**Scope Of Project:**

Now a days large amount of data is being generated in the world as a Data Scientist we need to process the data in a appropriate format ,we can manuplate the data ,visualize the data in understandable format.

The aim of our Project is to load the data and manuplate the data as per requirment and visualize the data in understandable format

**Technologies and tools:**

**Software Requirments :**

**→**Languages :Python

**→**Packages :numpy, pandas, matplotlib,seaborn

**Hardware Requirments:**

**→** RAM :4GB

**Importing Libraries:**

**pandas:** Used to load the data,The dataset is in csv format . pandas is used to read csv data.

**numpy:**NumPy is a general-purpose array-processing package.It provides a high-performance multidimensional array object, and tools for working with these arrays.

**matplotlib:**it is used to visualization of data. visualizing data is important to understand easily.

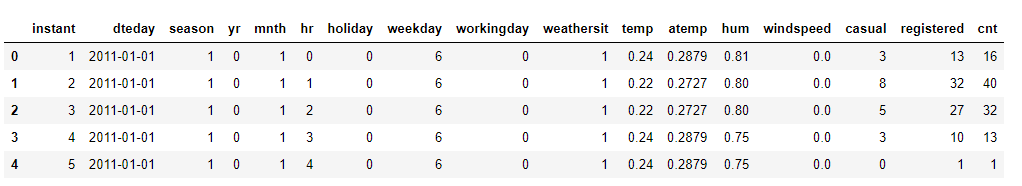
**Seaborn:**seaborn is also used for visualization of data

1. import numpy as np
2. import pandas as pd
3. import matplotlib.pyplot as plt
4. import seaborn as sns

**Load Data:**

In pandas read\_csv() is used to read csv file

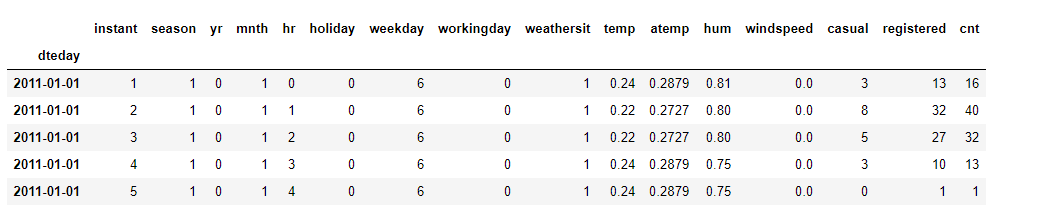
1. dataset=pd.read\_csv('hour.csv')
2. dataset.head()



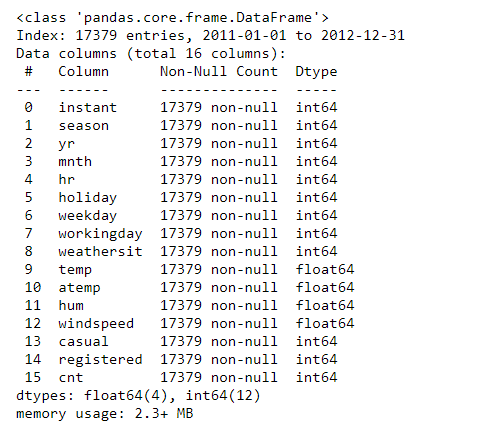
1. dataset.shape



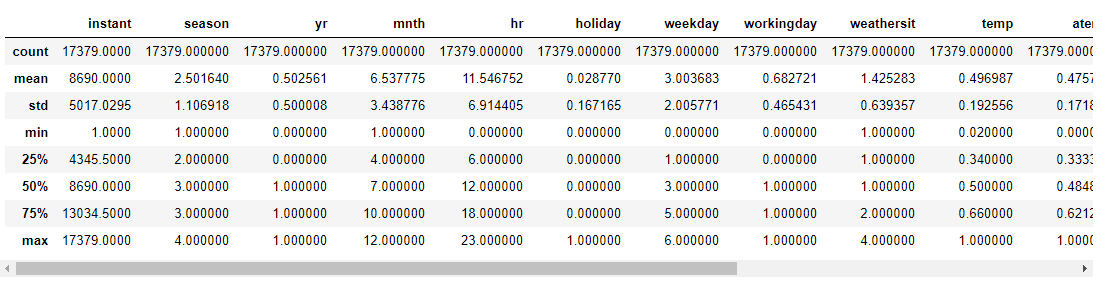
1. dataset.index=dataset['dteday']
2. dataset=dataset.drop(columns=['dteday'])
3. dataset.head()



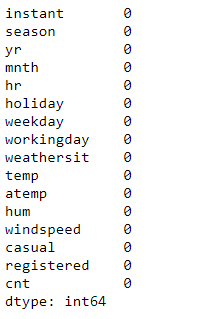
1. dataset.info()



