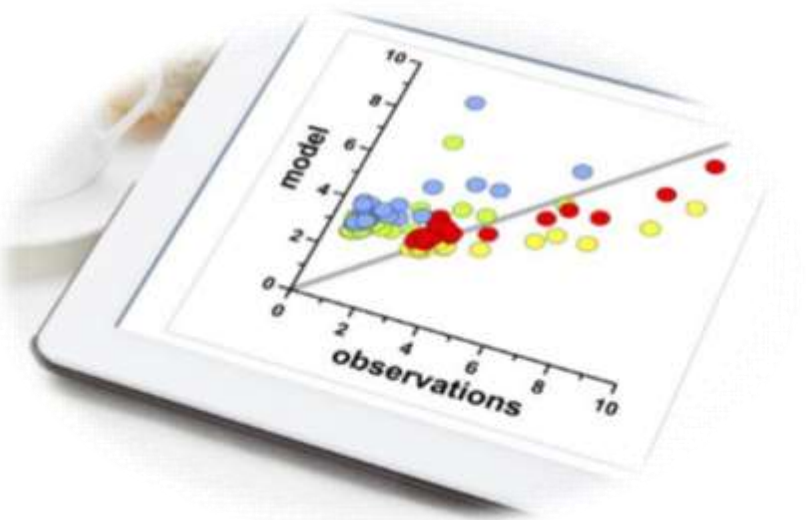
CORRELATION



Correlation is a statistic that measures the degree to which
two variables move in relation to each other. Correlation shows the strength of a relationship between two variables and is expressed numerically by the correlation coefficient. The correlation coefficient's values range between -1.0 and 1.0.

RGui (32-bit)

File Edit Packages Windows Help

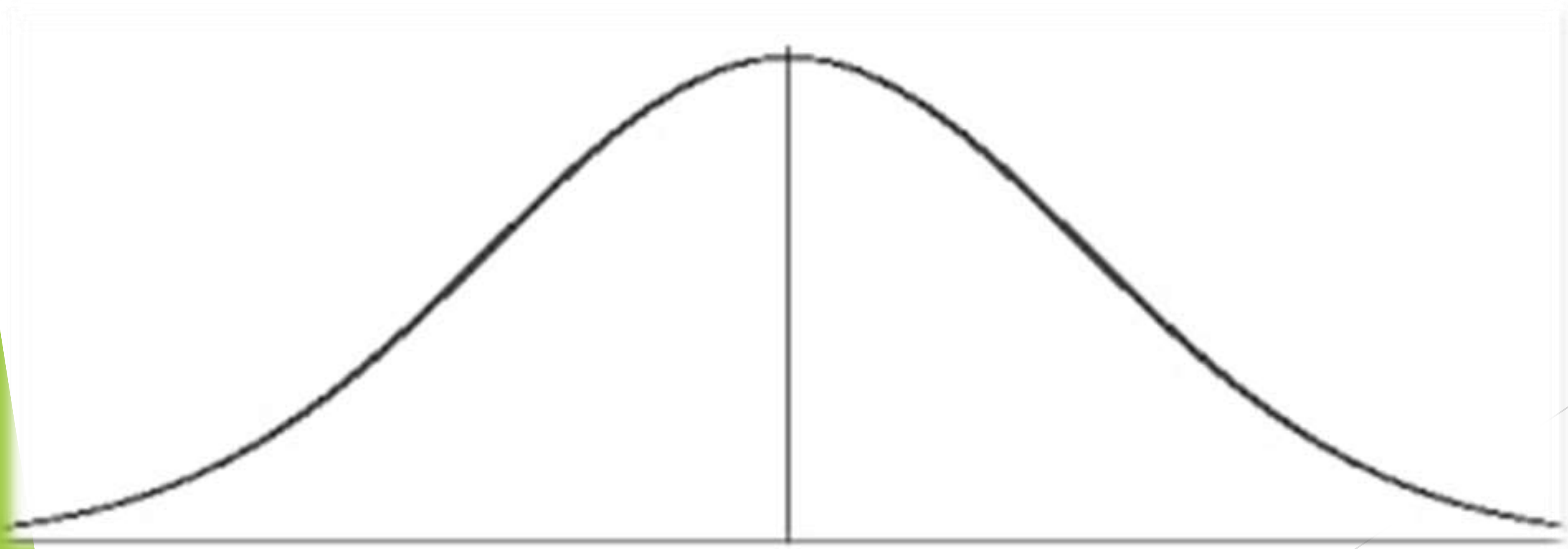
R Console

```
> x=c(41, 19, 23, 40, 55, 57, 33)
> y=c(94, 60, 74, 71, 82, 76, 61)
> cor.test(x,y,method="pearson")#which gives the karlpearson correlation coeffici$

        Pearson's product-moment correlation

data:  x and y
t = 1.434, df = 5, p-value = 0.211
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3592689  0.9192115
sample estimates:
      cor
0.5398442

> x=c(41, 19, 23, 40, 55, 57, 33)
> y=c(94, 60, 74, 71, 82, 76, 61)
> cor.test(x,y,method="spearman")#which gives the spearman rank correlation corre$

        Spearman's rank correlation rho

data:  x and y
S = 14, p-value = 0.06627
alternative hypothesis: true rho is not equal to 0
sample estimates:
 rho
0.75
```

Untitled - R Editor

```
x=c(41, 19, 23, 40, 55, 57, 33)
y=c(94, 60, 74, 71, 82, 76, 61)
cor.test(x,y,method="pearson")#which gives the karlpearson correlation coefficient
x=c(41, 19, 23, 40, 55, 57, 33)
y=c(94, 60, 74, 71, 82, 76, 61)
cor.test(x,y,method="spearman")#which gives the spearman rank correlation correlation coefficien
```

