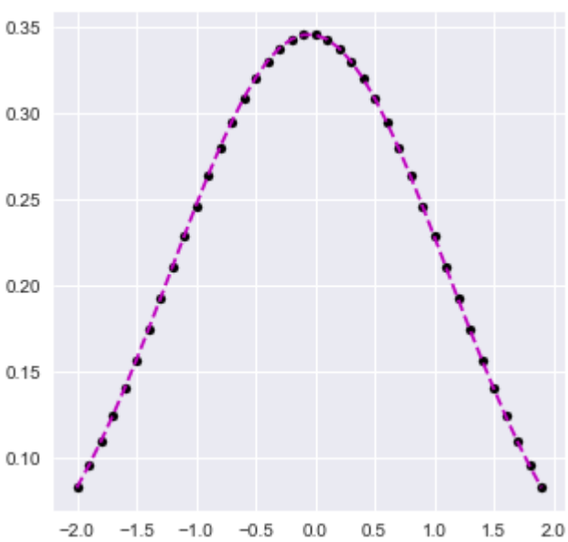
In [ ]:
```
# N.D is use to check for Numeric Variables
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
# Draw the Normal distribution

def pdf(x):
    mean = np.mean(x)
    std = np.std(x)
    y_out = 1/(std * np.sqrt(2 * np.pi)) * np.exp( - (x - mean)**2 / (2 * std**2))
    return y_out

# To generate an array of x

x = np.arange(-2, 2, 0.1)
y = pdf(x)

# plotting the normal curve / bell curve or Gaussian distribution
plt.style.use('seaborn')
plt.figure(figsize=(5,5))
plt.plot(x, y, color = 'm', linestyle = '--')
plt.scatter(x, y,  marker= 'o', s = 25, color = 'k')
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x148a5539100>
```



Normal Distribution and its tests

1. import datasets
2. subsetting a dataset
3. visual test for normal distribution
    A. Histogram
    B. qq norm plot
4. statistical test
    A. Shapiro Wilk Test
    B. D' Agostino's K^2 Test
    C. Anderson-Darling Test

In [ ]:
```
# 1. import a dataset
kashti = sns.load_dataset('titanic')
kashti.head()
```

Out[ ]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C

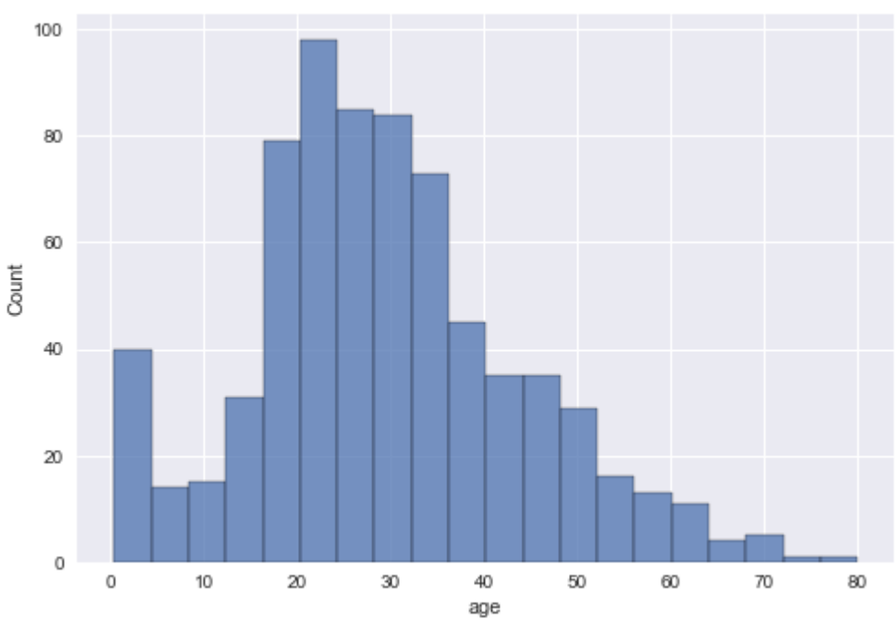
In [ ]:
```
# 2. subsetting a dataset
kashti1 = kashti[['sex', 'age', 'fare']]
kashti1.head()
```

Out[ ]:

	sex	age	fare
0	male	22.0	7.2500
1	female	38.0	71.2833
2	female	26.0	7.9250
3	female	35.0	53.1000
4	male	35.0	8.0500

In [ ]:
```
# 3. visual test (HIstogram)
sns.histplot(kashti1['age'])  # it is about normal distribution
```

