Statistical Inference

Normality Assessment and One Sample t test using Python

DATA SCIENCE INSTITUTE

Contents

DATA SCIENCE
INSTITUTE

Case Study

To assess normality of data in Python, we shall consider the below case as an example.

Background

Data has 2 variables recorded for 80 guests in a large hotel.
Customer Satisfaction Index (csi) & Total Bill Amount in thousand Rs. (billamt)

Objective

To check if variables follow normal distribution

Sample Size

Sample size: 80
Variables: id, csi, billamt

Data Snapshot

Normality Testing Data

Variables

Observations

id	csi	billamt
1	38.35	34.85
2	47.02	10.99
3	36.96	24.73
4	43.07	7.9
5	38.77	9.38

Column	Description	Type	Measurement	Possible Values
id	Customer ID	Numeric	-	-
csi	Customer Satisfaction Index	Numeric	-	Positive value
billamt	Total Bill Amount in thousand Rs.	Numeric	Rs.	Positive value

DATA SCIENCE INSTITUTE

Quantile-Quantile plot

- Very powerful graphical method of assessing Normality.
- Quantiles are calculated using sample data and plotted against expected quantiles under Normal distribution.
- If Normality assumption is valid then high correlation is expected between sample quantiles and expected(theoretical quantiles under normal distribution) quantiles.
- The Y axis plots the actual quantiles values based on sample. The X axis plots theoretical values.
- If the data is truly sampled from a Normal distribution, the QQ plot will be linear.

QQ Plot in Python For Variable csi

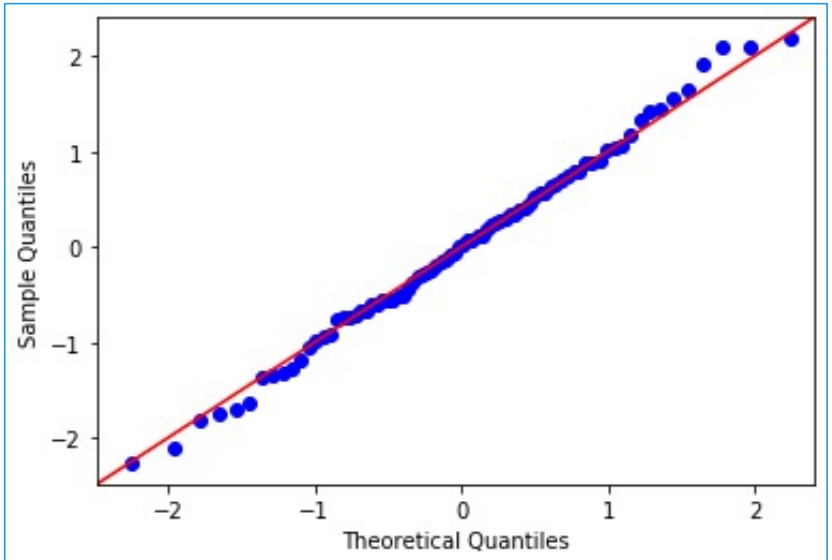
```
#Import Data
import pandas as pd
data=pd.read_csv('Normality Testing Data.csv')
```

```
#QQ Plot
import statsmodels.api as sm
sm.graphics.qqplot(data.csi, line='45', fit=True)
```

- ***qqplot()*** *produces a plot with theoretical quantiles on x axis against the sample quantiles on y axis. Column for which normality is being tested is specified in the first argument.*
- ***line=*** *is an argument that adds reference line to the qqplot. Here it adds a 45-degree line*
- ***fit=True*** *indicates, parameters are fit using the distribution's fit() method.*

QQ Plot in Python For Variable csi

```
# Output:
```



Interpretation :

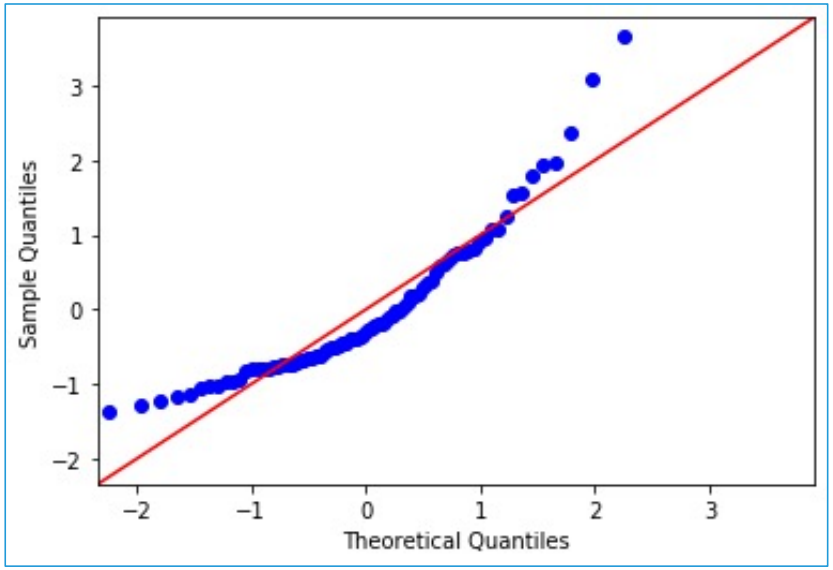
- *Q-Q plot is Linear. Distribution of 'csi' can be assumed to be normal.*

Q-Q plot in Python For Variable billamt

```
# Q-Q plot for the variable billamt
sm.graphics.qqplot(data.billamt, line='45', fit=True)
```

- ***data.billamt*** *is the variable for which normality is to be checked.*

```
# Output:
```



Interpretation :

- *Q-Q plot is deviated from linearity. Distribution of 'billamt' appears to be non-normal.*

DATA SCIENCE INSTITUTE

Shapiro-Wilk test

Shapiro-Wilk test is widely used statistical test for assessing **Normality**.

Objective	To test the **normality** of the data.

Null Hypothesis (H_0): **Sample is drawn from Normal Population**

Alternate Hypothesis (H_1): **Sample is drawn from Non-Normal Population**

The test is performed for the variables, 'csi' and 'billamt' separately.

Test Statistic	It correlates sample ordered values with expected Normal scores. (actual calculation is very complex so we will avoid details)
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

* Here we continue to use previous data of CSI and Bill Amount for Shapiro-Wilk test.

DATA SCIENCE INSTITUTE

Shapiro Wilk Test For Variable csi

```
# Shapiro Wilk Test
import scipy as sp
sp.stats.shapiro(data.csi)

# Output
(0.9919633269309998, 0.9037835597991943)
```

shapiro() *from scipy package, returns correlation coefficient w and p-value.*

Interpretation :

- *Since p-value is >0.05,do not reject H0. Distribution of 'csi' can be assumed to be normal.*

Shapiro-Wilk test For Variable bilamt

```
# Shapiro Wilk test for the variable billamt
sp.stats.shapiro(data.billamt)
```

- ***data.billamt*** *is the variable for which normality is to be checked.*

```
# Output:
(0.8903077244758606, 4.858443844568683e-06)
```

Interpretation :

- *Since p-value is <0.05, reject H0. Distribution of 'billamt' appears to be non-normal.*

DATA SCIENCE INSTITUTE

One sample t-test

- One sample t test is used to test the hypothesis about a single population mean.
- We use one-sample t-test when we collect data on a single sample drawn from a defined population.
- For this design, we have one group of subjects, collect data on these subjects and compare sample statistic to the hypothesized value of population parameter.
- Subjects in the study can be patients, customers, retail stores etc.

Case Study

To execute Parametric test in Python, we shall consider the below case as an example.

Background

A large company is concerned about time taken by employees to complete weekly MIS report.

Objective

To check if average time taken to complete the MIS report is more than 90 minutes

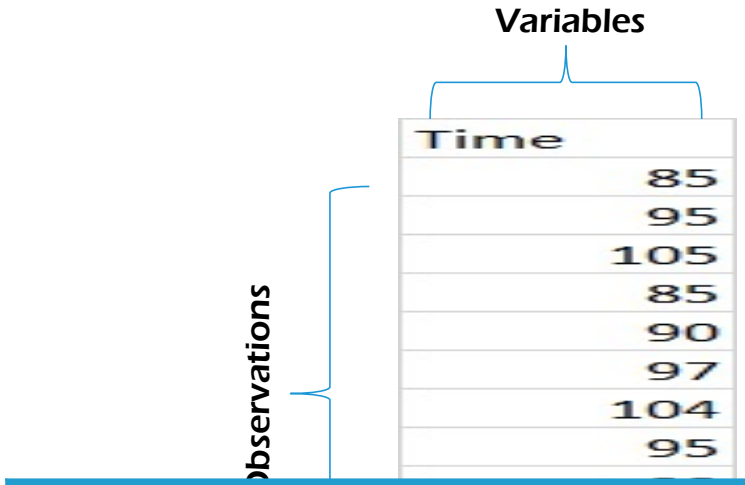
Sample Size

Sample size: 12
Variables: Time

DATA SCIENCE INSTITUTE

Data Snapshot

ONE SAMPLE t TEST



Columns	Description	Type	Measurement	Possible values
Time	Time taken to complete MIS	Numeric	Minutes	Positive Values

DATA SCIENCE INSTITUTE

Assumptions for one sample t-test

- The assumptions of the one-sample t-test are listed below:
  - Random sampling from a defined population
    (employees are selected at random from the company)

  - Population is normally distributed
    (Time taken to complete MIS report should be normally distributed).

  - Variable under study should be continuous.

- Normality test can be performed by any of the methods explained earlier.

- The validity of the test is not seriously affected by moderate deviations from 'Normality' assumption.

One sample t-test

Testing whether mean is equal to a test value.

Objective	To test the average time taken to complete MIS is more than 90 minutes

Null Hypothesis (H_0): $\mu = 90$

Alternate Hypothesis (H_1): $\mu > 90$

Test Statistic	$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$	Where $\bar{x}$ is the sample mean, s is the sample standard deviation, n is the sample size. The quantity t follows a distribution called as 't distribution' with n-1 degrees of freedom.
Decision Criteria	Reject the null hypothesis if p-value < 0.05	

DATA SCIENCE INSTITUTE

Computation

	Notation	Value
Sample Size	n	12
Mean	$\bar{x}$	93.5833
Standard Deviation	S	6.4731
Standard Error	s/√n	1.8686
Difference	$\bar{x}$-μ_0	93.5833-90=3.5833
t	$\frac{\bar{x}-\mu_0}{S.E}$	1.9176

DATA SCIENCE INSTITUTE

One sample t-test in Python

```
# Import data
data2=pd.read_csv('ONE SAMPLE t TEST.csv')

# t-test for one sample
from scipy.stats import ttest_1samp
ttest_1samp(data2.Time, popmean=90,alternative='greater')
```

- *ttest_1samp() from scipy package, returns t and p-value.*
- ***data.time** is the variable under study.*

```
# Output:
Ttest_1sampResult(statistic=1.9176218472595046, pvalue=0.04074043079962237)
```

Interpretation :

- *Since p value <0.05, reject H0. … report is more than 90 minutes ”*

DATA SCIENCE INSTITUTE