Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

Academic Year: 2020-2021

Subject: Data Mining and Business Intelligence

Evaluation of Bike Sharing in Washington D.C.

A mini-project submitted for

Business Intelligence Lab (Sem VI)

By

SAP ID	Name
60003180050	Shwet Shah
60003198002	Soham Dave

Problem Statement

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back to another position. Currently, there are about over 500 bike-sharing programs around the world which are composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system in Washington, DC with the corresponding weather and seasonal information.

Dataset link: *https://www.kaggle.com/marklvl/bike-sharing-dataset*

In [ ]:

```
import numpy as np
import pandas as pd
```

In [ ]:

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [ ]:

```
day_df=pd.read_csv('day.csv')
hour_df=pd.read_csv('hour.csv')
```

In [ ]:

```
day_df.head()
```

Out[ ]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	ca
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	

In [ ]:

```
hour_df.head()
```

Out[ ]:

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8

2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	1	
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	

In [ ]:

```
day_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   instant     731 non-null    int64  
 1   dteday      731 non-null    object 
 2   season      731 non-null    int64  
 3   yr          731 non-null    int64  
 4   mnth        731 non-null    int64  
 5   holiday     731 non-null    int64  
 6   weekday     731 non-null    int64  
 7   workingday  731 non-null    int64  
 8   weathersit  731 non-null    int64  
 9   temp        731 non-null    float64
 10  atemp       731 non-null    float64
 11  hum         731 non-null    float64
 12  windspeed   731 non-null    float64
 13  casual      731 non-null    int64  
 14  registered  731 non-null    int64  
 15  cnt         731 non-null    int64  
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

In [ ]:

```
hour_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17379 entries, 0 to 17378
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   instant     17379 non-null  int64  
 1   dteday      17379 non-null  object 
 2   season      17379 non-null  int64  
 3   yr          17379 non-null  int64  
 4   mnth        17379 non-null  int64  
 5   hr          17379 non-null  int64  
 6   holiday     17379 non-null  int64  
 7   weekday     17379 non-null  int64  
 8   workingday  17379 non-null  int64  
 9   weathersit  17379 non-null  int64  
 10  temp        17379 non-null  float64
 11  atemp       17379 non-null  float64
 12  hum         17379 non-null  float64
 13  windspeed   17379 non-null  float64
 14  casual      17379 non-null  int64  
 15  registered  17379 non-null  int64  
 16  cnt         17379 non-null  int64  
dtypes: float64(4), int64(12), object(1)
memory usage: 2.3+ MB
```

In [ ]:

```
day_df.drop('instant',axis=1,inplace=True)
```

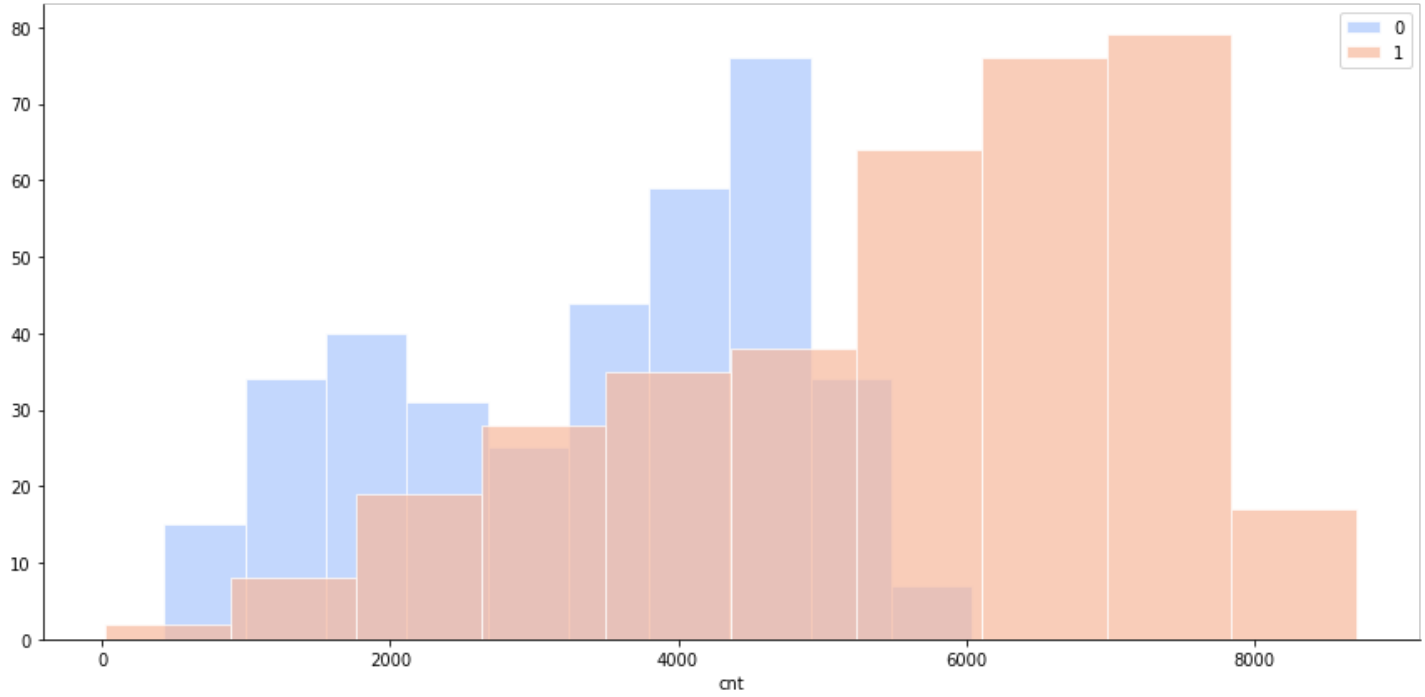
In [ ]:

```
g=sns.FacetGrid(day_df, hue='yr', palette='coolwarm',size=6,aspect=2)
g=g.map(plt.hist,'cnt',alpha=0.7, edgecolor='w')
plt.legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` p
arameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7f1334f90fd0>
```



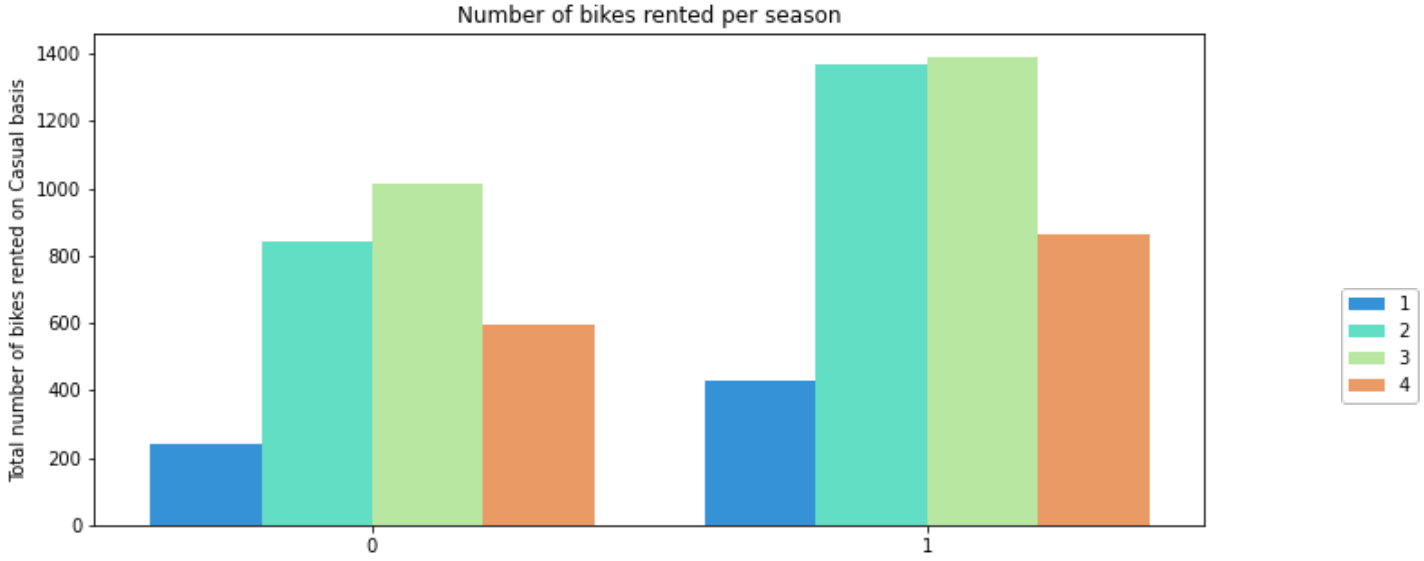
In [ ]:

```
plt.figure(figsize=(11,5))
sns.barplot('yr','casual',hue='season', data=day_df,palette='rainbow', ci=None)
plt.legend(loc='upper right',bbox_to_anchor=(1.2,0.5))
plt.xlabel('Year')
plt.ylabel('Total number of bikes rented on Casual basis')
plt.title('Number of bikes rented per season')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
Text(0.5, 1.0, 'Number of bikes rented per season')
```



Year

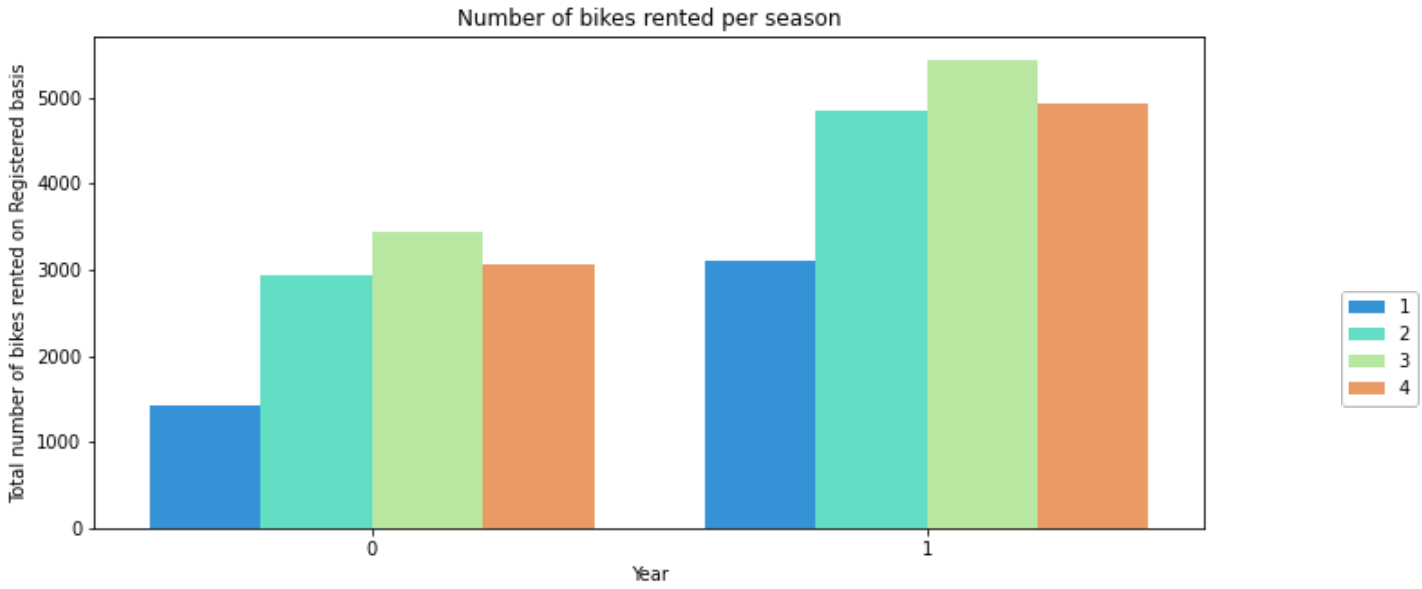
In [ ]:

```
plt.figure(figsize=(11,5))
sns.barplot('yr','registered',hue='season', data=day_df,palette='rainbow', ci=None)
plt.legend(loc='upper right',bbox_to_anchor=(1.2,0.5))
plt.xlabel('Year')
plt.ylabel('Total number of bikes rented on Registered basis')
plt.title('Number of bikes rented per season')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
Text(0.5, 1.0, 'Number of bikes rented per season')
```



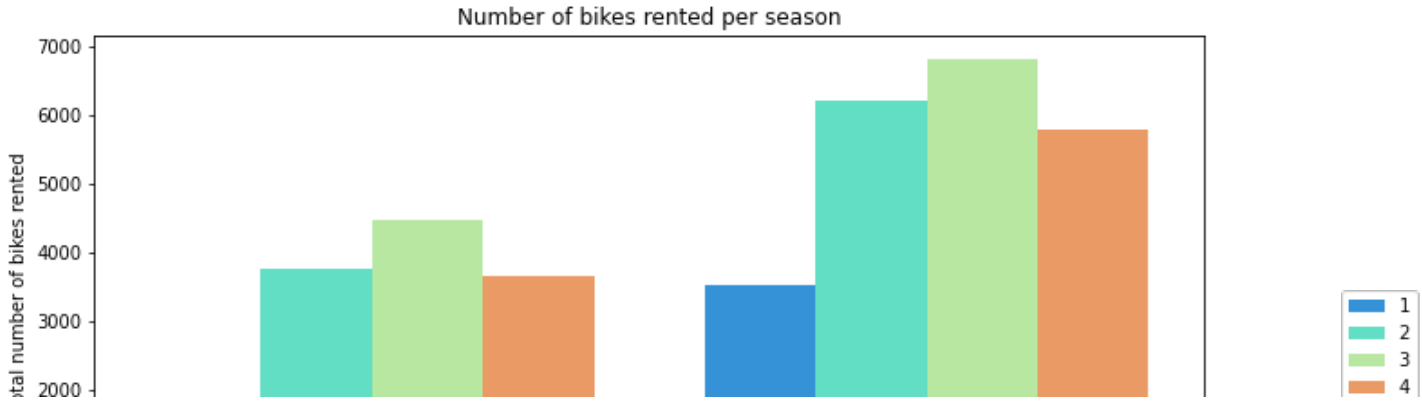
In [ ]:

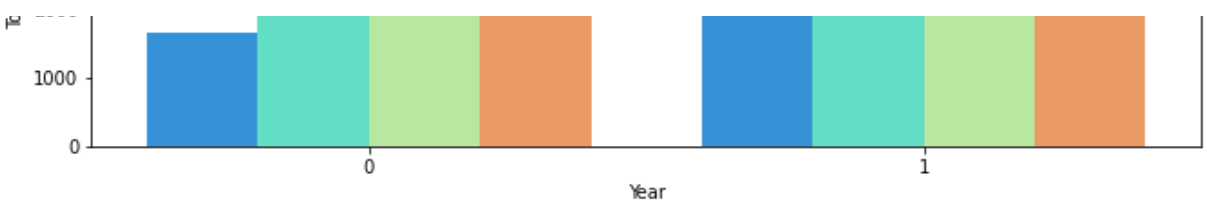
```
plt.figure(figsize=(11,5))
sns.barplot('yr','cnt',hue='season', data=day_df,palette='rainbow', ci=None)
plt.legend(loc='upper right',bbox_to_anchor=(1.2,0.5))
plt.xlabel('Year')
plt.ylabel('Total number of bikes rented')
plt.title('Number of bikes rented per season')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
Text(0.5, 1.0, 'Number of bikes rented per season')
```





In [ ]:

```
df_season_fall=day_df[day_df['season']==3]
```

In [ ]:

```
df_season_fall.mnth.nunique()
```

Out[ ]:

```
4
```

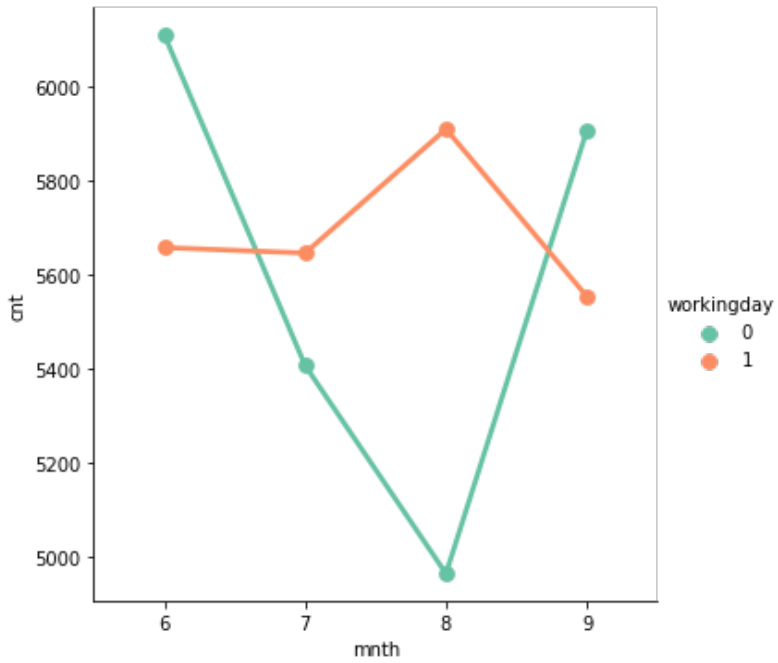
In [ ]:

```
sns.factorplot('mnth','cnt',hue='workingday',data=df_season_fall, ci=None, palette='Set2
')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3714: UserWarning: The `fac
torplot` function has been renamed to `catplot`. The original name will be removed in a f
uture release. Please update your code. Note that the default `kind` in `factorplot` (``p
oint``) has changed ``strip`` in `catplot`.
  warnings.warn(msg)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x7f1333697190>
```



In [ ]:

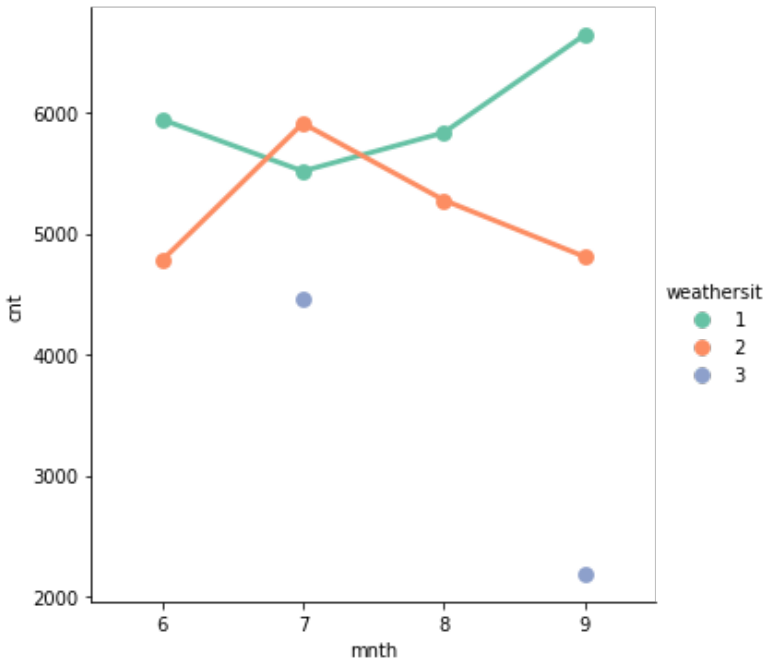
```
sns.factorplot('mnth','cnt',hue='weathersit',data=df_season_fall, ci=None, palette='Set2
')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:3714: UserWarning: The `fac
torplot` function has been renamed to `catplot`. The original name will be removed in a f
uture release. Please update your code. Note that the default `kind` in `factorplot` (``p
oint``) has changed ``strip`` in `catplot`.
  warnings.warn(msg)
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
```

```
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x7f1333cb91d0>
```



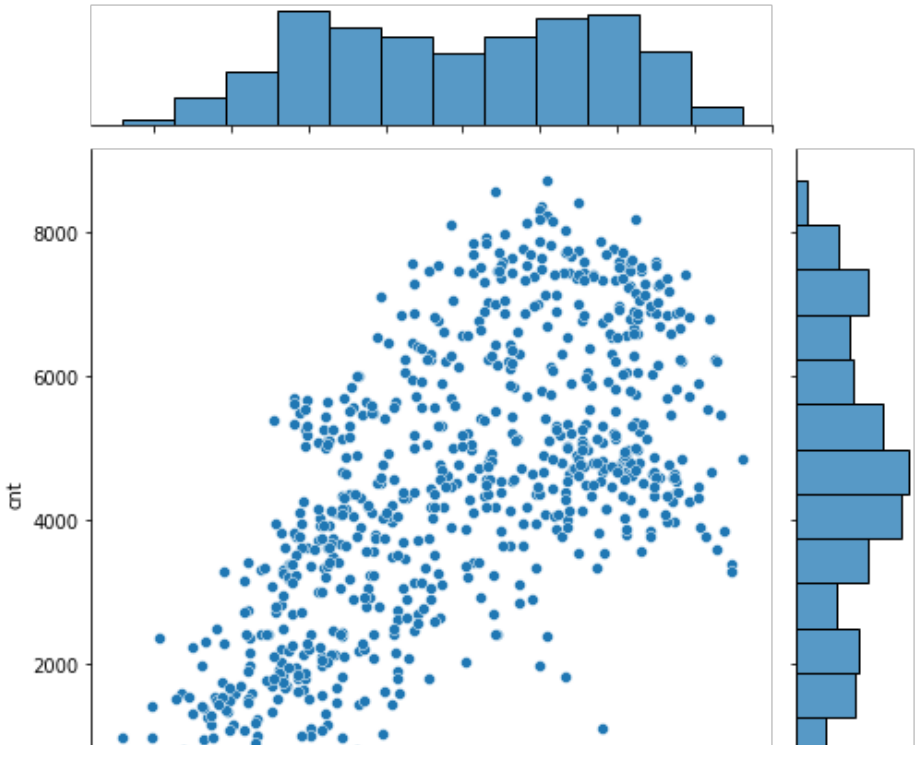
In [ ]:

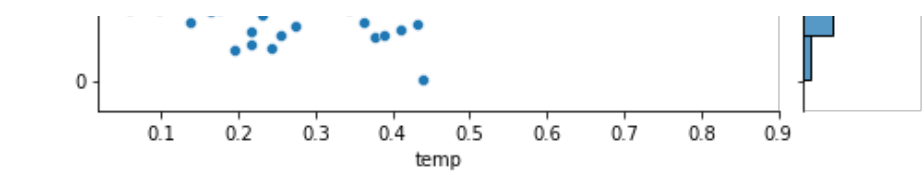
```
sns.jointplot('temp','cnt',data=day_df,size=7)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variables as keyword args: x, y. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will resu
lt in an error or misinterpretation.
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:2073: UserWarning: The `size`
parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

Out[ ]:

```
<seaborn.axisgrid.JointGrid at 0x7f1333557410>
```





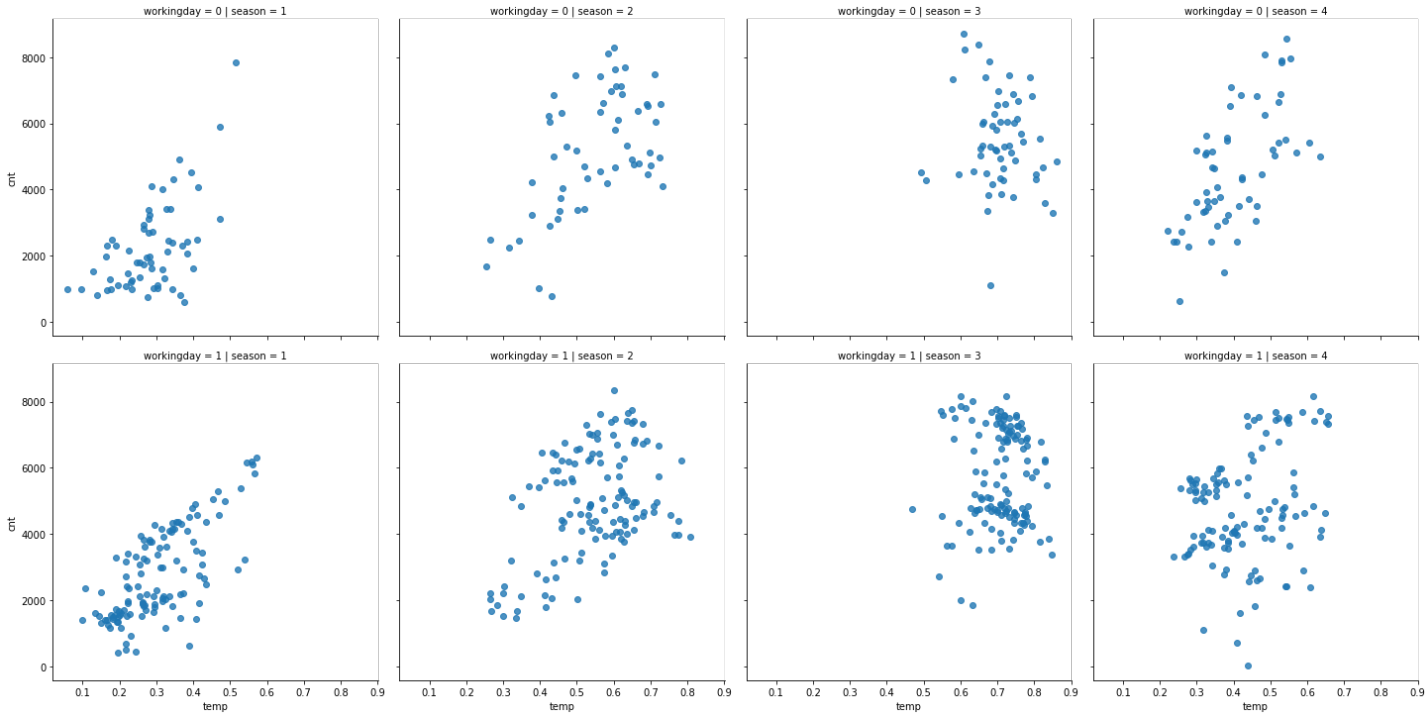
In [ ]:

```
sns.lmplot('temp','cnt',row='workingday',col='season',data=day_df,palette='RdBu_r',fit_reg=False)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x7f13333a2f10>
```



In [ ]:

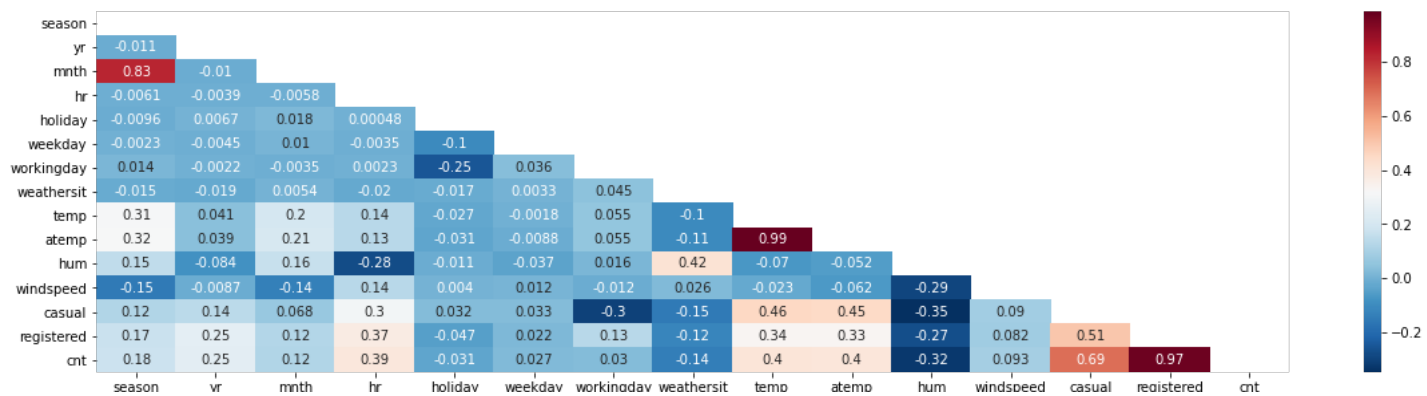
```
hour_df.drop('instant',axis=1,inplace=True)
```

In [ ]:

```
plt.figure(figsize=(20,5))
mask = np.zeros_like(hour_df.corr(), dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(hour_df.corr(),cmap='RdBu_r',mask=mask, annot=True)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1333599650>
```

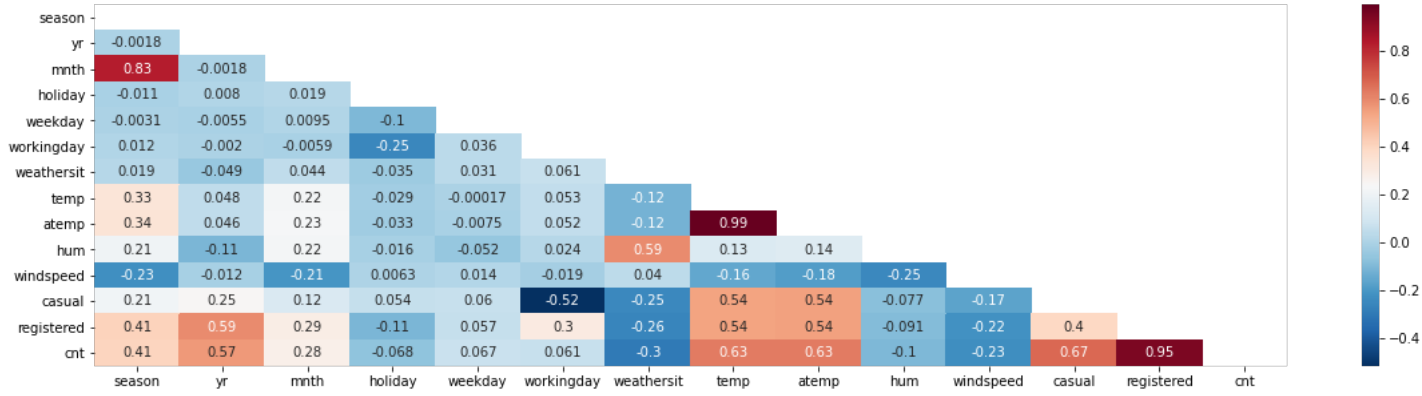


In [ ]:

```
plt.figure(figsize=(20,5))
mask = np.zeros_like(day_df.corr(), dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
sns.heatmap(day_df.corr(),cmap='RdBu_r',mask=mask, annot=True)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1331695fd0>
```



In [ ]:

```
df=pd.merge(day_df,hour_df,how='left',left_on='dteday',right_on='dteday')
```

In [ ]:

```
df.head()
```

Out[ ]:

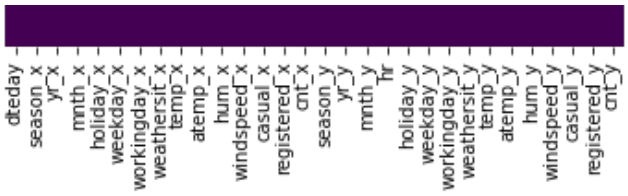
	dteday	season_x	yr_x	mnth_x	holiday_x	weekday_x	workingday_x	weathersit_x	temp_x	atemp_x	hum_x	windspe
0	**2011-01-01**	**1**	**0**	**1**	**0**	**6**	**0**	**2**	**0.344167**	**0.363625**	**0.805833**	**0.16**
1	**2011-01-01**	**1**	**0**	**1**	**0**	**6**	**0**	**2**	**0.344167**	**0.363625**	**0.805833**	**0.16**
2	**2011-01-01**	**1**	**0**	**1**	**0**	**6**	**0**	**2**	**0.344167**	**0.363625**	**0.805833**	**0.16**
3	**2011-01-01**	**1**	**0**	**1**	**0**	**6**	**0**	**2**	**0.344167**	**0.363625**	**0.805833**	**0.16**
4	**2011-01-01**	**1**	**0**	**1**	**0**	**6**	**0**	**2**	**0.344167**	**0.363625**	**0.805833**	**0.16**

In [ ]:

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f13314c7610>
```

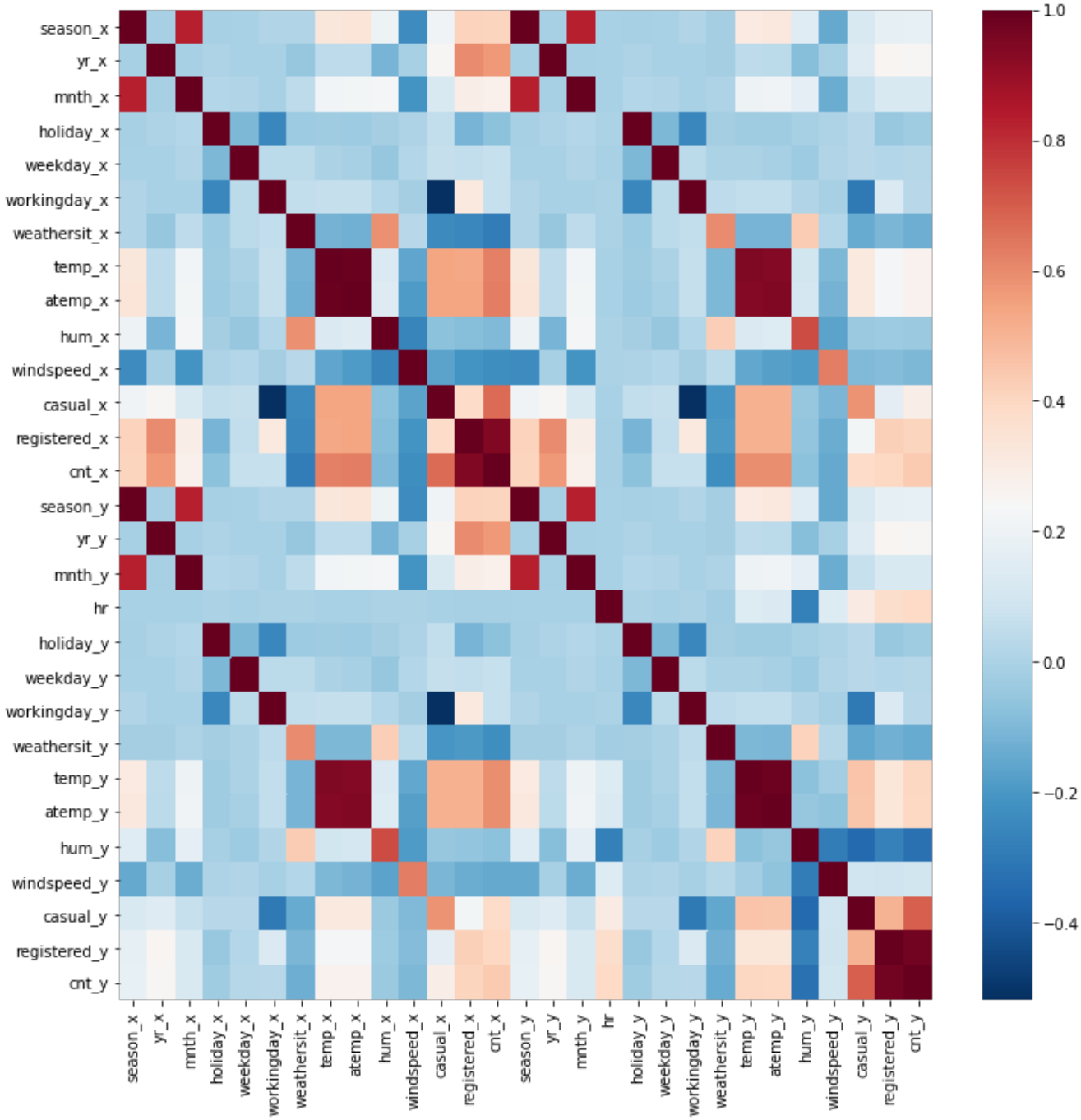


In [ ]:

```
plt.figure(figsize=(12,12))
sns.heatmap(df.corr(),cmap='RdBu_r')
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f13314c2350>
```



In [ ]:

```
X=df.drop(['dteday','cnt_y'],axis=1)
y=df['cnt_y']
```

In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['dteday', 'season_x', 'yr_x', 'mnth_x', 'holiday_x', 'weekday_x',
       'workingday_x', 'weathersit_x', 'temp_x', 'atemp_x', 'hum_x',
       'windspeed_x', 'casual_x', 'registered_x', 'cnt_x', 'season_y', 'yr_y',
       'mnth_y', 'hr', 'holiday_y', 'weekday_y', 'workingday_y',
       'weathersit_y', 'temp_y', 'atemp_y', 'hum_y', 'windspeed_y', 'casual_y',
       'registered_y', 'cnt_y'],
      dtype='object')
```

In [ ]:

```
from sklearn.model_selection import train_test_split
```

In [ ]:

```
X_train, X_test, y_train, y_test= train_test_split(X,y)
```

In [ ]:

```
from sklearn.linear_model import LinearRegression
```

In [ ]:

```
lm=LinearRegression()
```

In [ ]:

```
lm.fit(X_train,y_train)
```

Out[ ]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [ ]:

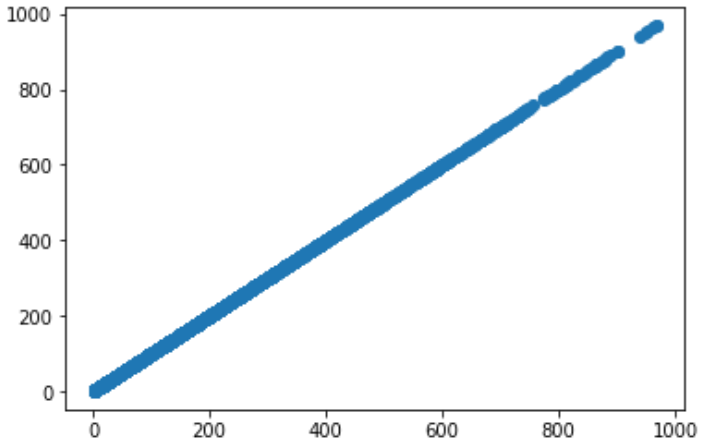
```
predictions=lm.predict(X_test)
```

In [ ]:

```
plt.scatter(y_test,predictions)
```

Out[ ]:

```
<matplotlib.collections.PathCollection at 0x7f133137b050>
```



In [ ]:

```
from sklearn import metrics
```

In [ ]:

```
print('MAE= ', metrics.mean_absolute_error(y_test,predictions))
print('MSE= ', metrics.mean_squared_error(y_test,predictions))
print('RMS= ', np.sqrt(metrics.mean_squared_error(y_test,predictions)))
```

```
MAE=  2.6609581179598595e-12
MSE=  1.2570070510448574e-23
```

RMS= 3.545429524112498e-12

In [ ]:

```
pd.DataFrame(data=lm.coef_, index=X.columns, columns=['Coefficient'])
```

Out[ ]:

	Coefficient
season_x	-4.060769e-15
yr_x	-7.403397e-07
mnth_x	-1.035950e-06
holiday_x	-3.548994e-07
weekday_x	6.743899e-07
workingday_x	-2.895134e-08
weathersit_x	9.344258e-15
temp_x	-6.270922e-13
atemp_x	1.789259e-13
hum_x	1.123360e-14
windspeed_x	2.788286e-14
casual_x	7.311627e-03
registered_x	7.311627e-03
cnt_x	-7.311627e-03
season_y	-1.963652e-15
yr_y	7.403396e-07
mnth_y	1.035950e-06
hr	-2.473795e-15
holiday_y	3.548992e-07
weekday_y	-6.743899e-07
workingday_y	2.895137e-08
weathersit_y	-1.063208e-15
temp_y	5.115475e-13
atemp_y	-1.709650e-13
hum_y	5.480059e-14
windspeed_y	-1.628038e-15
casual_y	1.000000e+00
registered_y	1.000000e+00

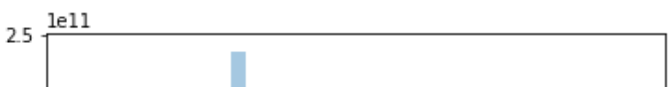
In [ ]:

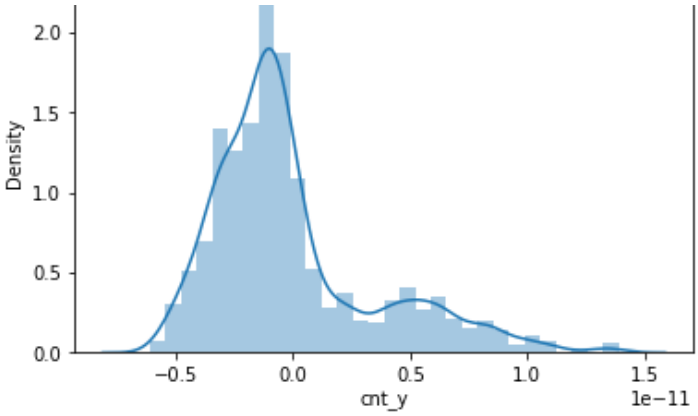
```
sns.distplot(y_test-predictions,bins=30)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `dis
tplot` is a deprecated function and will be removed in a future version. Please adapt you
r code to use either `displot` (a figure-level function with similar flexibility) or `his
tplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f132c1a2090>
```





In [ ]:

```
#End of the project
```

Conclusion

The analysis gives us a detailed visualisation of the Median Bike Sharing in Washington D.C. We noticed that the major renting have a huge difference in median bike sharing with respect to the climate. Most of the sales are achieved when the weather fits perfectly. The count of the bikes are also striking, which the analysis concludes that the median count hovers around heavy density.

From these analysis, marketing team can focus on targeted marketing easily