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.

Business Problem 💡

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 📊

The company collected the hourly rental data of customers who has rented the electric bike from the yulu.

The dataset has the following features:

Column	Description
datetime	datetime
season	season (1: spring, 2: summer, 3: fall, 4: winter)
holiday	whether day is a holiday or not (extracted from http://dchr.dc.gov/page/holiday-schedule)
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

Importing Required Libraries 🤝

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
sns.set(style="whitegrid")
bi_palette = ["#2fcdfd", "#ffae42"]
three_set_palette = ["#ffffff", "#2fcdfd", "#ffae42"]
four_set_palette = ["#ffffff", "#2fcdfd", "#ffae42", "#5c5c5c"]
five_set_palette = ["#ffffff", "#2fcdfd", "#ffae42", "#5c5c5c", "#ff0000"]
six_set_palette = ["#ffffff", "#2fcdfd", "#ffae42", "#5c5c5c", "#ff0000", "#ffc0cb"]
```

Read Dataset 🔍

In [3]:
```python
df = pd.read_csv('../data/bike_sharing.txt', sep=',')
df.sample(5)
```

Out[3]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual
7634	2012-05-17 11:00:00	2	0	1	1	23.78	27.275	49	15.0013	77
304	2011-01-14 03:00:00	1	0	1	1	4.10	6.820	54	7.0015	0
6804	2012-04-01 19:00:00	2	0	0	1	21.32	25.000	55	15.0013	110
10574	2012-12-07 00:00:00	4	0	1	1	9.84	12.880	70	7.0015	3
5164	2011-12-09 06:00:00	4	0	1	1	9.02	12.880	80	6.0032	1

In [4]:

```
print("Shape of the data: ", df.shape)
print("The Given Dataset has {} rows and {} columns".format(df.shape[0], df.shape[1]))
print("Columns: ", df.columns.to_list())
```

```
Shape of the data:  (10886, 12)
The Given Dataset has 10886 rows and 12 columns
Columns:  ['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp', 'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count']
```

In [5]:

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

Out[5]:

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

🕵️ Shape and Structure:

- The dataset comprises 10,886 rows and 12 columns, representing a substantial volume of transactional data.
- Each row corresponds to a total number bike rental for a specific hour interval.
- Data is provided for the time period of 2011-01-01 00:00:00 and 2012-12-19 23:00:00

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
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
```

In [7]:

```python
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
```

In [8]:

```python
df.duplicated().sum()
```

Out[8]:

```
0
```

🕵️ Dataset Information:

- **Data Consistency**: All columns have the same non-null count, indicating no missing values in the dataset.
- **Data Types**: Columns are classified into integer, float and object types.

In [9]:

```python
df['datetime'] = pd.to_datetime(df['datetime']) # convert to datetime

df['date'] = df['datetime'].dt.date # extract date
df['date'] = df['date'].astype('datetime64[ns]')

## Converting the data types of the columns to category
for col in ['season','holiday','workingday','weather']:
    df[col] = df[col].astype('category')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  category      
 2   holiday     10886 non-null  category      
 3   workingday  10886 non-null  category      
 4   weather     10886 non-null  category      
 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  date        10886 non-null  datetime64[ns]
dtypes: category(4), datetime64[ns](2), float64(3), int64(4)
memory usage: 808.7 KB
```

In [10]:
```python
df.describe().T
```

Out[10]:

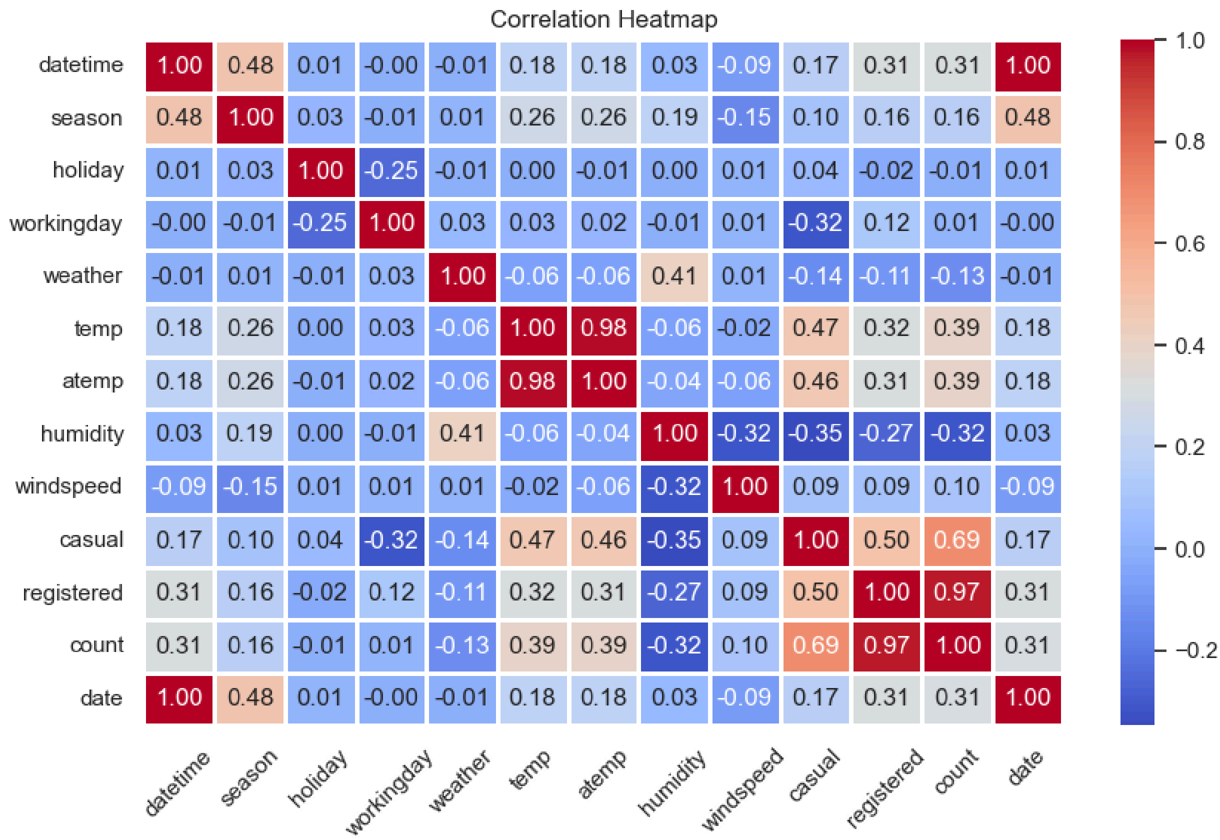
	count	mean	min	25%	50%	75%	max	std
datetime	10886	2011-12-27 05:56:22.399411968	2011-01-01 00:00:00	2011-07-02 07:15:00	2012-01-01 20:30:00	2012-07-01 12:45:00	2012-12-19 23:00:00	NaN
temp	10886.0	20.23086	0.82	13.94	20.5	26.24	41.0	7.79159
atemp	10886.0	23.655084	0.76	16.665	24.24	31.06	45.455	8.474601
humidity	10886.0	61.88646	0.0	47.0	62.0	77.0	100.0	19.245033
windspeed	10886.0	12.799395	0.0	7.0015	12.998	16.9979	56.9969	8.164537
casual	10886.0	36.021955	0.0	4.0	17.0	49.0	367.0	49.960477
registered	10886.0	155.552177	0.0	36.0	118.0	222.0	886.0	151.039033
count	10886.0	191.574132	1.0	42.0	145.0	284.0	977.0	181.144454
date	10886	2011-12-26 18:23:52.592320256	2011-01-01 00:00:00	2011-07-02 00:00:00	2012-01-01 00:00:00	2012-07-01 00:00:00	2012-12-19 00:00:00	NaN

🕵️ Statistical Information:

- **Count**: All columns have the same count, indicating no missing values in the dataset.
- **datetime column**: The data provided are within the dates 2011-01-01 to 2012-12-19
- **temp column**: The temparature ranges from 0.82 degree to 41.0 degree with the mean temparature of 20.23
- **humidity column**: The humidity ranges from 0 to 100 with the mean of 61.88.

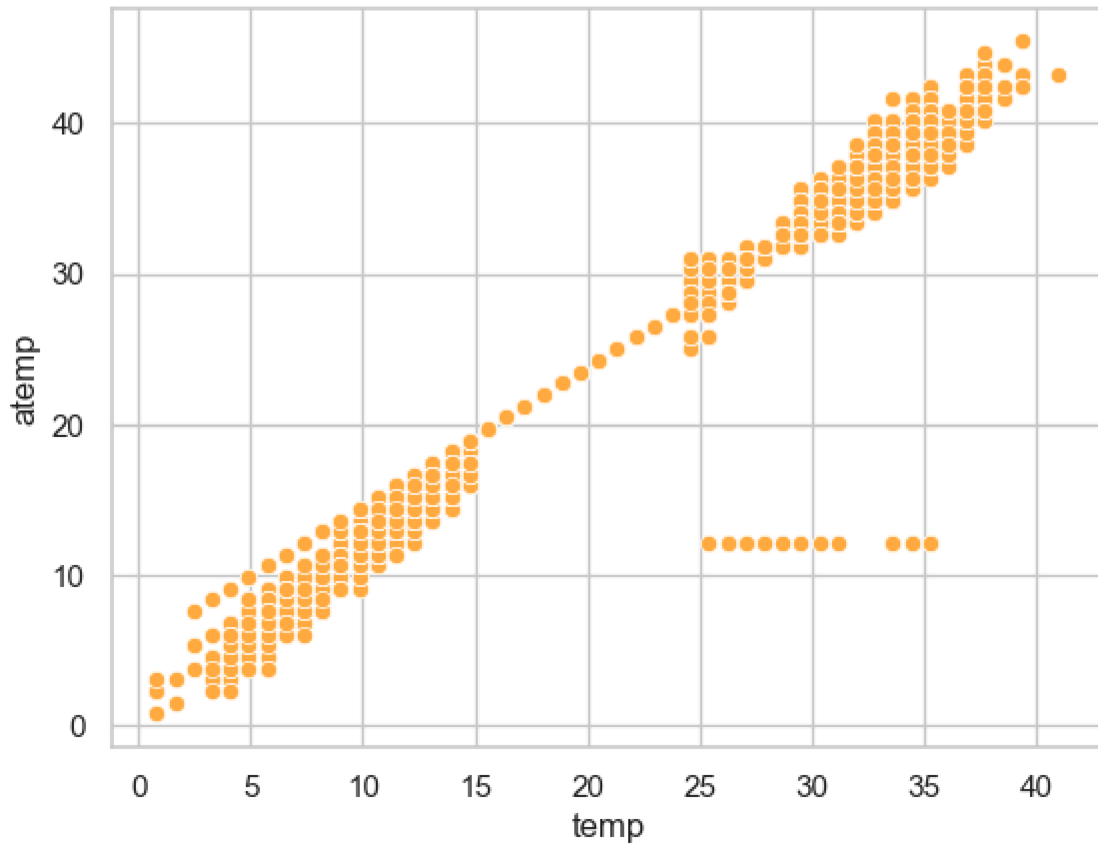
In [11]:
```python
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f', linewidths=2)
plt.title('Correlation Heatmap')
```

```
plt.xticks(rotation=45)
plt.show()
```



In [12]:
```
sns.scatterplot(x='temp', y='atemp', data=df, color='#ffae42')
```

Out[12]:
```
<Axes: xlabel='temp', ylabel='atemp'>
```



🕵🏾 Attribute Correlation:

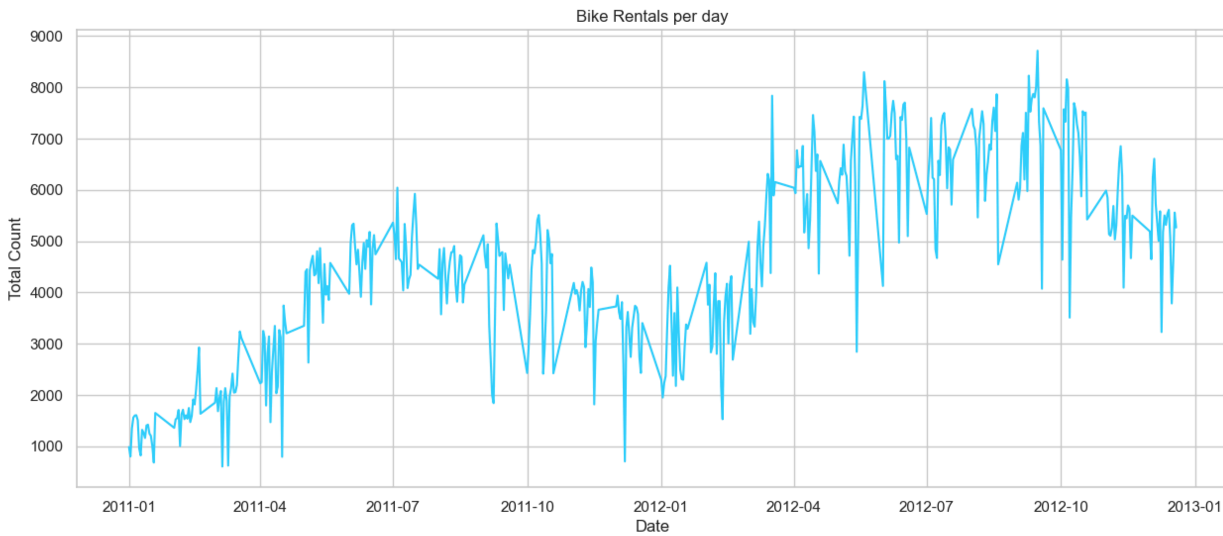
- **Strong positive correlations** are observed between 'count' and features like 'casual' (0.69), 'registered' (0.97), 'datetime' (0.31), 'date' (0.31), 'temp' (0.39), and 'atemp' (0.39), indicating that these features have a significant impact on the demand for shared electric cycles.
- Additionally, 'season' also shows a *moderate positive correlation* with 'count' (0.16), suggesting a seasonal influence on cycle demand.
- **Negative correlations** are observed between 'count' and features like 'workingday' (-0.32), 'weather' (-0.13), and 'humidity' (-0.32), indicating that these factors might suppress cycle demand.