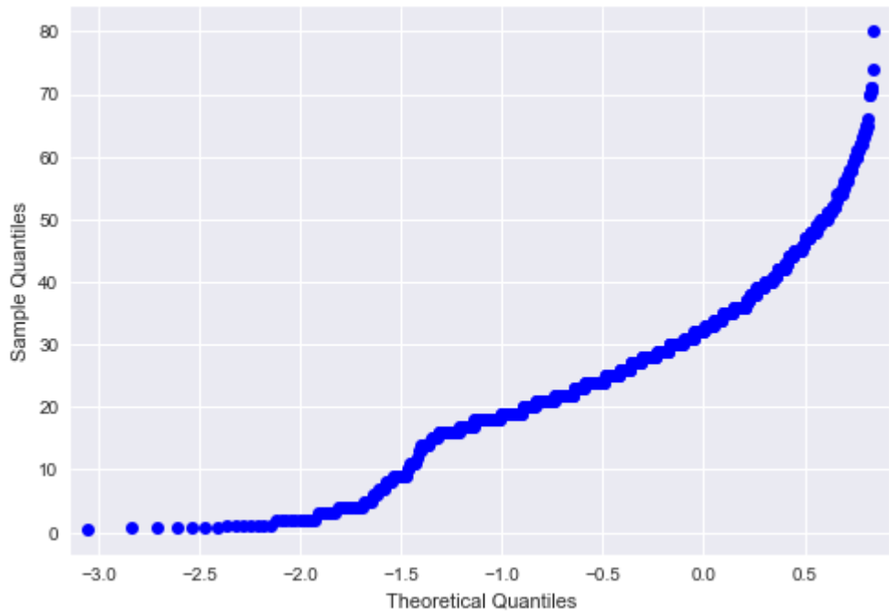
Out[ ]:
```
<AxesSubplot:xlabel='age', ylabel='Count'>
```



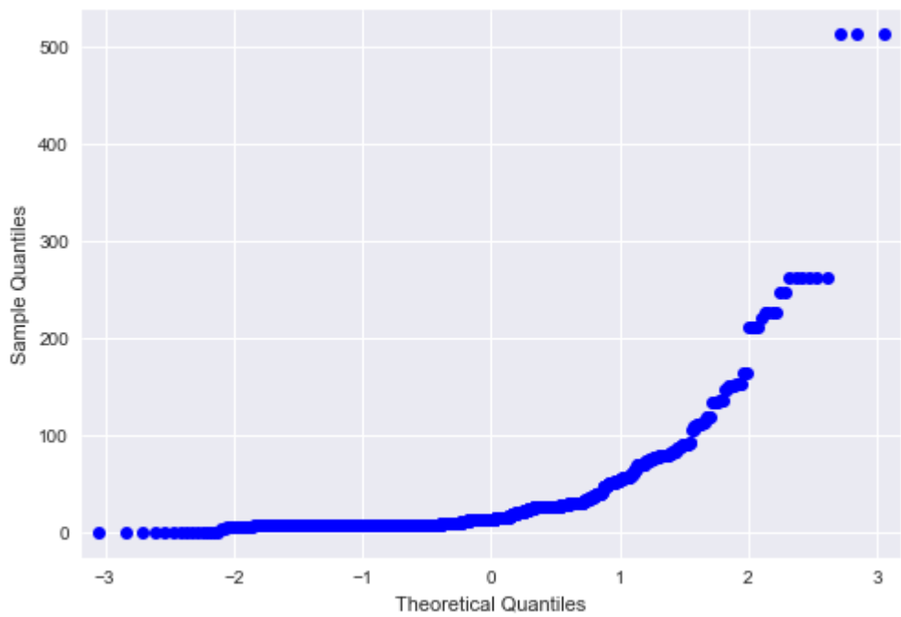
In [ ]:
```
# 3. visual test (HIstogram)
sns.histplot(kashti1['fare'])   # its tell it is not normal
```

Out[ ]:
```
<AxesSubplot:xlabel='fare', ylabel='Count'>
```

In [ ]:
```
# 3. visual test (qq norm test)
from statsmodels.graphics.gofplots import qqplot
qqplot(kashti1['age'])                                  # Told us that it is about No
plt.show()
```



In [ ]:
```
# 3. visual test (qq norm test)
from statsmodels.graphics.gofplots import qqplot
qqplot(kashti1['fare'])                                  # Told us that it is not a N
plt.show()
```



4. **Statistical Test for Normality**

There are many tests that we can use to quantify whether a sample of data looks as though it was drawn from a Gaussian Distribution.\ Each test makes diff assumptions and consider diff aspects of the data.\ We will look at 3 commonly used tests in this section that you can apply to your own data sample.\

1. Shapiro Wilk Test
2. D' Agostino's K^2 Test
3. Anderson-Darling Test

> **p <= alpha(0.05): reject HO, not normal.\ p >= alpha(0.05): fails to reject HO, normal.**

1- Shapiro Wilk Test (Best One)

The Shapiro Wilk Test eveluate a data sample and quantifies how likely it is that the data was drawn from a Gaussian Distribution, named for Samual Shapiro and Martin Wilk.

