jupyter QQ Plot Presentation Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

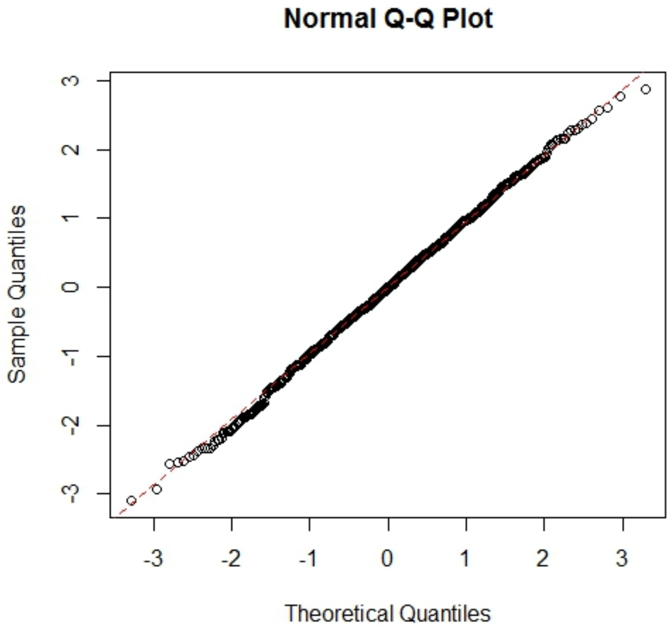
Run Markdown

QQ Plot (Quantile- Quantile Plot)

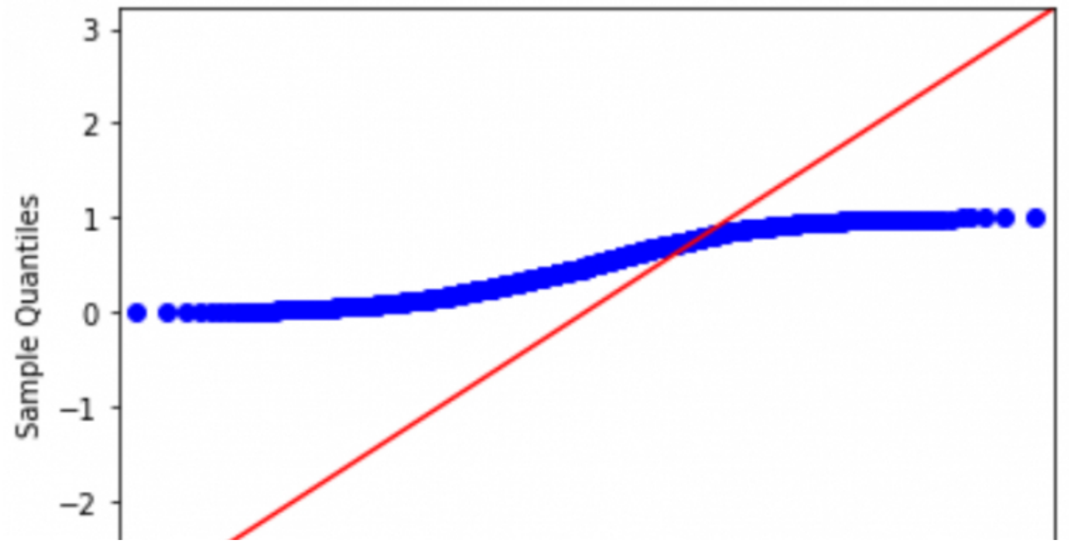
1.Introduction

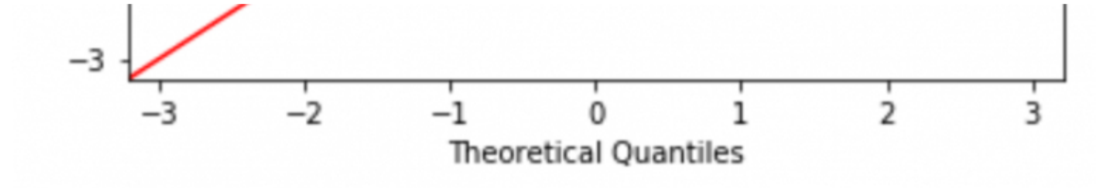
- A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another.
- If both sets of quantiles came from the same distribution, we should see the points forming a line that's roughly straight.
- x axis displays the theoretical quantiles.
- y axis displays the actual data

QQ Plot for a normally distributed sample



QQ Plot of samples which are not normally distributed





2.Where the visualization can be used

- Some machine learning models like linear and logestic regression assume that the variables are normally distributed.
- The normal distributed varaiables may boost the machine learning algorithm performance.
- So we can use QQ plot to check a set of observations are normally distributed.

3.Python Code for QQ Plot

Libraries used

- pandas: It offers data structures and operations for manipulating numerical tables.
- numpy: Python library used for working with arrays
- matplotlib: Used for visualization.
- scipy: Is a library that uses NumPy for more mathematical functions.

In [43]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.stats as stats
```

Dataset Used

- We are using titanic data set.
- For our purpose we are only taking 3 columns Age,Fare and surivive.
- Top 5 datas are shown below

In [44]:

```
data=pd.read_csv('titanic.csv',usecols=['Age','Fare','Survived'])
data.head()
```

Out[44]:

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

Data preprocessing

- When we check for null values, we can find that the age contains 177 null values.
- We are removing rows with null value.

In [45]:

```
data.isnull().sum()
```

Out[45]:

```
Survived      0
Age         177
Fare          0
dtype: int64
```

In [46]:

```
df = data[pd.notnull(data['Age'])]
df.head()
```

Out[46]:

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000

```
4          0  35.0   8.0500
```

- We can see that all the null values are removed.

In [47]:
```
df.isnull().sum()
```

Out[47]:
```
Survived    0
Age         0
Fare        0
dtype: int64
```

QQ Plot

- A function is defined to draw the QQ plot.
- Two parameters are accepted the dataset and the varaiable.
- QQ plot is drawn using calling stats.probplot()

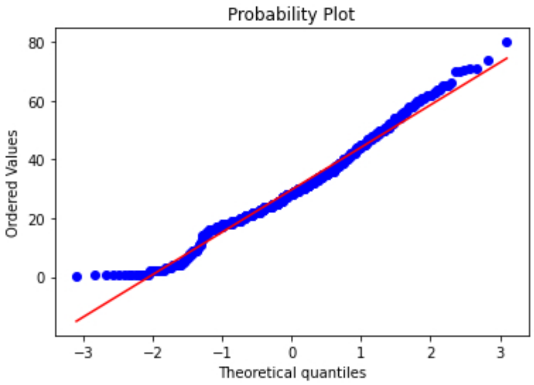
In [48]:
```
def diagnostic_plots(df, variable):

    plt.subplot(1, 1, 1)
    stats.probplot(df[variable], dist="norm", plot=plt)

    plt.show()
```

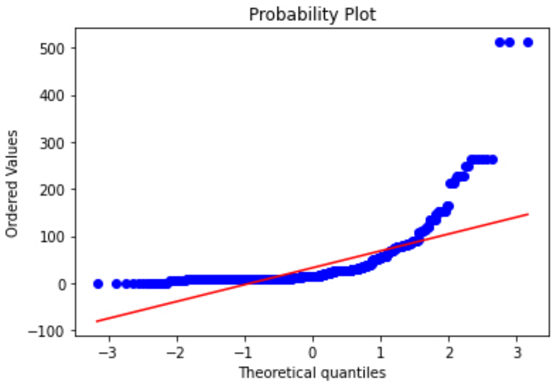
- QQ plot for Age column.
- We can see that most of the points are near the line.

In [49]:
```
diagnostic_plots(df, 'Age')
```



- QQ plot for Fare column.
- We can see that most of the points are not on the red line, so we can say it is not a linear distributed.

In [50]:
```
diagnostic_plots(data, 'Fare')
```

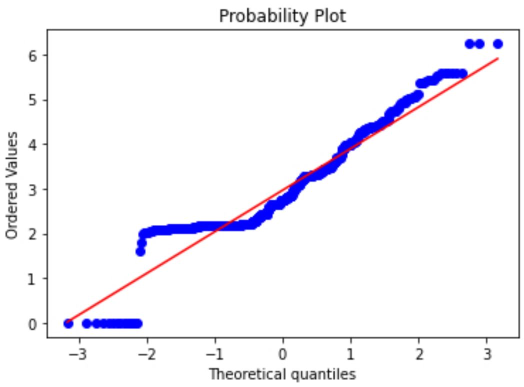


- If a variable is not normally distributed, sometimes it is possible to find a mathematical transformation.
- One of such transformation is Logarithmic transformation.
- Here we are doing this logarithmic transformation to the 'fare'.
- After the transformation we can see that it is better than the last one.

In [51]:

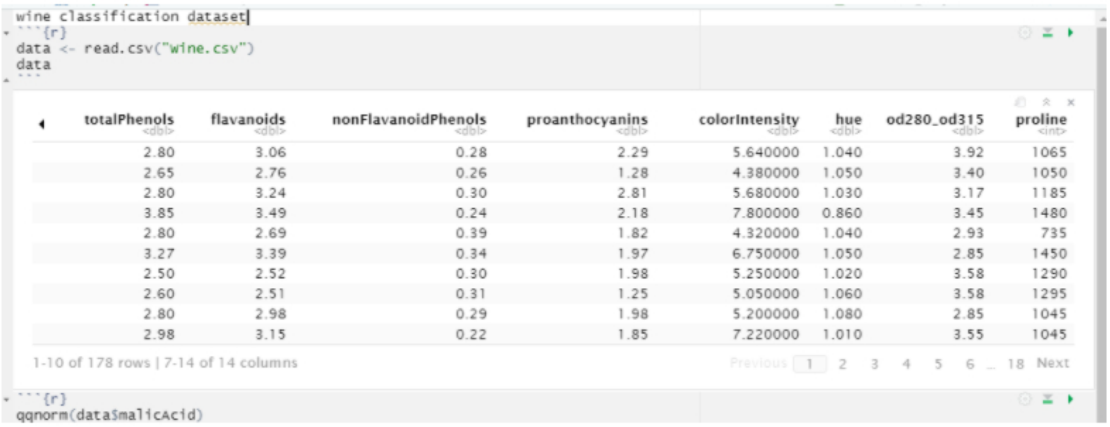
```python
data['Log_Fare']=np.log(data['Fare']+1)
diagnostic_plots(data,'Log_Fare')

```



4. R code for QQ plot

- Here we are using wine classification dataset.
- We are mainly considering hue and malicAcid columns for ploting.



wine classification dataset

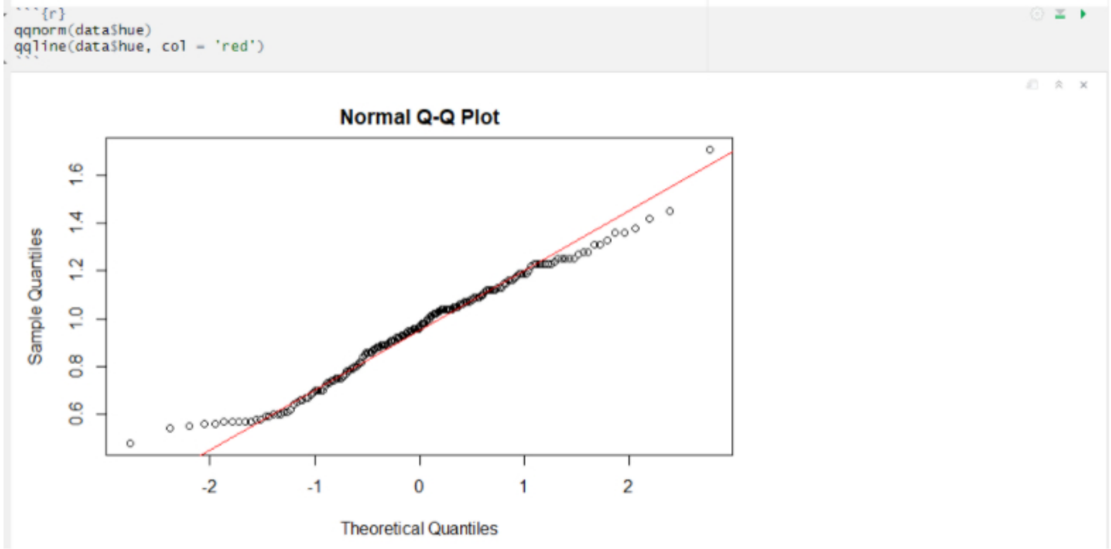
~~~
```{r}
data <- read.csv("wine.csv")
data
```
~~~

	totalPhenols <dbl>	flavanoids <dbl>	nonFlavanoidPhenols <dbl>	proanthocyanins <dbl>	colorIntensity <dbl>	hue <dbl>	od280_od315 <dbl>	proline <int>
	2.80	3.06	0.28	2.29	5.640000	1.040	3.92	1065
	2.65	2.76	0.26	1.28	4.380000	1.050	3.40	1050
	2.80	3.24	0.30	2.81	5.680000	1.030	3.17	1185
	3.85	3.49	0.24	2.18	7.800000	0.860	3.45	1480
	2.80	2.69	0.39	1.82	4.320000	1.040	2.93	735
	3.27	3.39	0.34	1.97	6.750000	1.050	2.85	1450
	2.50	2.52	0.30	1.98	5.250000	1.020	3.58	1290
	2.60	2.51	0.31	1.25	5.050000	1.060	3.58	1295
	2.80	2.98	0.29	1.98	5.200000	1.080	2.85	1045
	2.98	3.15	0.22	1.85	7.220000	1.010	3.55	1045

1-10 of 178 rows | 7-14 of 14 columns

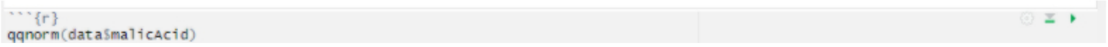
~~~
```{r}
qqnorm(data$malicAcid)
~~~

- qqnorm() is used to draw qq plot on R
- qqline() is used to draw the 45 degree line.
- We can see that hue varaiable is almost linearly distributed.

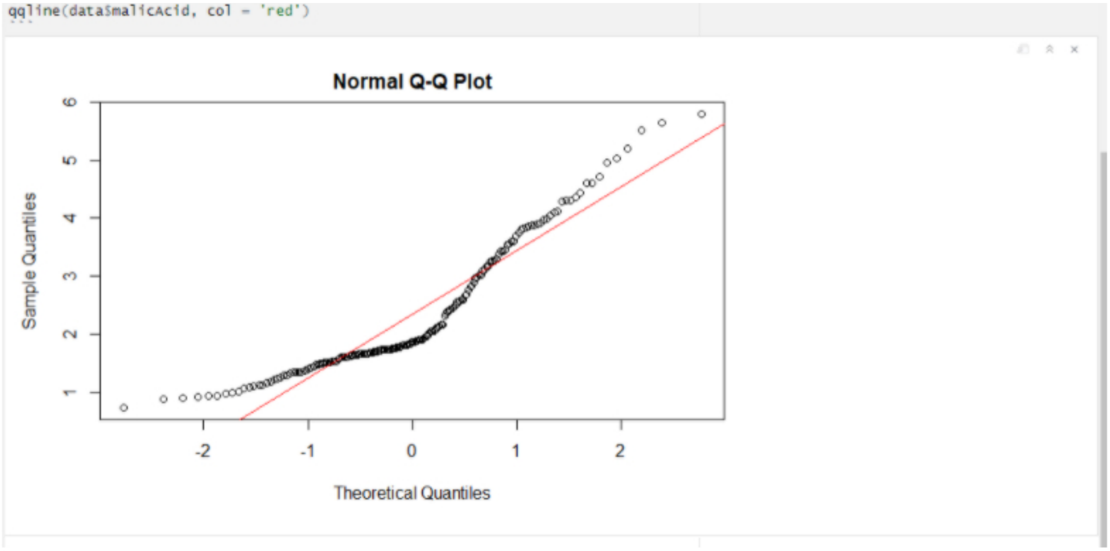


~~~
```{r}
qqnorm(data$hue)
qqline(data$hue, col = 'red')
```
~~~

- We can see that malicAcid varaiable is not linearly distributed. As most of the points doesnt pass through the line.



~~~
```{r}
qqnorm(data$malicAcid)
~~~



~~~
qqline(data$malicAcid, col = 'red')
```
~~~

5. Purpose of the visualization

- In most cases, this plot is used to determine whether or not a set of data follows a normal distribution.
- All point lie on or close to straight line at an angle of 45 degree from x – axis. It indicates that the samples have similar distributions.

Thank You

```
In [ ]:  1
```