In [21]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Dataset

In [22]:

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving day.csv to day (1).csv
```

In [23]:

```
# Load the 'day.csv' file
df = pd.read_csv('day.csv')

# Preview the data
df.head()
```

Out[23]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathe
0	1	2011-01-01	1	0	1	0	6	0	
1	2	2011-01-02	1	0	1	0	0	0	
2	3	2011-01-03	1	0	1	0	1	1	
3	4	2011-01-04	1	0	1	0	2	1	
4	5	2011-01-05	1	0	1	0	3	1	

Data Cleaning and Preparation

In [24]:

```
df['dteday'] = pd.to_datetime(df['dteday'])
df.rename(columns={
    'dteday': 'date',
    'yr': 'year',
    'mnth': 'month',
    'weekday': 'weekday_num',
    'weathersit': 'weather',
    'hum': 'humidity',
    'cnt': 'total_rentals'
}, inplace=True)

df['weekday'] = df['date'].dt.day_name()
df['month_name'] = df['date'].dt.month_name()
```

```
df['season_name'] = df['season'].map({1: 'Spring', 2: 'Summer', 3: 'Fall', 4
df['weather_label'] = df['weather'].map({
    1: 'Clear/Cloudy',
    2: 'Mist/Cloudy',
    3: 'Light Rain/Snow',
    4: 'Heavy Rain/Snow'
})
```

In [30]:

```
# Create binned columns
df['humidity_bin'] = pd.cut(df['humidity'], bins=[0, 0.3, 0.5, 0.7, 0.9, 1.0
df['windspeed_bin'] = pd.cut(df['windspeed'], bins=[0, 0.1, 0.2, 0.3, 0.4, 0

# Create the plots
plots = []
```

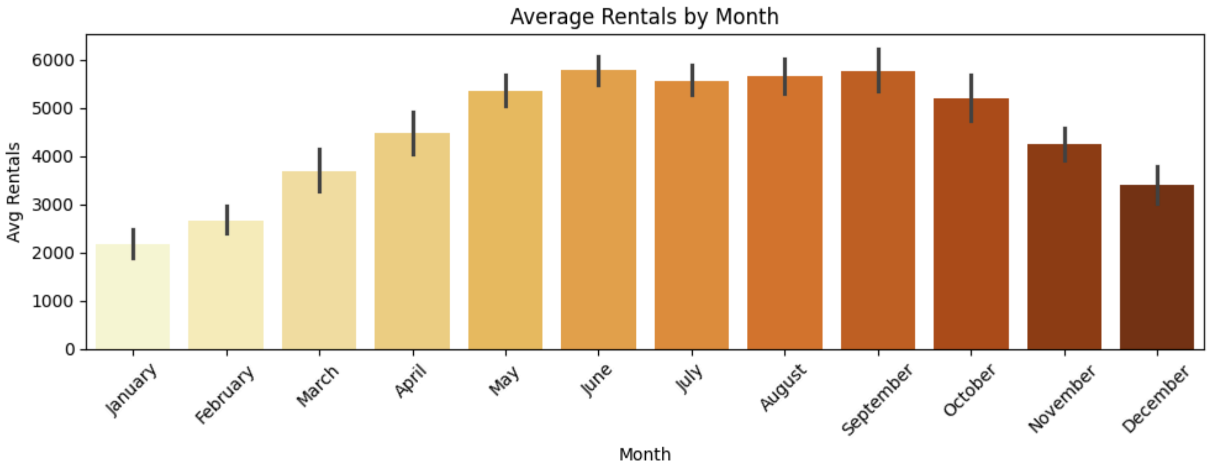
Exploratory Data Analysis

A. Rentals by Month

In [32]:

```
# Rentals by month
plt.figure(figsize=(10, 4))
month_order = ['January', 'February', 'March', 'April', 'May', 'June',
               'July', 'August', 'September', 'October', 'November', 'Decemb
df['month_name'] = pd.Categorical(df['month_name'], categories=month_order,
sns.barplot(x='month_name', y='total_rentals', data=df, palette='YlOrBr')
plt.title('Average Rentals by Month')
plt.xlabel('Month')
plt.ylabel('Avg Rentals')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
plt.close()
```

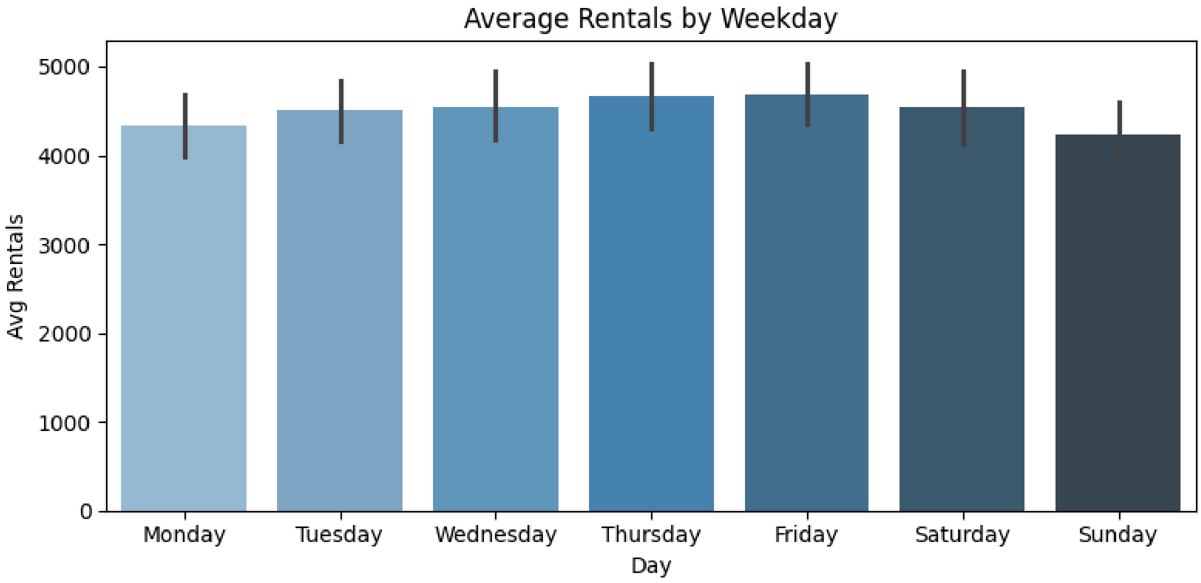
```
<ipython-input-32-be0cfd1ea91e>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='month_name', y='total_rentals', data=df, palette='YlOrBr')
```



B. Rentals by Weekday

In [33]:
```
# Average rentals by weekday
plt.figure(figsize=(8, 4))
order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday',
sns.barplot(x='weekday', y='total_rentals', data=df, order=order, palette='B
plt.title('Average Rentals by Weekday')
plt.xlabel('Day')
plt.ylabel('Avg Rentals')
plt.tight_layout()
plt.show()
plt.close()
```

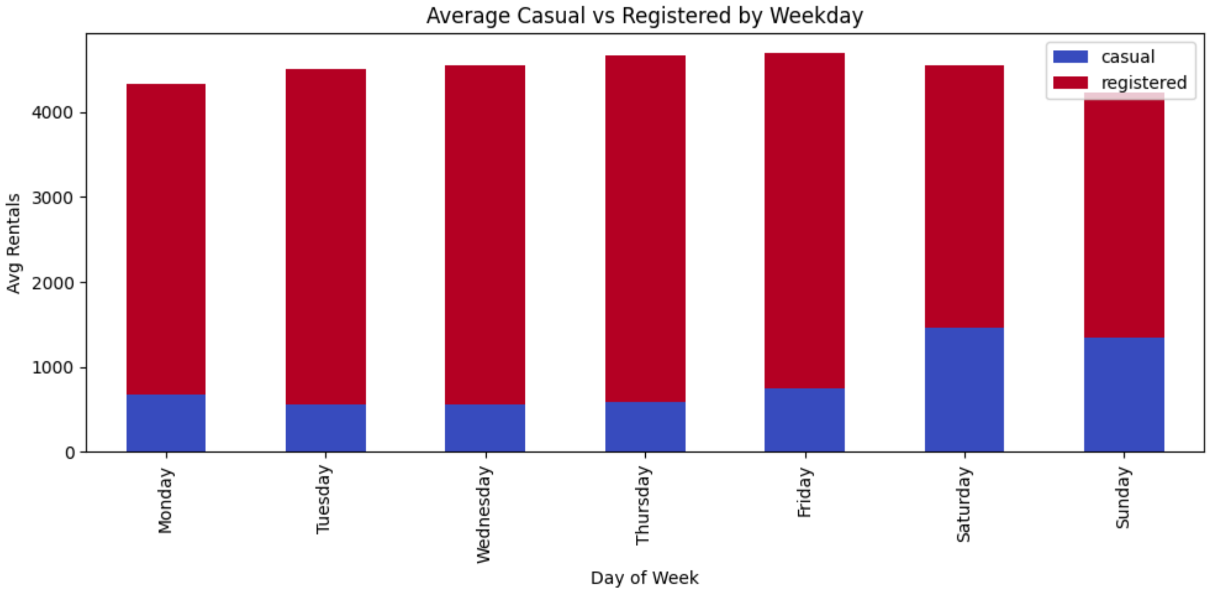
```
<ipython-input-33-418d6c60c1be>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='weekday', y='total_rentals', data=df, order=order, palette
='Blues_d')
```



C. Registered vs Casual Users by Weekday

In [35]:

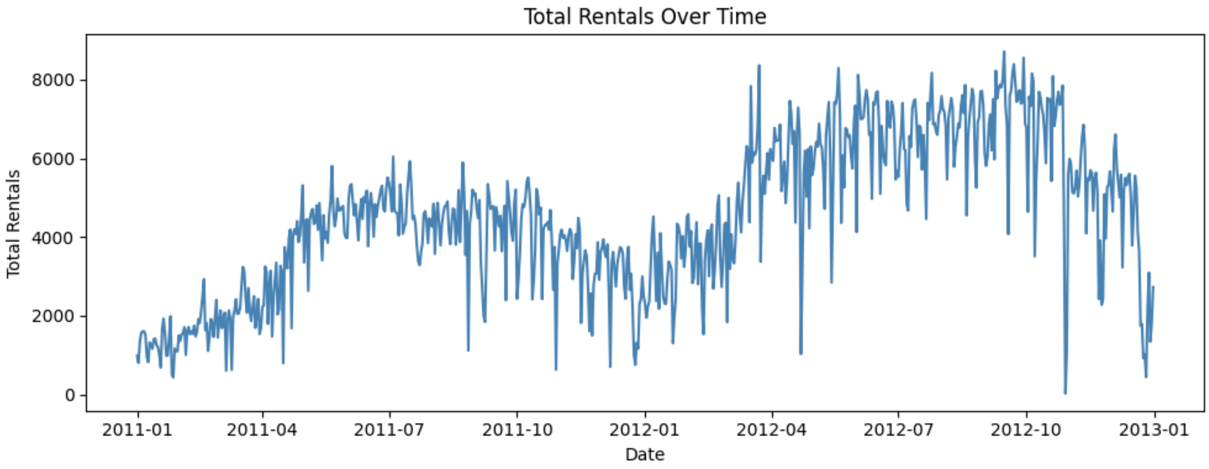
```
# Registered vs Casual by weekday
grouped = df.groupby('weekday')[['casual', 'registered']].mean().reindex(orc
grouped.plot(kind='bar', stacked=True, figsize=(10, 5), colormap='coolwarm')
plt.title('Average Casual vs Registered by Weekday')
plt.xlabel('Day of Week')
plt.ylabel('Avg Rentals')
plt.tight_layout()
plt.show()
plt.close()
```



D. Rentals Over Time

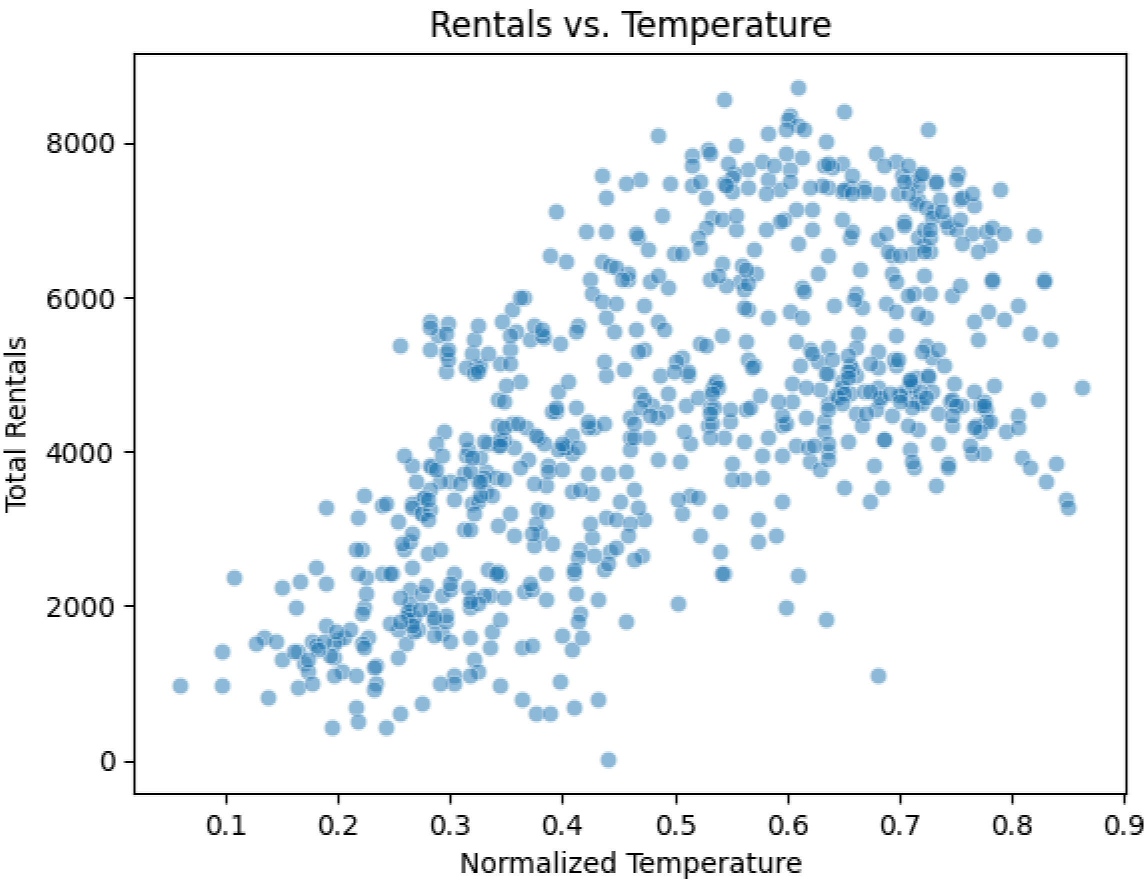
In [34]:

```
# Rentals over time
plt.figure(figsize=(10, 4))
plt.plot(df['date'], df['total_rentals'], color='steelblue')
plt.title('Total Rentals Over Time')
plt.xlabel('Date')
plt.ylabel('Total Rentals')
plt.tight_layout()
plt.show()
plt.close()
```



E. Rentals vs. Temperature

```
In [29]: #Rentals vs Temperature
sns.scatterplot(data=df, x='temp', y='total_rentals', alpha=0.5)
plt.title('Rentals vs. Temperature')
plt.xlabel('Normalized Temperature')
plt.ylabel('Total Rentals')
plt.show()
```

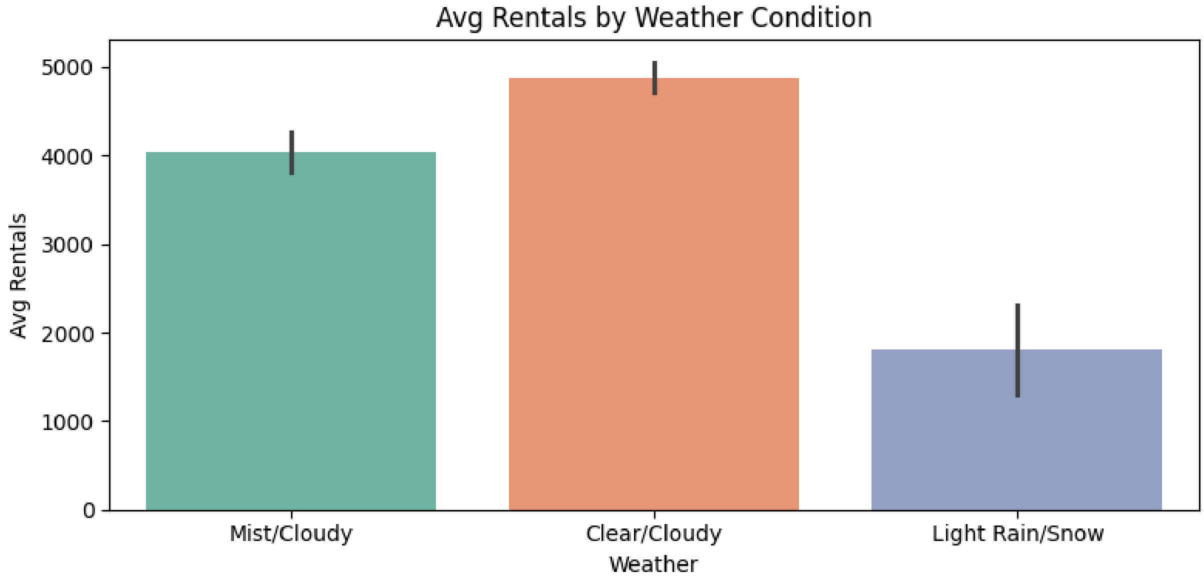


F. Average Rentals by Weather Conditions

In [36]:

```
#Rentals by Weather Condition
plt.figure(figsize=(8, 4))
sns.barplot(x='weather_label', y='total_rentals', data=df, palette='Set2')
plt.title('Avg Rentals by Weather Condition')
plt.xlabel('Weather')
plt.ylabel('Avg Rentals')
plt.tight_layout()
plt.show()
plt.close()
```

```
<ipython-input-36-f1be69b6edd1>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='weather_label', y='total_rentals', data=df, palette='Set2')
```

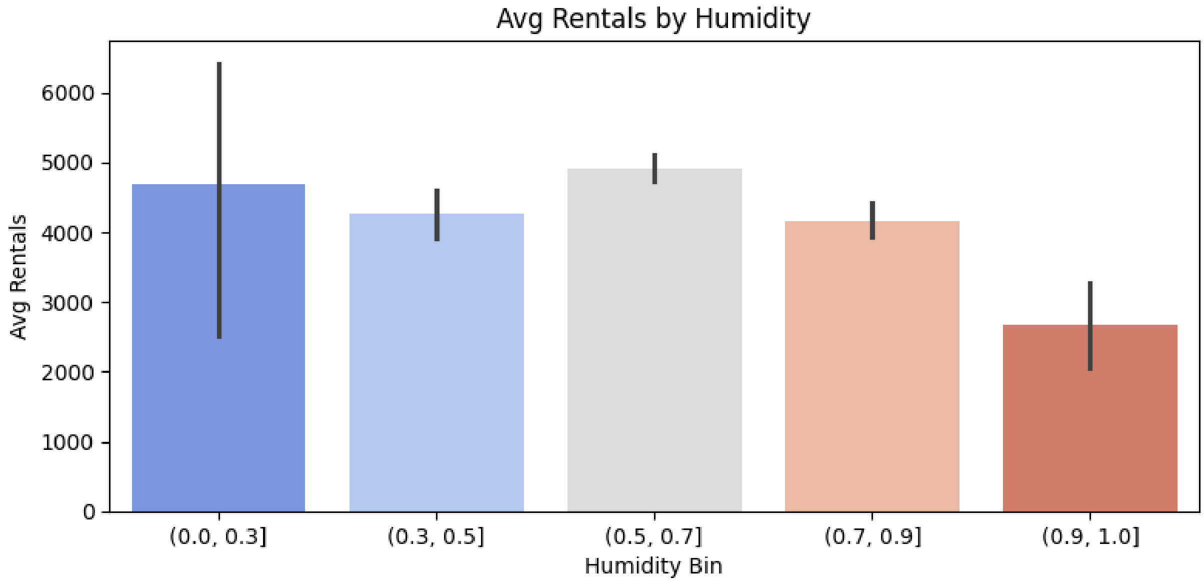


G. Average Rentals by Humidity

In [37]:

```
# Rentals by humidity bin
plt.figure(figsize=(8, 4))
sns.barplot(x='humidity_bin', y='total_rentals', data=df, palette='coolwarm'
plt.title('Avg Rentals by Humidity')
plt.xlabel('Humidity Bin')
plt.ylabel('Avg Rentals')
plt.tight_layout()
plt.show()
plt.close()
```

```
<ipython-input-37-79d569e89bc9>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='humidity_bin', y='total_rentals', data=df, palette='coolwar
m')
```

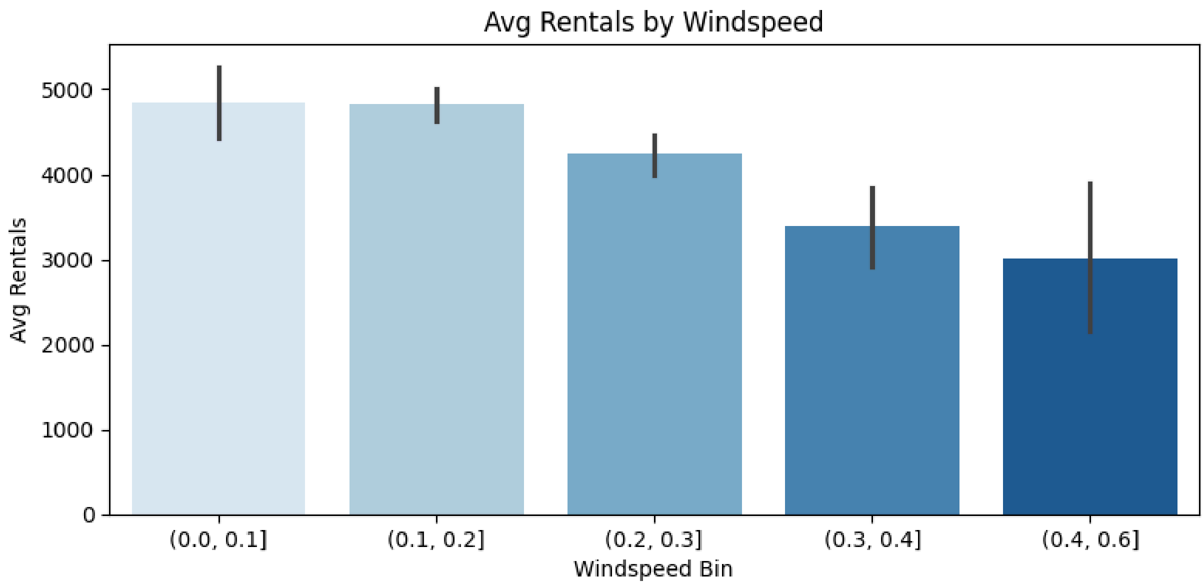


H. Average Rentals by Windspeed

In [38]:

```
# Rentals by windspeed bin
plt.figure(figsize=(8, 4))
sns.barplot(x='windspeed_bin', y='total_rentals', data=df, palette='Blues')
plt.title('Avg Rentals by Windspeed')
plt.xlabel('Windspeed Bin')
plt.ylabel('Avg Rentals')
plt.tight_layout()
plt.show()
plt.close()
```

```
<ipython-input-38-d8e215ae9989>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(x='windspeed_bin', y='total_rentals', data=df, palette='Blue
s')
```



Key Insights from Bike Sharing Demand Dat

Time-Based Trends

-Peak bike rentals occur on weekdays, especially Monday to Friday, indicating commute behavior.

-Weekend usage spikes for casual users, likely due to recreational activities.

-Rentals increase from Spring to Fall, peaking in September, then drop sharply in Winter.

Weather Influence

-Higher temperatures lead to more rentals, showing a positive correlation with comfort-level weather.

-Rentals decrease on humid, windy, or rainy days, especially during bad weather conditions (weathersit = 3 or 4).

User Behavior Patterns

-Registered users dominate weekday usage.

-Casual users prefer weekends and holidays, showing potential for tourism-focused offers.

Operational Recommendations

-Increase bike availability during rush hours and summer/fall months.

-Use weather-triggered promotions to increase off-peak or off-season demand.

-Create user-type-based promotions (e.g., loyalty discounts for registered users, weekend bundles for casual users).