In practice the Shapiro Wilk test is belived to be a reliable test for normality, although there is some suggestion tht the test may be suitable for smaller sample of data, e.g, thousands of observations or fewer.

The Shapiro() scipy function will calculate the Sahpiro Wilk on a given dataset. The function returns both the w-statistic calculated by the test and the p-value

Assumptions

- Observations in each sample are independent and identically distributed.

Interpretation

- HO: the sample has a Gaussian Distribution.
- H1: the sample does not have a Gaussian Distribution.

Python Code is here:

In [ ]:
```
# Shapiro Wilk Test
from scipy.stats import shapiro
stat, p = shapiro(kashti1['age'])
print('stat=%.3f, p=%.3f' % (stat, p))
if p>0.05:
    print('Probably Gaussian or Normal Distribution')
else:
    print('Probably not Gaussian nor Normal Distribution')
```

```
stat=nan, p=1.000
Probably Gaussian or Normal Distribution
```

In [ ]:
```
# Shapiro Wilk Test
from scipy.stats import shapiro
stat, p = shapiro(kashti1['fare'])
print('stat=%.3f, p=%.3f' % (stat, p))
if p>0.05:
    print('Probably Gaussian or Normal Distribution')
else:
    print('Probably not Gaussian nor Normal Distribution')
```

```
stat=0.522, p=0.000
Probably not Gaussian nor Normal Distribution
```

2- D'Agostino's K^2 Test

The D'Agostino's K^2 Test calculuate summary statistics from the data, namely kurtosis and skewness to determine if the data distribution departs from the Gaussian Distribution, named for D'Agostino's.

- **Skew** is a quantification of how much a distribution is pushed left or right, a measure of asymmetry in the distribution.
- **Kurtosis** quantifies how much of the distribution is in the tail. it is simple and commonly used statistical test for normality.

The D'Agostino's K^2 test is available via the normaltest() Scipy function and return the test statistics and p-value.

Assumptions

- Observations in each sample are independent and identically distributed.

Interpretation

- HO: the sample has a Gaussian Distribution.
- H1: the sample does not have a Gaussian Distribution.

Python Code is here:

In [ ]:
```
# D' Agostino's K^2 Test
from scipy.stats import normaltest
stat, p = normaltest(kashti1['age'])
print('stat=%.3f, p=%.3f' % (stat, p))
if p>0.05:
    print('Probably Gaussian or Normal Distribution')
else:
    print('Probably not Gaussian nor Normal Distribution')
```

```
stat=nan, p=nan
Probably not Gaussian nor Normal Distribution
```

In [ ]:
```
# D' Agostino's K^2 Test
from scipy.stats import normaltest
stat, p = normaltest(kashti1['fare'])
print('stat=%.3f, p=%.3f' % (stat, p))
if p>0.05:
    print('Probably Gaussian or Normal Distribution')
else:
    print('Probably not Gaussian nor Normal Distribution')
```

```
stat=904.587, p=0.000
Probably not Gaussian nor Normal Distribution
```

3- Anderson-Darling Test

The Anderson-Darling Test can be used to eveluate whether a data sample comes from one of among many known data samples, named for Theodore Anderson and Donald Darling.

It can be used to check whether a dataset is normal. The test is a modified version of a more sophisticated nonparametric goodness-of-fit statistical test called the Kolmogorov-Smirnov test.

A feature of the Anderson-Darling test is that it returns a list of critical values rather than a single p-value. This can provide a basic of more thorough interpretation of the result. The anderson() Scipy feature function impliments the Anderson-Darling test.It takes as parameters the data sample and the name of the distribution to test it against. By default the test will check the Gaussian distribution.

Assumptions

- Observations in each sample are independent and identically distributed.

Interpretation

- HO: the sample has a Gaussian Distribution.
- H1: the sample does not have a Gaussian Distribution.

Python Code is here:

In [ ]:
```
# Anderson-Darling Test
from scipy.stats import anderson
result = anderson(kashti1['age'])
print('stat=%.3f'  % (result.statistic))
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < cv:
        print('Probably Gaussian or Normal Distribution at the %0.1f%% level' % (sl))
    else:
        print('Probably not Gaussian nor Normal Distribution at the %0.1f%% level' %
```

```
stat=nan
Probably not Gaussian nor Normal Distribution at the 15.0% level
Probably not Gaussian nor Normal Distribution at the 10.0% level
```

```
Probably not Gaussian nor Normal Distribution at the 5.0% level
Probably not Gaussian nor Normal Distribution at the 2.5% level
```

In [ ]: