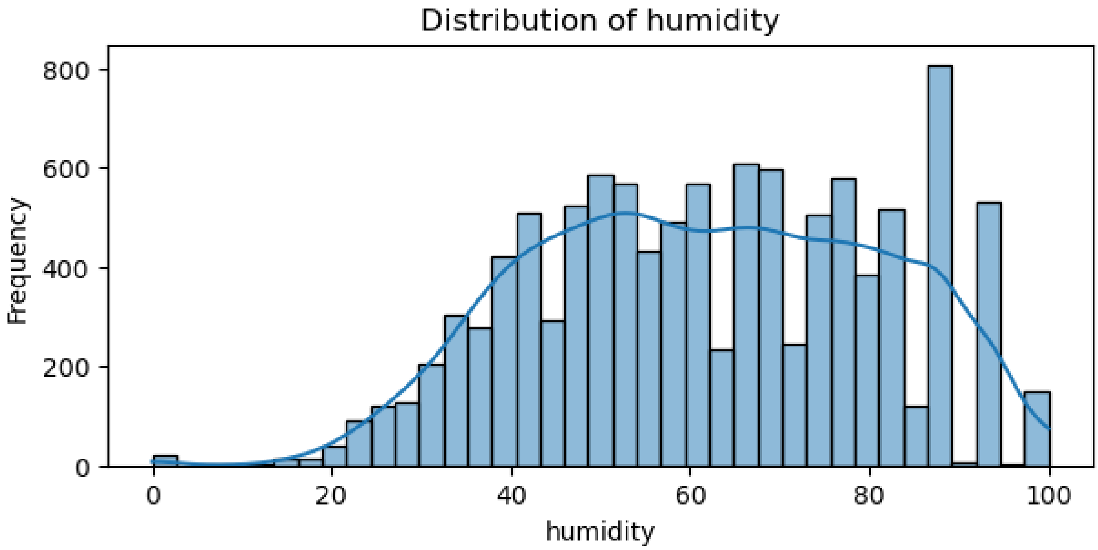
[62]:
```
continous_var = ['temp', 'atemp', 'humidity', 'windspeed', 'casual',␣
↪'registered', 'count' ]

for var in continous_var:
    plt.figure(figsize=(7,3))
    sns.histplot(df[var], kde='True')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.title(f'Distribution of {var}')
    plt.show()
```

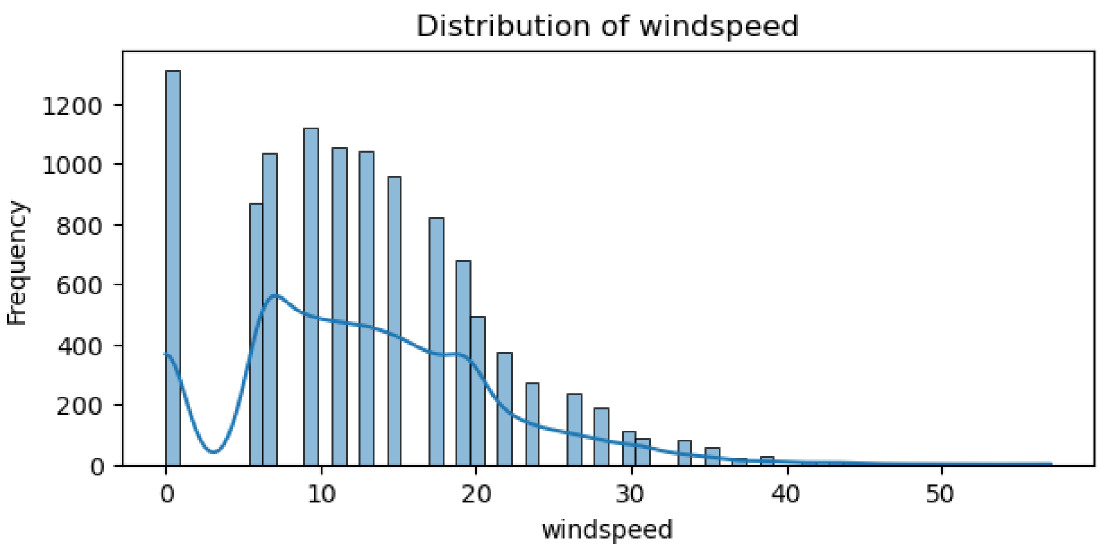


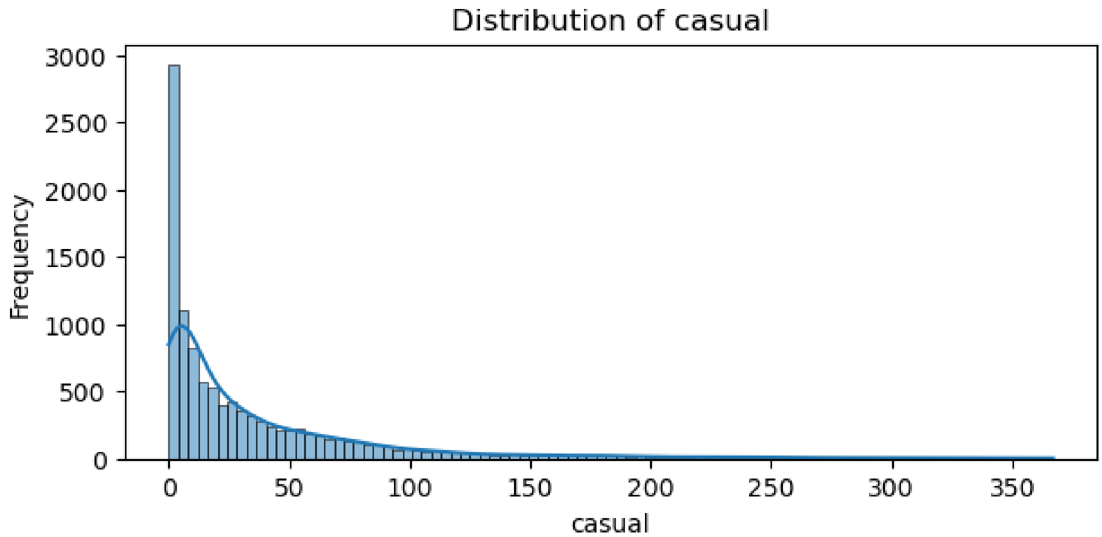
Distribution of temp
Frequency
temp
0
200
400
600
800
0
5
10
15
20
25
30
35
40

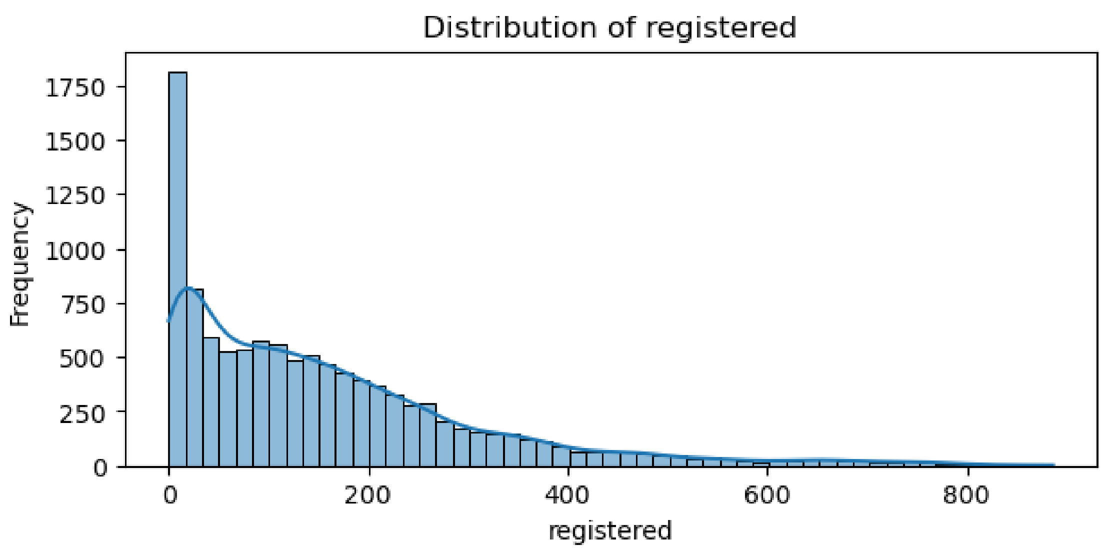


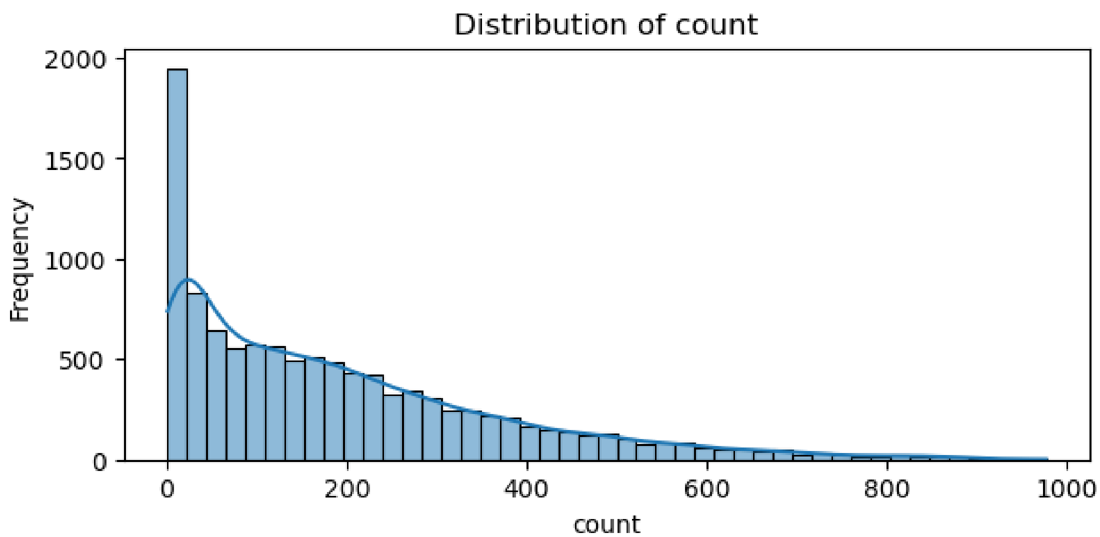
Distribution of atemp
Frequency
atemp
0
200
400
600
800
1000
0
10
20
30
40


Distribution of humidity
800
600
400
200
0
Frequency
0
20
40
60
80
100
humidity


Distribution of windspeed
1200
1000
800
600
400
200
0
Frequency
0
10
20
30
40
50
windspeed

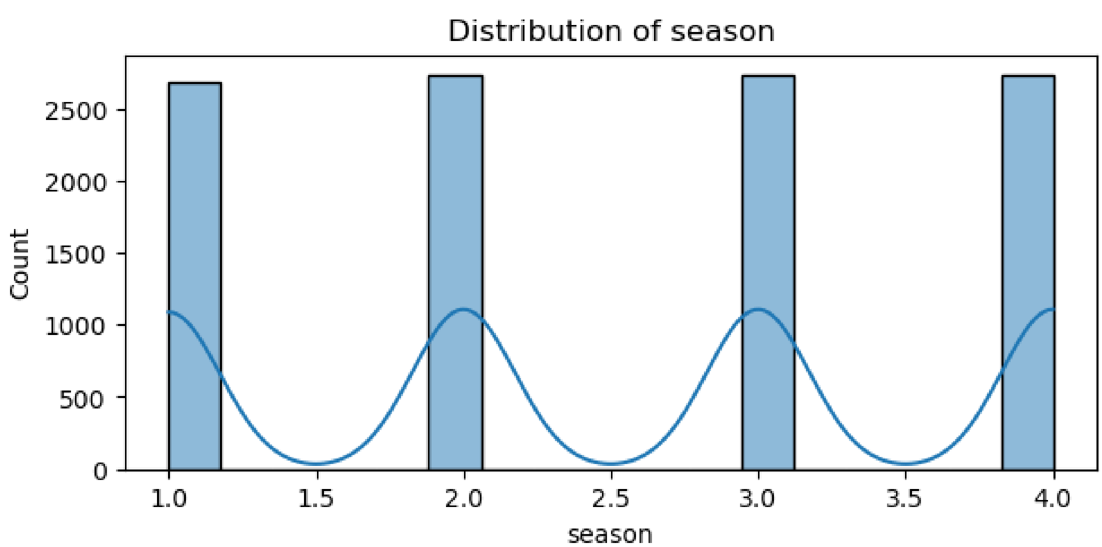

Distribution of casual
Frequency
casual
0
500
1000
1500
2000
2500
3000
50
100
150
200
250
300
350

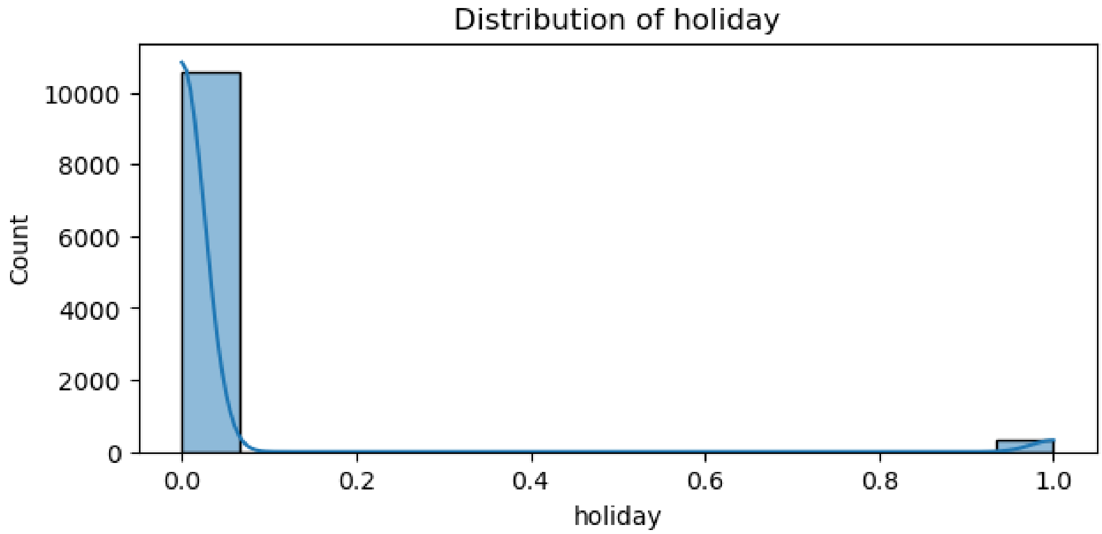

Distribution of registered
Frequency
registered
0
250
500
750
1000
1250
1500
1750
200
400
600
800

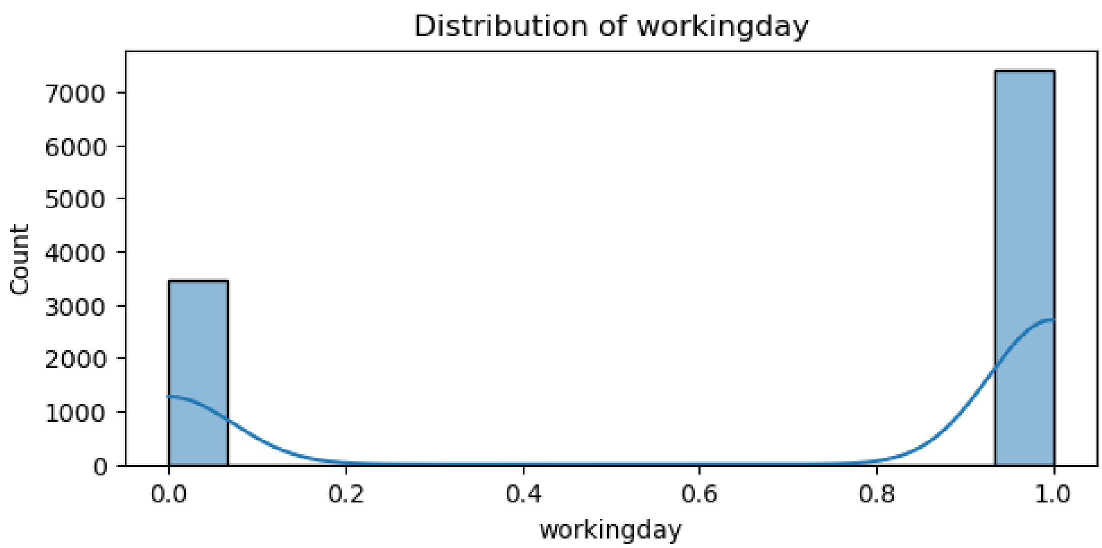


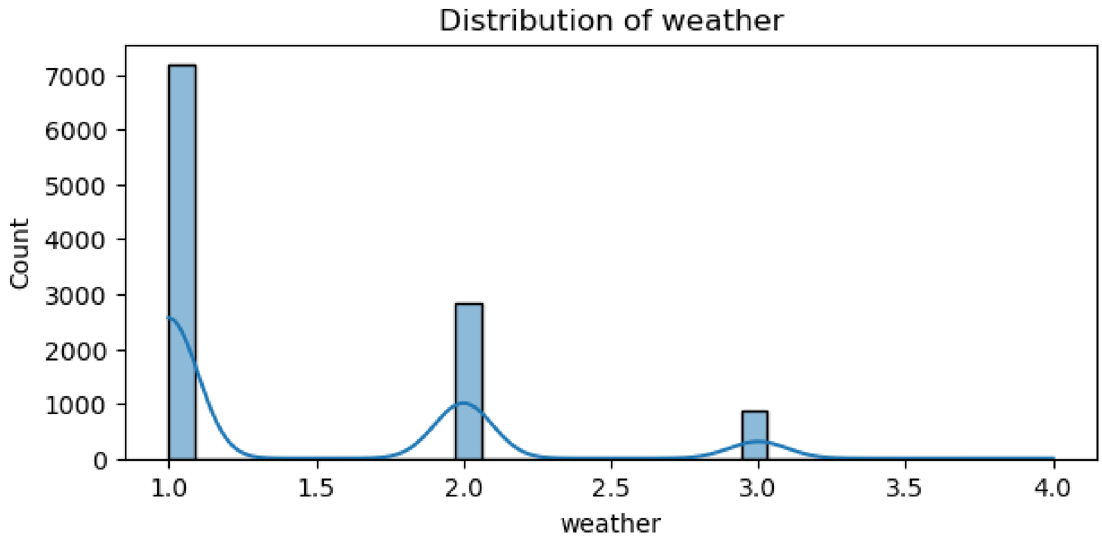
1.4.4 Univariate analysis - Distribution plots for Categorical variables

[64]:
```
categorical_var = ['season', 'holiday', 'workingday', 'weather']

for var in categorical_var:
    plt.figure(figsize=(7,3))
    sns.histplot(df[var], kde='True')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.title(f'Distribution of {var}')
    plt.show()
```




Distribution of holiday
Count
10000
8000
6000
4000
2000
0
0.0
0.2
0.4
0.6
0.8
1.0
holiday


Distribution of workingday
Count
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
workingday



1.4.5 Bivariate Analysis (Relationships between important variables such as workday and count, season and count, weather and count.

For bivariate analysis, we'll examine the relationships between important variables such as workday and count, season and count, and weather and count. We can use scatter plots or box plots to visualize these relationships.

1.4.6 Insights

Higher demand for bike rentals is observed during the summer and fall seasons compared to other seasons.

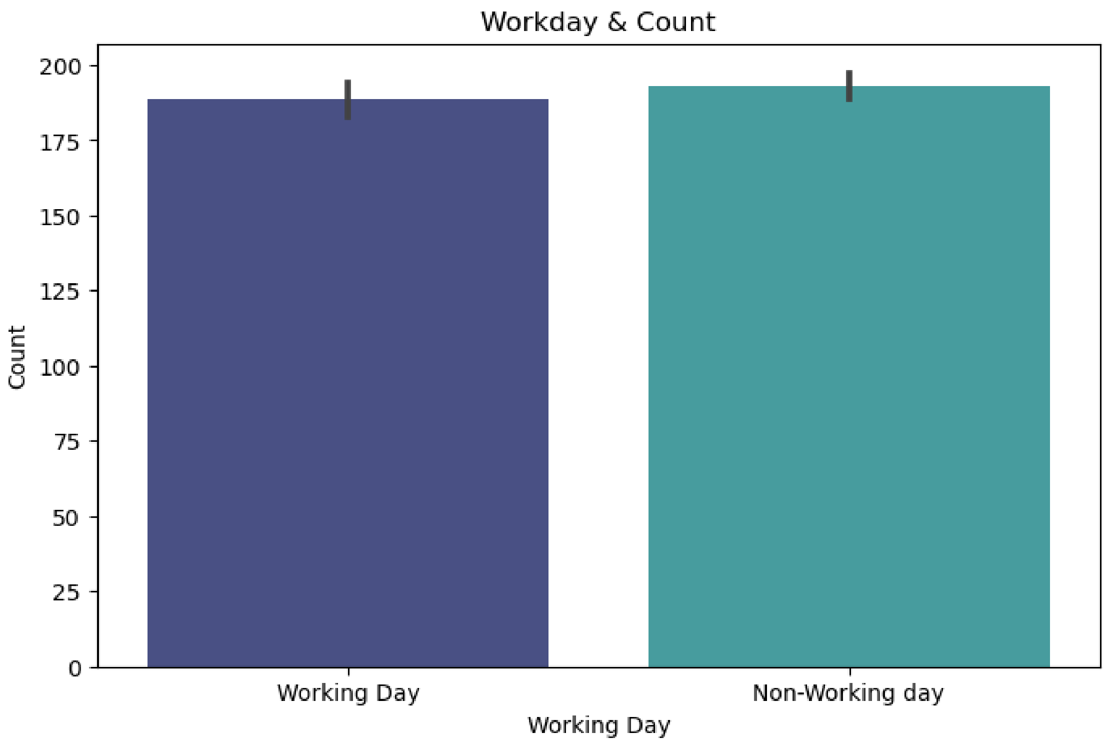
There is a notable increase in bike rentals during holidays.

Rental activity is slightly higher on holidays or weekends compared to regular working days.

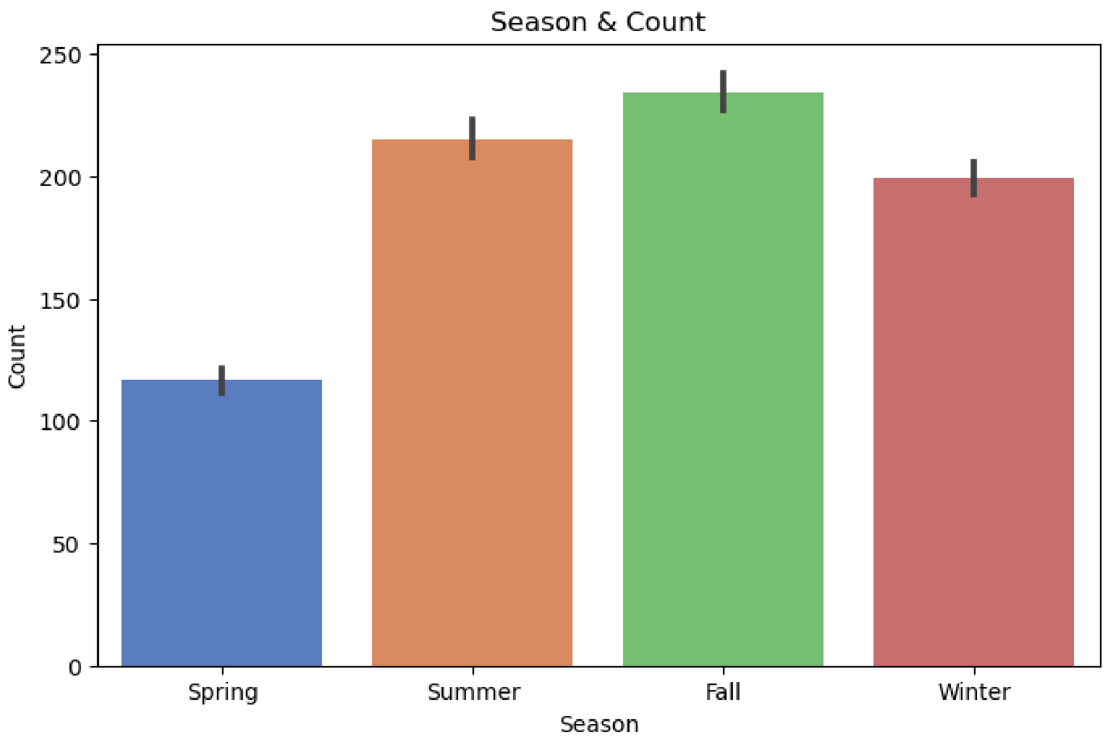
Conversely, during rainy, stormy, snowy, or foggy weather conditions, the demand for bike rentals decreases.

The customers may use the booking service for a variety of purposes beyond commuting to work, such as leisure activities or errands, which are not necessarily tied to working days.

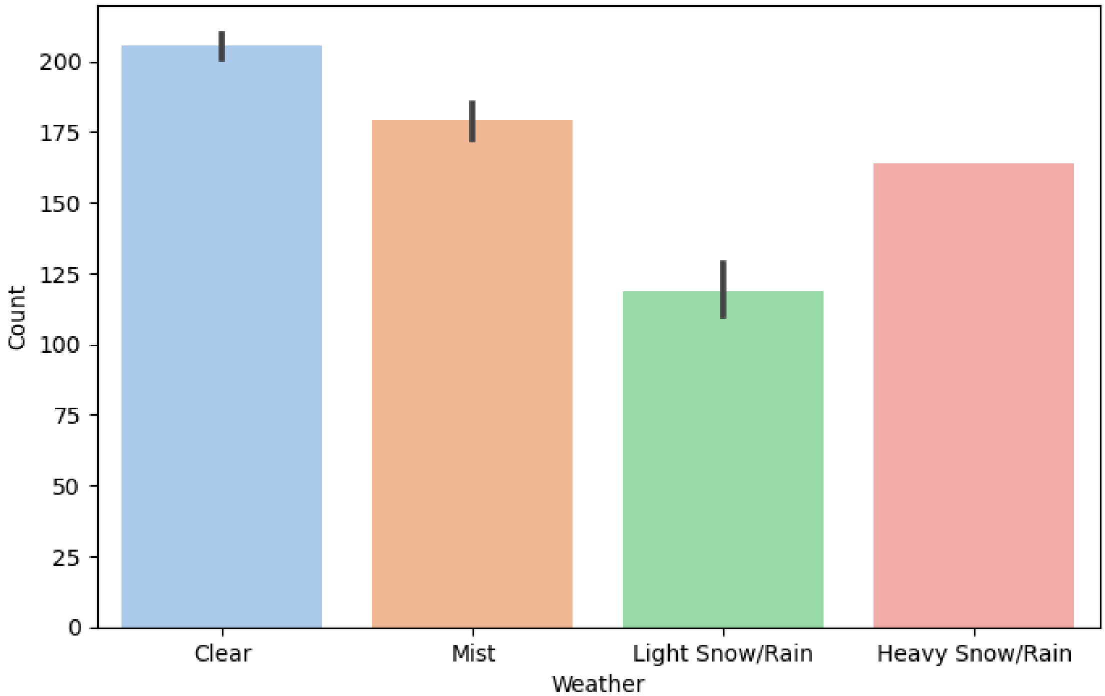
[125]:
```
plt.figure(figsize=(8,5))
sns.barplot(data = df, x='workingday', y='count', palette='mako')
plt.xlabel('Working Day')
plt.ylabel('Count')
plt.title("Workday & Count")
plt.xticks([0,1], ['Working Day', 'Non-Working day'])
plt.show()
```



[118]:
```
plt.figure(figsize=(8,5))
sns.barplot(data = df, x='season', y='count', palette="muted")
plt.xlabel('Season')
plt.ylabel('Count')
plt.title("Season & Count")
plt.xticks([0,1,2,3], ['Spring', 'Summer', 'Fall', 'Winter'])
plt.show()
```



[122]:
```
plt.figure(figsize=(8,5))
sns.barplot(data = df, x='weather', y='count', palette="pastel")
plt.xlabel('Weather')
plt.ylabel('Count')
plt.xticks([0,1,2,3], ['Clear', 'Mist', 'Light Snow/Rain', 'Heavy Snow/Rain'])
plt.show()
```



1.4.7 Insights

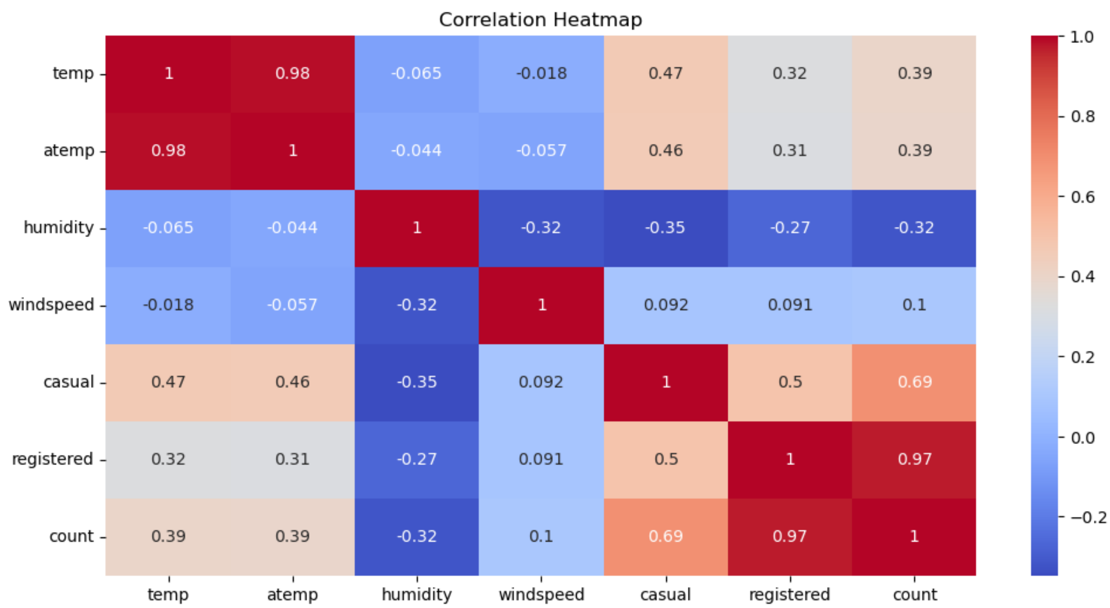
Clear Weather: The number of bookings increases during clear weather. Mist Weather: The number of bookings is medium during misty weather conditions. Heavy Rainfall: The number of bookings is medium during heavy rainfall. Light Snow: The number of bookings decreases during light snowfall.

The weather conditions influence bike bookings, other factors could also be at play.

1.4.8 Calculate the correlation matrix

[157]:
```
corr_matrix = df.corr()
plt.figure(figsize=(12, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', )
plt.title('Correlation Heatmap')
plt.show()
```

```
/var/folders/bc/byp79cv56b53hq4lpz78hyvh0000gn/T/ipykernel_70065/1542193356.py:2
: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  corr_matrix = df.corr()
```



1.4.9 Interpretation of the correlation matrix

temp and atemp: These two variables have a very high positive correlation of approximately 0.98, indicating a strong linear relationship. This is expected since "temp" and "atemp" are both measures of temperature, and it's natural for them to be highly correlated.

humidity and windspeed: These variables have weak correlations with other variables. "humidity" has a weak negative correlation with "temp" and "atemp", which makes sense as higher temperatures usually correlate with lower humidity. "windspeed" also has weak correlations with other variables.

casual, registered, and count: These variables show positive correlations with each other. It's expected that "casual" and "registered" users would be positively correlated with the total count of bike rentals ("count").

count and other variables: The total count of bike rentals ("count") shows moderate positive correlations with "temp" and "atemp", indicating that higher temperatures may be associated with higher bike rental counts. It also shows a weaker positive correlation with "casual" and "registered" users, which suggests that both types of users contribute to the overall demand for bike rentals.

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

1.6 Hypothesis Testing 1

Null Hypothesis (H0): There is no significant difference in the number of bike rides between weekdays and weekends.

Alternate Hypothesis (H1): There is a significant difference in the number of bike rides between weekdays and weekends.

We'll use a **2-sample independent t-test** because we want to compare the means of two independent groups (weekdays and weekends) to determine if they are significantly different.

We'll set the significance level (alpha) to 0.05 (5%)

[147]:
```
# Split the data into weekdays and weekends
weekdays_data = df[df['workingday'] == 1]['count']
weekends_data = df[df['workingday'] == 0]['count']

# Perform 2-sample independent t-test
t_statistic, p_value = stats.ttest_ind(weekdays_data, weekends_data)

alpha = 0.05

print(f"t-statistic: {t_statistic}")
print(f"p-value: {p_value}")

# Draw conclusions
if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant difference in the␣
 ↪number of bike rides between weekdays and weekends.")
else:
    print("Fail to Reject Null Hypothesis: There is no significant difference␣
 ↪in the number of bike rides between weekdays and weekends.")
```

```
t-statistic: 1.2096277376026694
p-value: 0.22644804226361348
Fail to Reject Null Hypothesis: There is no significant difference in the number
of bike rides between weekdays and weekends.
```

1.6.1 Results

Based on the 2-sample independent t-test performed with a significance level of 0.05 (alpha = 5%), the following conclusions can be drawn:

t-statistic: 1.2096 p-value: 0.2264 Since the p-value (0.2264) is greater than the significance level (0.05), we fail to reject the null hypothesis. Therefore, we do not have sufficient evidence to conclude that there is a significant difference in the number of bike rides between weekdays and weekends.

Inferences and Recommendations:

The analysis indicates that there is no significant difference in the number of bike rides between weekdays and weekends.

This suggests that the demand for bike rides remains relatively consistent throughout the week, regardless of whether it's a weekday or weekend.

Yulu may continue to offer their bike rental services with a consistent approach across weekdays and weekends.

1.7 Check if the demand of bicycles on rent is the same for different Weather conditions?

1.8 Hypothesis Testing 2

Null Hypothesis (H0): The mean demand for bicycles on rent is the same across all weather conditions.

Alternate Hypothesis (H1): At least one of the weather conditions has a different mean demand for bicycles on rent.

We'll use a **one-way ANOVA test** because we have one categorical independent variable (weather conditions) with more than two levels and one continuous dependent variable (demand for bicycles).

We'll set the significance level (alpha) to 0.05 (5%)

[152]:
```
from scipy.stats import f_oneway
# Extract demand data for different weather conditions
weather_1 = df[df['weather'] == 1]['count']
weather_2 = df[df['weather'] == 2]['count']
weather_3 = df[df['weather'] == 3]['count']
weather_4 = df[df['weather'] == 4]['count']

# Perform one-way ANOVA test
f_statistic, p_value = f_oneway(weather_1, weather_2, weather_3, weather_4)

alpha = 0.05

print(f"One-way ANOVA Test - F-statistic: {f_statistic}, p-value: {p_value}",␣
 ↪'\n')

if p_value <= alpha:
    print("Reject Null Hypothesis: There are significant differences in the␣
 ↪mean demand across different weather conditions.")
else:
    print("Fail to Reject Null Hypothesis: There is no significant difference␣
 ↪in the mean demand across different weather conditions.")
```

```
One-way ANOVA Test - F-statistic: 65.53024112793271, p-value:
5.482069475935669e-42

Reject Null Hypothesis: There are significant differences in the mean demand
across different weather conditions.
```

1.8.1 Results

We conclude that there are significant differences in the mean demand for bicycles on rent across different weather conditions. This suggests that weather conditions have a significant impact on the demand for bicycles, indicating that people's preferences or behaviors regarding bicycle rentals may vary depending on the weather conditions.

1.9 Check if the demand of bicycles on rent is the same for different Seasons?

1.10 Hypothesis Testing 3

Null Hypothesis (H0): The mean demand for bicycles on rent is the same across all seasons.

Alternate Hypothesis (H1): At least one of the seasons has a different mean demand for bicycles on rent.

We'll use a **one-way ANOVA test** because we have one categorical independent variable (weather conditions) with more than two levels and one continuous dependent variable (demand for bicycles).

We'll set the significance level (alpha) to 0.05 (5%)

[155]:
```python
from scipy.stats import f_oneway
# Extract demand data for different season conditions
season_1 = df[df['season'] == 1]['count']
season_2 = df[df['season'] == 2]['count']
season_3 = df[df['season'] == 3]['count']
season_4 = df[df['season'] == 4]['count']

# Perform one-way ANOVA test
f_statistic, p_value = f_oneway(season_1, season_2, season_3, season_4)

alpha = 0.05

print(f"One-way ANOVA Test - F-statistic: {f_statistic}, p-value: {p_value}",
 ↪'\n')

if p_value <= alpha:
    print("Reject Null Hypothesis: There are significant differences in the
 ↪mean demand across different seasons.")
else:
    print("Fail to Reject Null Hypothesis: There is no significant difference
 ↪in the mean demand across different seasons.")
```

```
One-way ANOVA Test - F-statistic: 236.94671081032106, p-value:
6.164843386499654e-149

Reject Null Hypothesis: There are significant differences in the mean demand
across different seasons.
```

1.10.1 Results

We conclude that there are significant differences in the mean demand for bicycles on rent across different seasons. This suggests that seasons have a significant impact on the demand for bicycles, indicating that people's preferences or behaviors regarding bicycle rentals vary across different seasons.

1.11 Check if the Weather conditions are significantly different during different Seasons?

1.12 Hypothesis Testing 4

Null Hypothesis (H0): There is no significant association between weather conditions and seasons

Alternate Hypothesis (H1): There is a significant association between weather conditions and seasons.

We'll use the **chi-square test** to determine if there is a significant association between weather conditions and seasons.

We'll set the significance level (alpha) to 0.05 (5%)

[150]:
```python
# Encode the categorical variables (Weather and Season)
encoded_data = df[['weather', 'season']]

# Perform chi-square test of independence
chi2_stat, p_value, _, _ = chi2_contingency(pd.
 ↪crosstab(encoded_data['weather'], encoded_data['season']))

alpha = 0.05

print(f"Chi-square Statistic: {chi2_stat}")
print(f"P-value: {p_value}")

if p_value <= alpha:
    print("Reject Null Hypothesis: There is a significant association between␣
 ↪weather conditions and seasons.")
else:
    print("Fail to Reject Null Hypothesis: There is no significant association␣
 ↪between weather conditions and seasons.")
```

```
Chi-square Statistic: 49.15865559689363
P-value: 1.5499250736864862e-07
Reject Null Hypothesis: There is a significant association between weather
conditions and seasons.
```

1.12.1 Results

we conclude that there is a significant association between weather conditions and seasons. This indicates that weather conditions and seasons are not independent of each other, suggesting that certain weather conditions may be more prevalent during specific seasons.