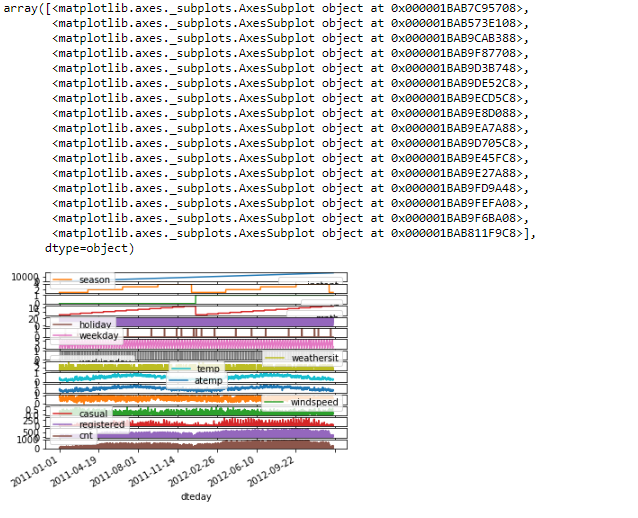
1. dataset.describe()



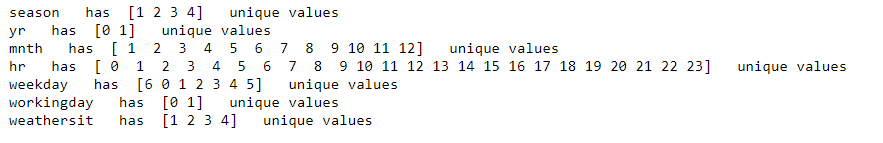
1. dataset.isnull().sum()



1. dataset.plot(subplots=True)

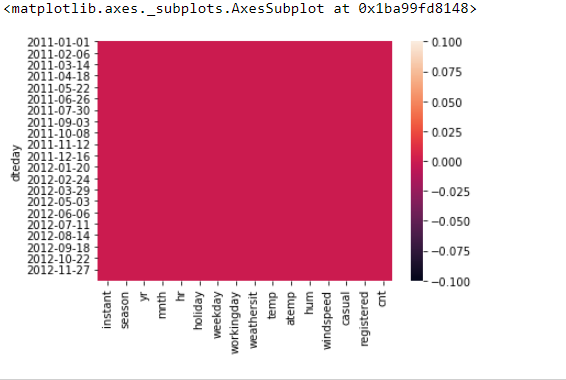


1. col=['season','yr','mnth','hr','weekday','workingday','weathersit']
2. for i in col:
3. print(i,' has ',dataset[i].unique(),' unique values')

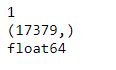


Checking for Null Values

1. sns.heatmap(dataset.isnull())



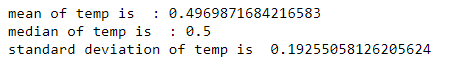
1. temp=dataset['temp']
2. temp=np.array(temp)
3. print(temp.ndim)
4. print(temp.shape)
5. print(temp.dtype)



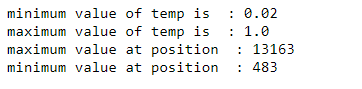
1. temp=np.reshape(temp,(-1,1))
2. print(temp.shape)
3. print(temp.ndim)



1. print('mean of temp is :',np.mean(temp))
2. print('median of temp is :',np.median(temp))
3. print('standard deviation of temp is ',np.std(temp))



1. print('minimum value of temp is :',temp.min())
2. print('maximum value of temp is :',temp.max())
3. print('maximum value at position :',temp.argmax())
4. print('minimum value at position :',temp.argmin())



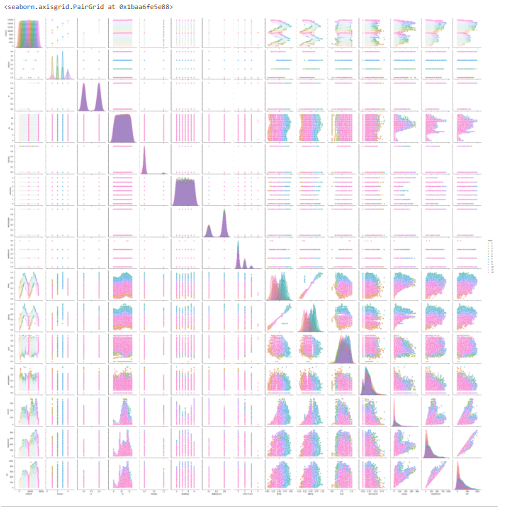
1. hum=dataset['hum']
2. hum=np.array(hum)
3. hum=np.reshape(hum,(-1,1))
4. print(hum.shape)
5. print(hum.ndim)



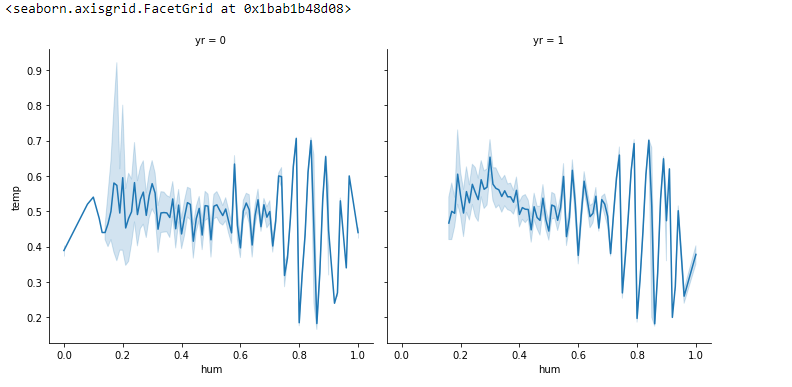
1. hum=hum.flatten()
2. temp=hum.flatten()
3. dot\_result=np.dot(temp,hum)
4. print(dot\_result)



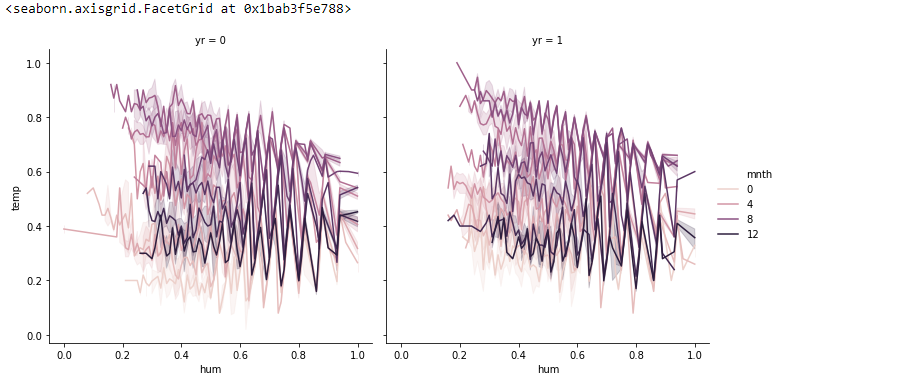
1. sns.pairplot(dataset,hue='mnth')



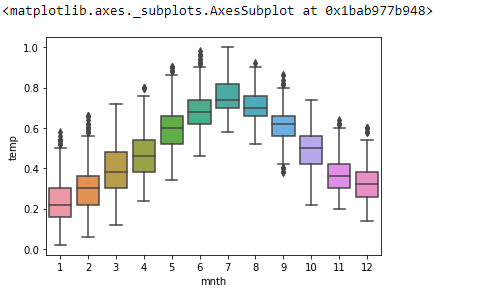
1. sns.relplot(x='hum',y='temp',col='yr',data=dataset,kind='line')



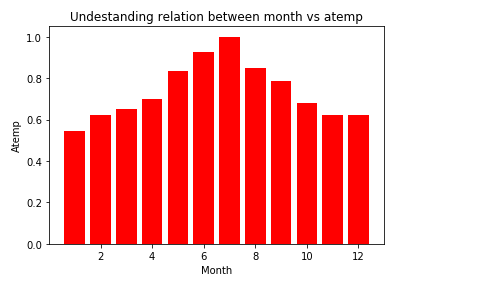
1. sns.relplot(x='hum',y='temp',col='yr',data=dataset,hue='mnth',kind='line')



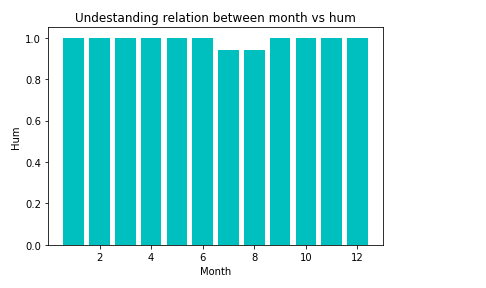
1. sns.boxplot(x='mnth',y='temp',data=dataset)



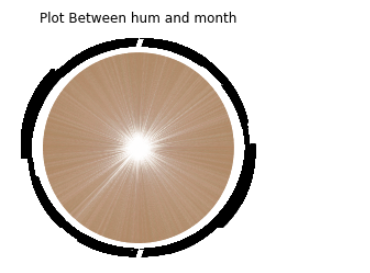
1. plt.bar(dataset['mnth'],dataset['atemp'],color='r')
2. plt.xlabel('Month')
3. plt.ylabel('Atemp')
4. plt.title('Undestanding relation between month vs atemp')
5. plt.show()



1. plt.bar(dataset['mnth'],dataset['hum'],color='c')
2. plt.xlabel('Month')
3. plt.ylabel('Hum')
4. plt.title('Undestanding relation between month vs hum')
5. plt.show()



1. plt.pie(dataset['hum'],colors=['r','g'],labels=dataset['mnth'])
2. plt.title('Plot Between hum and month')
3. plt.show()



1. plt.hist(dataset['cnt'])
2. plt.title('Graph for cnt')
3. plt.ylabel('cnt')
4. plt.show()

