Histogram

Normality tests help us understand the chance that any data we have with us may have come from a normal or [Gaussian distribution](http://www.itl.nist.gov/div898/handbook/eda/section3/eda3661.htm).

NORMAL Q-Q PLOTS

#Generating 10k points of data and arranging them into 100 columns

x<-rnorm(10000,10,1)

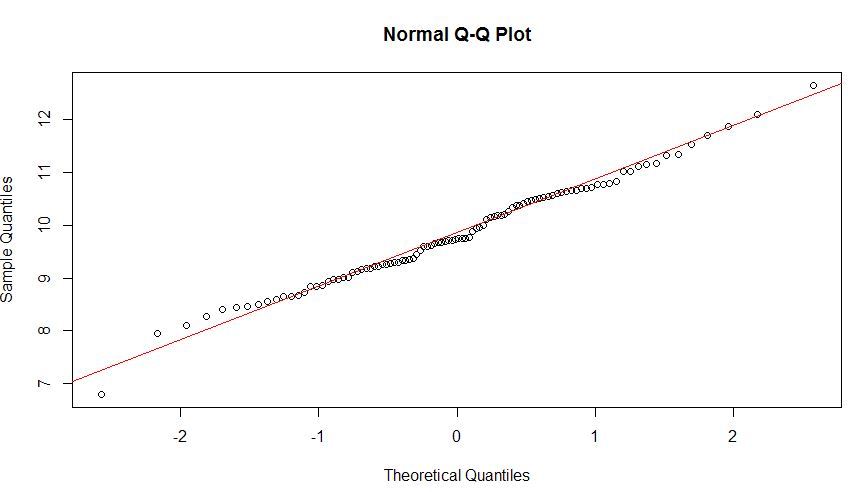
dim(x)<-c(100,100)

#Generating a simple normal quantile-quantile plot for the first column

#Generating a line for the qqplot

qqnorm(x[,1])

qqline (x[,1], col=2)



The Q-Q plot tells us what proportion of the data set (in this case, the first column of variable x), compares with the expected proportion (theoretically) of the normal distribution model based on the sample’s mean and standard deviation

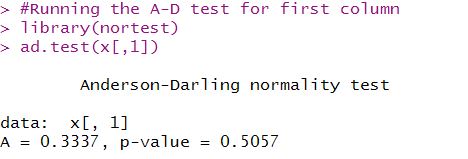
The normal distribution is thicker around the mean, and thinner as you move away from it – specifically, around 68% of the points you can expect to see in normally distributed data will only be 1 standard deviation away from the mean. There are similar metrics for normally distributed data, for 2 and 3 standard deviations (95.4% and 99.7% respectively).

ANDERSON DARLING NORMALITY TEST

As one of the commonly used normality tests, this is very commonly used to tell us whether or not a sample may represent normally distributed data.

This is done in R by using the ad.test() command, in the **“**[nortest](https://cran.r-project.org/web/packages/nortest/nortest.pdf)**“** package

|  |
| --- |
| #Running the A-D test for first column  library(nortest)  ad.test(x[,1]) |



The A-D test uses this test statistic to calculate the probability that this sample could have come from a normal distribution.

p-value tell us probability that we can see the data we see in this sample purely by random chance.

It is one tailed test

INTERPRETING AND UNDERSTANDING A-D TEST RESULTS

Naturally, as the p-values from an Anderson-Darling normality test become smaller and smaller, there is a smaller and smaller chance that we are looking at data from a normal distribution

 Most statistical studies peg the “significance” level at which we reject the default hypothesis (that this data comes from a normal distribution) outright, at p-values of 0.05 (5%) or lesser.

The easiest way to do this in R is to define a function that returns the p-values from each column, and use that in an **“apply”** command.

#Running the A-D test for first column

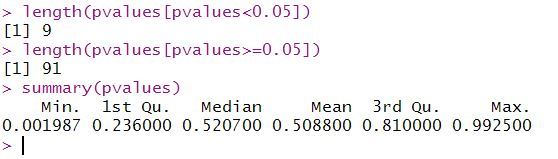
library(nortest)

#defining a function called "adt" to run the A-D tests and return p-values

adt<-function(x){ test<-ad.test(x); return(test$p.value) }

#store the p-values for each column in a separate variable

pvalues<-apply(x,MARGIN = 2, adt)



When you summarize p-values, you can see how approximately 9 of the 100 don’t pass the significance criteria (of p>=0.05).

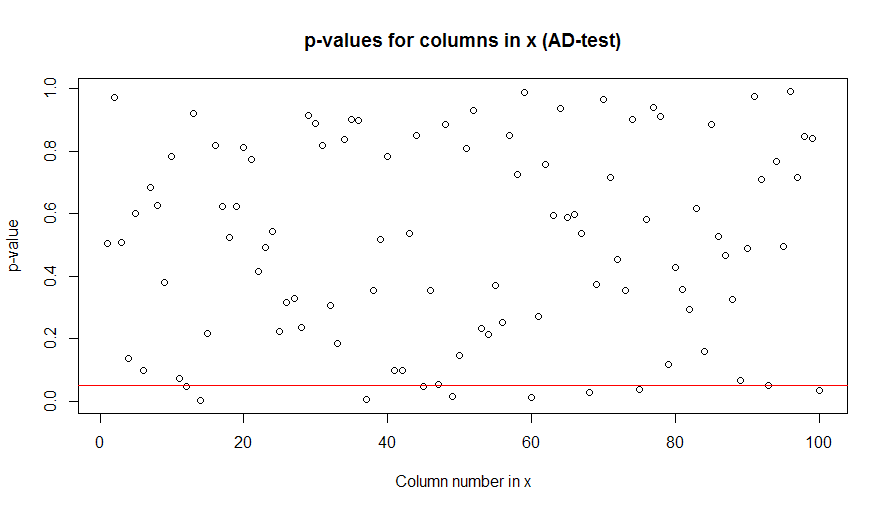
You can also see that the p-values in this set of randomly generated samples are randomly distributed over the entire range of probabilities from 0 to 1.

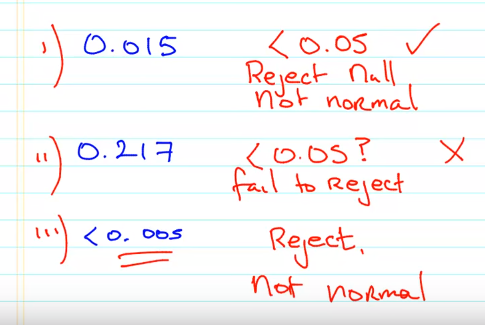
#visulization

#Plotting the sample p-values and drawing a significance line

plot(pvalues, main = "p-values for columns in x (AD-test)", xlab = "Column number in x", ylab = "p-value")

abline(h=0.05, col ="red")





## SHAPIRO-WILK TEST

#Generating 10k points of data and arranging them into 100 columns

x<-rnorm(10000,10,1)

dim(x)<-c(100,100)

#Generating a simple normal quantile-quantile plot for the first column

#Generating a line for the qqplot

qqnorm(x[,1])

qqline (x[,1], col=2)

#Running the A-D test for first column

library(nortest)

#defining a function called "swt" to run the Shapiro-Wilk tests and return p-values

swt<-function(x){ test<-shapiro.test(x); return(test$p.value) }

#store the p-values for each column in a separate variable

swpvalues<-apply(x,MARGIN = 2, swt)

#Plotting the sample p-values and drawing a significance line

plot(swpvalues, main = "p-values for columns in x (Shapiro-Wilk-test)", xlab = "Column number in x", ylab = "p-value")

abline(h=0.05, col ="red")