In [13]:
```
## Dropping the 'atemp' column as it is highly correlated with 'temp'
df.drop('atemp', axis=1, inplace=True)

## Mapping the values of the columns to their respective categories
df['season'] = df['season'].map({1: 'Spring', 2: 'Summer', 3: 'Fall', 4: 'Winter'})
df['weather'] = df['weather'].map({1: 'Clear', 2: 'Mist', 3: 'Light Snow, Light Rain',
df['holiday'] = df['holiday'].map({0: 'No', 1: 'Yes'})
df['workingday'] = df['workingday'].map({0: 'No', 1: 'Yes'})
```

Analysis 👩🏽‍🔬

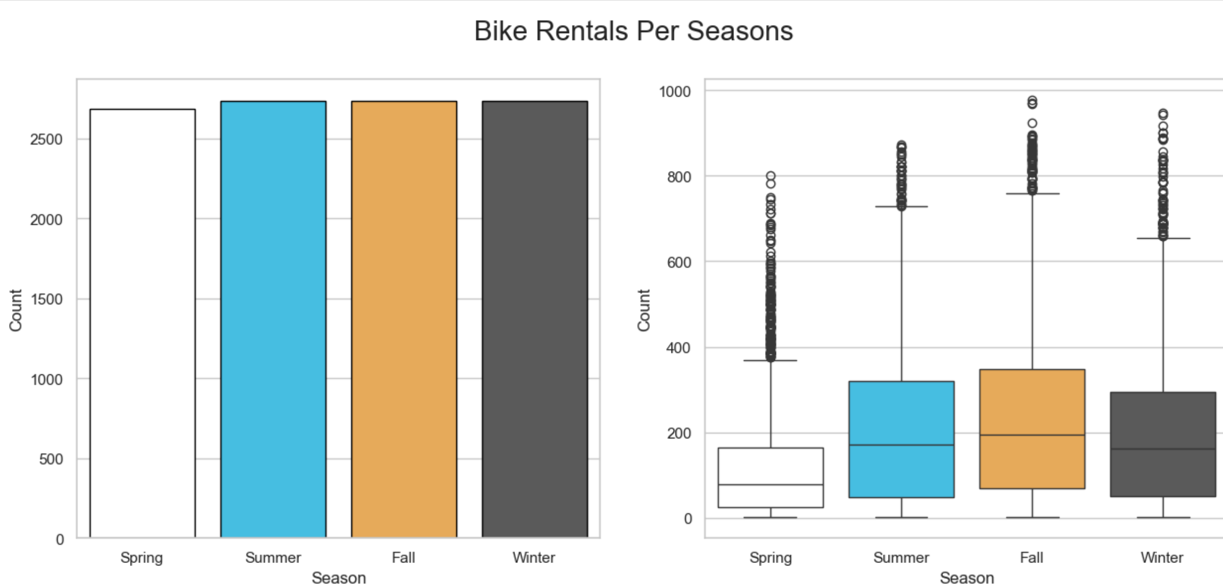
In [14]:
```
plt.figure(figsize=(15, 6))
# df.groupby('date')['count'].sum().plot()
sns.lineplot(x='date', y='count', data=df.groupby('date')['count'].sum().reset_index()
plt.xlabel('Date')
plt.ylabel('Total Count')
```

```python
plt.title('Bike Rentals per day')
plt.show()
```



In [15]:

```python
plt.figure(figsize=(15, 6))
plt.suptitle('Bike Rentals Per Seasons', fontsize=20)

plt.subplot(1, 2, 1)
sns.countplot(x='season', data=df, palette=four_set_palette, edgecolor='black')
# df['season'].value_counts().plot(kind='pie', color=bi_palette)
plt.xlabel('Season')
plt.ylabel('Count')

plt.subplot(1, 2, 2)
sns.boxplot(x='season', y='count', data=df, palette=four_set_palette)
plt.xlabel('Season')
plt.ylabel('Count')

plt.show()
```

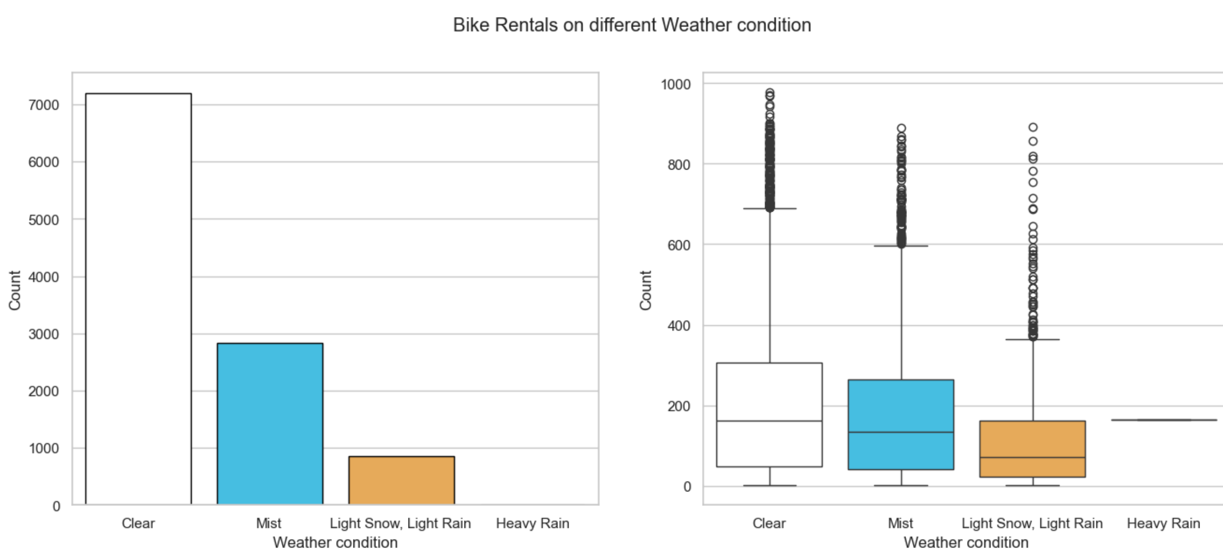


🕵️ Bike Rental Per Season:

- Given data has almost equal spread of data for each seasons

- The season with the most bikes rented is Fall. There were around 1000 bikes rented, with the median of ~200 rentals.
- The season with the least bike rented is Spring. There were around 800 bikes rented.
- There is wide spread of outliers in Spring in compare to any other season. This indicates the no. of bikes rented in spring season can vary more than in other seasons.

In [16]:
```
plt.figure(figsize=(16, 6))
plt.suptitle('Bike Rentals on different Weather condition')

plt.subplot(1, 2, 1)
sns.countplot(x='weather', data=df, palette=four_set_palette, edgecolor='black')
# df['season'].value_counts().plot(kind='pie', color=bi_palette)
plt.xlabel('Weather condition')
plt.ylabel('Count')

plt.subplot(1, 2, 2)
sns.boxplot(x='weather', y='count', data=df, palette=four_set_palette)
plt.xlabel('Weather condition')
plt.ylabel('Count')
plt.show()
```

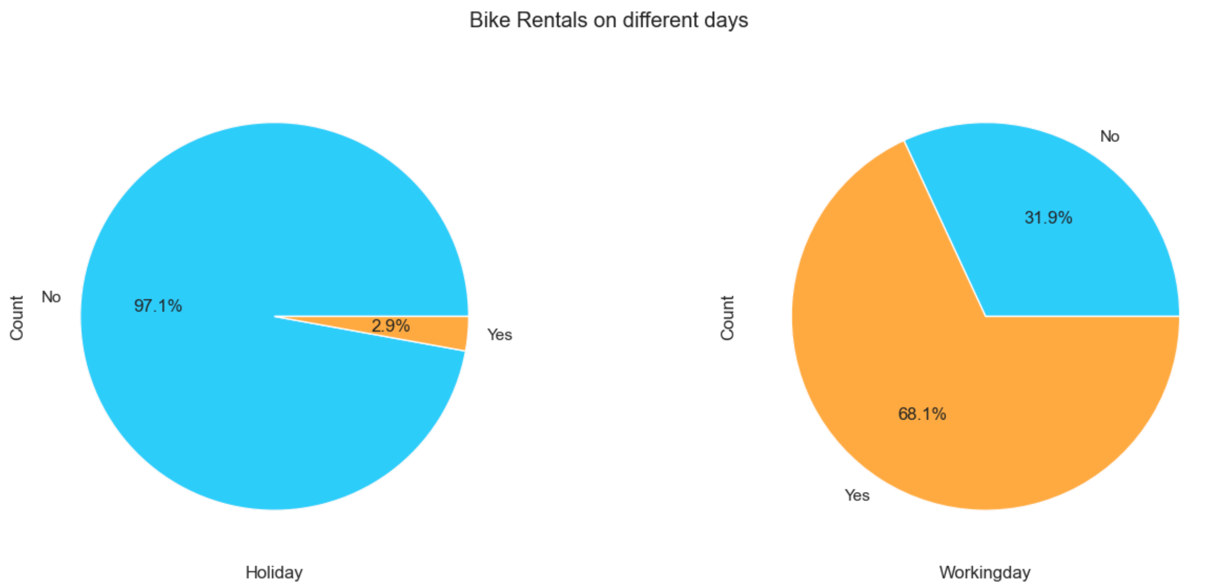


🕵🏽 Bike Rental on different weather conditions:

- Given data has almost ~66% in clear weather condition.
- Notably only one rental data under Heavy rain condition.
- The IQR of clear box range from 50 to 300 with the median of 180.
- The IQR of Mist box range from 50 to 250 with the median of 120.
- There is wide spread of outliers in Light rain/show in compare to any other weather condition. This indicates the no. of bikes rented in Light rain/show can vary more than in other weather condition.

In [17]:
```
plt.figure(figsize=(16, 6))
plt.suptitle('Bike Rentals on different days')

plt.subplot(1, 2, 1)
df_ = df['holiday'].value_counts().sort_index()
```

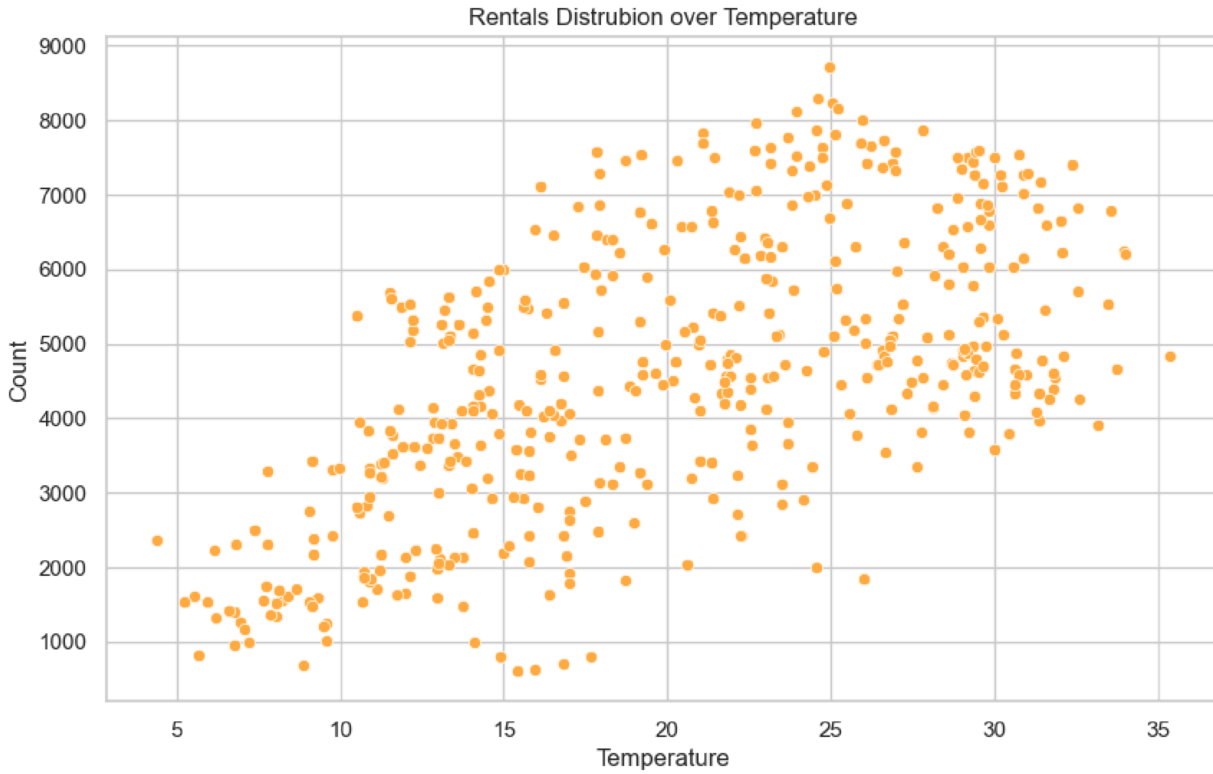
```python
plt.pie(df_, labels=df_.index, colors=bi_palette, autopct='%1.1f%%')
plt.xlabel('Holiday')
plt.ylabel('Count')

plt.subplot(1, 2, 2)
df_ = df['workingday'].value_counts().sort_index()
plt.pie(df_, labels=df_.index, colors=bi_palette, autopct='%1.1f%%')
plt.xlabel('Workingday')
plt.ylabel('Count')
plt.show()
```



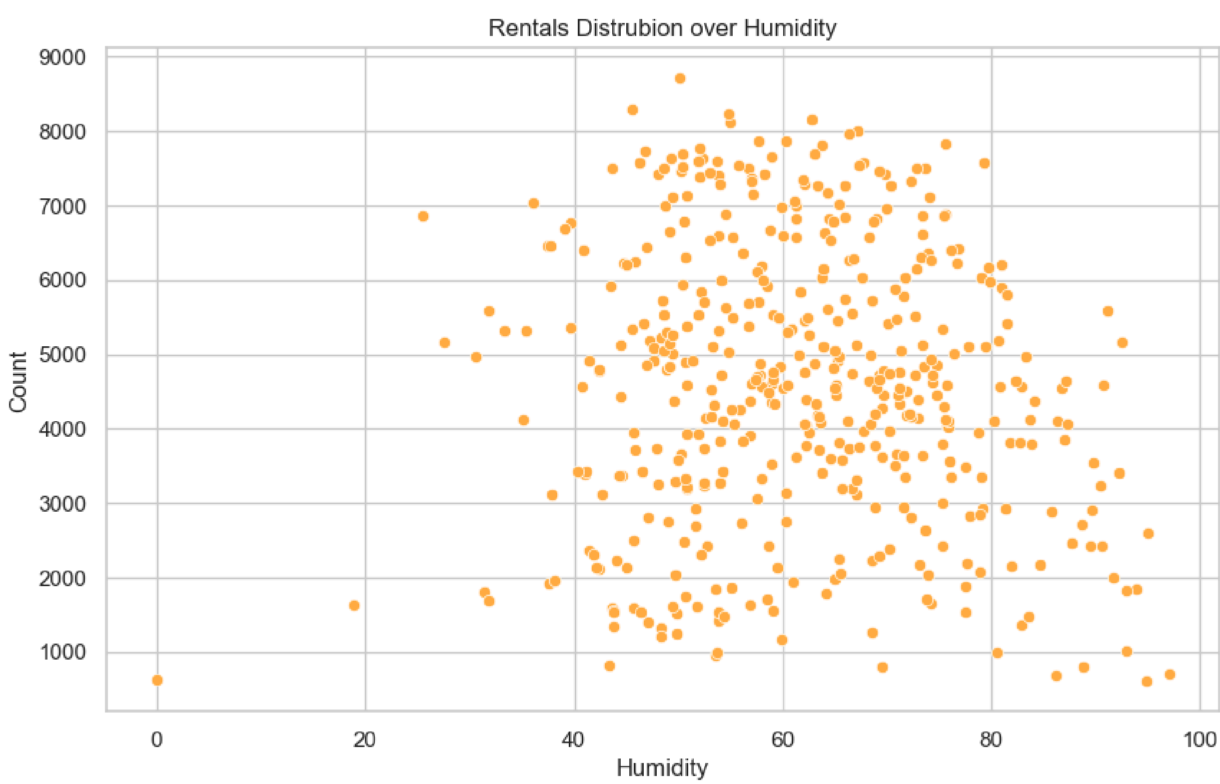
🕵️ Bike Rental on different Days:

- Given data has ~3% of rental information on the Holiday.
- ~68% of the rental data comes from the working day. And 30% from the Non-working day

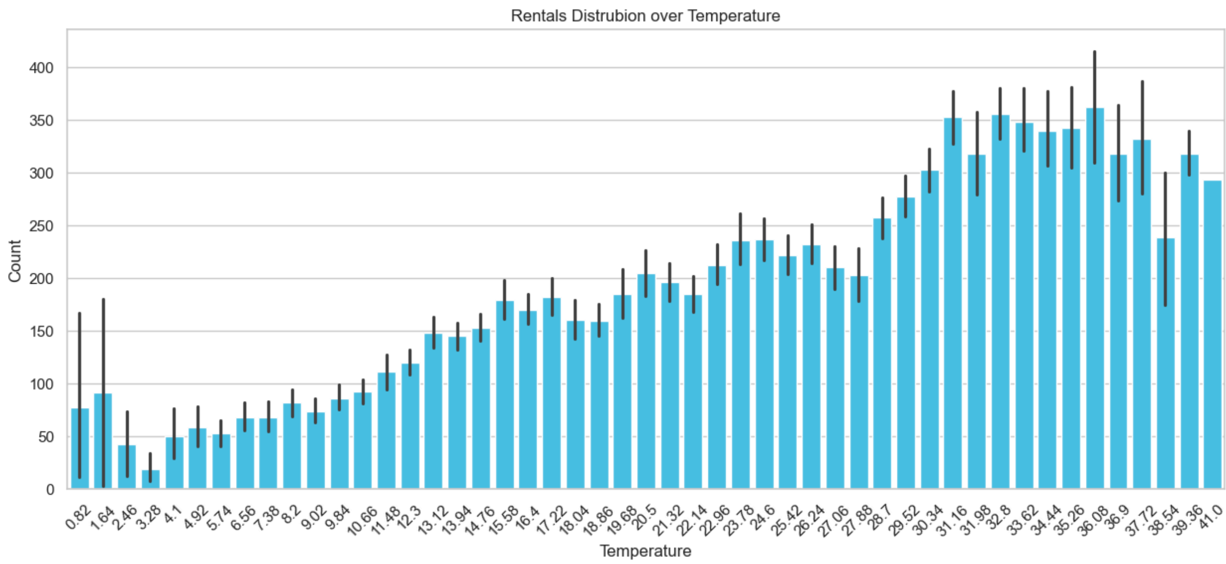
In [18]:
```python
plt.figure(figsize=(10, 6))
df_ = df.groupby('date').agg({'temp':'mean','count': 'sum'}).reset_index()
sns.scatterplot(x='temp', y='count', data=df_, color='#ffae42')
plt.xlabel('Temperature')
plt.ylabel('Count')
plt.title('Rentals Distrubion over Temperature')
plt.show()
```



In [19]:
```
plt.figure(figsize=(10, 6))
df_ = df.groupby('date').agg({'humidity':'mean','count': 'sum'}).reset_index()
sns.scatterplot(x='humidity', y='count', data=df_, color='#ffae42')
plt.xlabel('Humidity')
plt.ylabel('Count')
plt.title('Rentals Distrubion over Humidity')
plt.show()
```



In [20]:
```
plt.figure(figsize=(15, 6))
sns.barplot(x='temp', y='count', data=df, color='#2fcdfd')
plt.xlabel('Temperature')
plt.xticks(rotation=45)
plt.ylabel('Count')
plt.title('Rentals Distrubion over Temperature')
plt.show()
```



🕵🏾 Impact on Temperature:

- It shows that people rent bikes during the temperature ranges from 30-37 degree celsius.
- And comparetively lesser in the low temperature less than 10 degree.

Hypothesis Testing 🧪

Does Working Day has an effect on the number of electric cycles rented

In [21]:
```
# Hypothesis formulation

# Null Hypothesis:
# There is no significant difference in the number of electric cycles rented on workin

# Alternative Hypothesis:
# There is a significant difference in the number of electric cycles rented on working


working_day = df[df['workingday'] == 'Yes']['count']
non_working_day = df[df['workingday'] == 'No']['count']

# Performing the t-test
t_stat, p_val = stats.ttest_ind(working_day, non_working_day)
print("T-Statistic: ", t_stat)
print("P-Value: ", p_val)

# set the significance level
alpha = 0.05
```

```python
# decision making
if p_val < alpha:
    print("(Reject Null Hypothesis): There is a significant difference in the number c
    rented on working days and non-working days.")
else:
    print("(Failed to Reject Null Hypothesis): There is no significant difference in t
    rented on working days and non-working days.")
```

```
T-Statistic:  1.2096277376026694
P-Value:  0.22644804226361348
(Failed to Reject Null Hypothesis): There is no significant difference in the number
of electric cycles rented on working days and non-working days.
```

👦🔬 Based on the above T-test:

There is no significant difference in the number of electric cycles rented on working days and non-working days.

No. of cycles rented is similar or different in different weather

In [22]:

```python
# Hypothesis formulation

# Null Hypothesis:
# There is no significant difference in the number of electric cycles rented in differ

# Alternative Hypothesis:
# There is a significant difference in the number of electric cycles rented in differe

weather_1 = df[df['weather'] == 'Clear']['count']
weather_2 = df[df['weather'] == 'Mist']['count']
weather_3 = df[df['weather'] == 'Light Snow, Light Rain']['count']
weather_4 = df[df['weather'] == 'Heavy Rain']['count']

# Performing the ANOVA test
f_stat, p_val = stats.f_oneway(weather_1, weather_2, weather_3, weather_4)
print("F-Statistic: ", f_stat)
print("P-Value: ", p_val)

# set the significance level
alpha = 0.05

# decision making
if p_val < alpha:
    print("(Reject Null Hypothesis): There is a significant difference in the number c
    rented in different weather conditions.")
else:
    print("(Failed to Reject Null Hypothesis): There is no significant difference in t
    rented in different weather conditions.")
```

```
F-Statistic:  65.53024112793271
P-Value:  5.482069475935669e-42
(Reject Null Hypothesis): There is a significant difference in the number of electric
cycles rented in different weather conditions.
```

👦🔬 Based on the above ANOVA-test:

There is a significant difference in the number of electric cycles rented in different weather conditions.

No. of cycles rented is similar or different in different season

In [23]:
```
# Hypothesis formulation
# Null Hypothesis: There is no significant difference in the number of electric cycles
# Alternative Hypothesis: There is a significant difference in the number of electric

spring = df[df['season'] == 'Spring']['count']
summer = df[df['season'] == 'Summer']['count']
fall = df[df['season'] == 'Fall']['count']
winter = df[df['season'] == 'Winter']['count']

# performing the ANOVA test
f_stat, p_val = stats.f_oneway(spring, summer, fall, winter)
print("F-Statistic: ", f_stat)
print("P-Value: ", p_val)

# set the significance level
alpha = 0.05

# decision making
if p_val < alpha:
    print("(Reject Null Hypothesis): There is a significant difference in the number c
    rented in different seasons.")
else:
    print("(Failed to Reject Null Hypothesis): There is no significant difference in t
    rented in different seasons.")
```

```
F-Statistic:  236.94671081032106
P-Value:  6.164843386499654e-149
(Reject Null Hypothesis): There is a significant difference in the number of electric
cycles rented in different seasons.
```

👦🔬 Based on the above ANOVA-test:

There is a significant difference in the number of electric cycles rented in different seasons.

Is Weather dependent on the season

In [24]:
```
# Hypothesis formulation
# Null Hypothesis: Weather is independent of the season
# Alternative Hypothesis: Weather is dependent on the season

contingency_table = pd.crosstab(df['season'], df['weather'])
print("Contingency Table: \n", contingency_table, "\n")

# Performing the Chi-Square test
chi2, p, dof, expected = stats.chi2_contingency(contingency_table)
print("Chi2: ", chi2)
print("P-Value: ", p)

# set the significance level
alpha = 0.05

# decision making
if p < alpha:
    print("(Reject Null Hypothesis): Weather is dependent on the season.")
```

```python
else:
    print("(Failed to Reject Null Hypothesis): Weather is independent of the season.")
```

```
Contingency Table:
 weather  Clear  Mist  Light Snow, Light Rain  Heavy Rain
season
Spring    1759   715                      211           1
Summer    1801   708                      224           0
Fall      1930   604                      199           0
Winter    1702   807                      225           0

Chi2:  49.158655596893624
P-Value:  1.549925073686492e-07
(Reject Null Hypothesis): Weather is dependent on the season.
```

🧑‍🔬 Based on the above Chi-Square test:

Weather is dependent on the season.

```
In [ ]:
```