NORMAL DISTRIBUTION



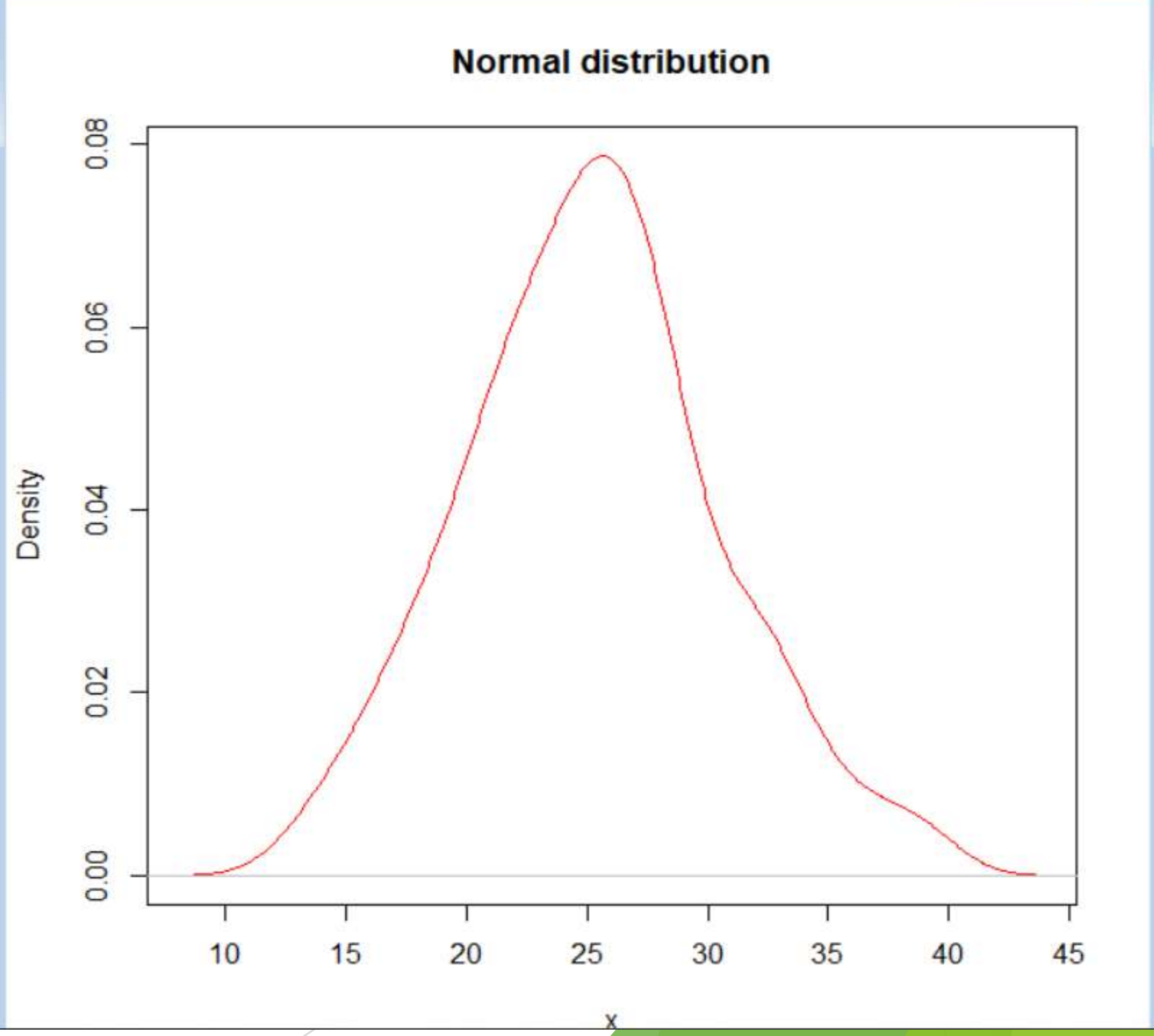
R Console

```
> x=rnorm(100,25,5)
> x
  [1] 20.81912 27.04226 21.83944 28.74034 22.49530 17.64128 26.12654 19.80359
  [9] 32.23006 22.79251 13.15147 24.23004 26.75500 27.42123 23.32643 19.93035
 [17] 23.06309 25.42483 23.51121 19.81128 30.73146 25.96814 17.26478 21.39583
 [25] 20.95771 30.01278 22.46398 21.57285 27.28210 19.42773 19.34907 27.87450
 [33] 25.08025 27.68925 27.93027 22.49029 17.71134 16.58586 19.32949 21.07117
 [41] 24.83860 26.65687 17.05034 28.81423 28.87760 33.07609 36.33339 24.87211
 [49] 38.95004 24.07807 34.19551 24.95435 15.43695 14.63349 29.12675 30.63966
 [57] 25.37408 33.01092 31.58248 25.05012 27.49078 36.15386 27.89803 32.78314
 [65] 29.34952 23.37823 27.28611 27.61882 27.83632 26.67948 22.31832 23.53783
 [73] 25.47758 38.54592 26.14368 19.86300 28.57875 25.27992 32.59681 14.90846
 [81] 23.48397 32.12400 30.18208 22.66758 18.24439 22.64950 21.10928 34.42324
 [89] 25.64916 26.04689 27.77377 24.68229 25.17532 21.85420 31.45554 24.91702
 [97] 24.68167 17.79900 26.23592 20.73721
> plot(density(x),xlab="x",main="Normal distribution",col="red")
>
```

Untitled - R Editor

```
x=rnorm(100,25,5)
x
plot(density(x),xlab="x",main="Normal distribution",col="red")
```

R Graphics: Device 2 (ACTIVE)

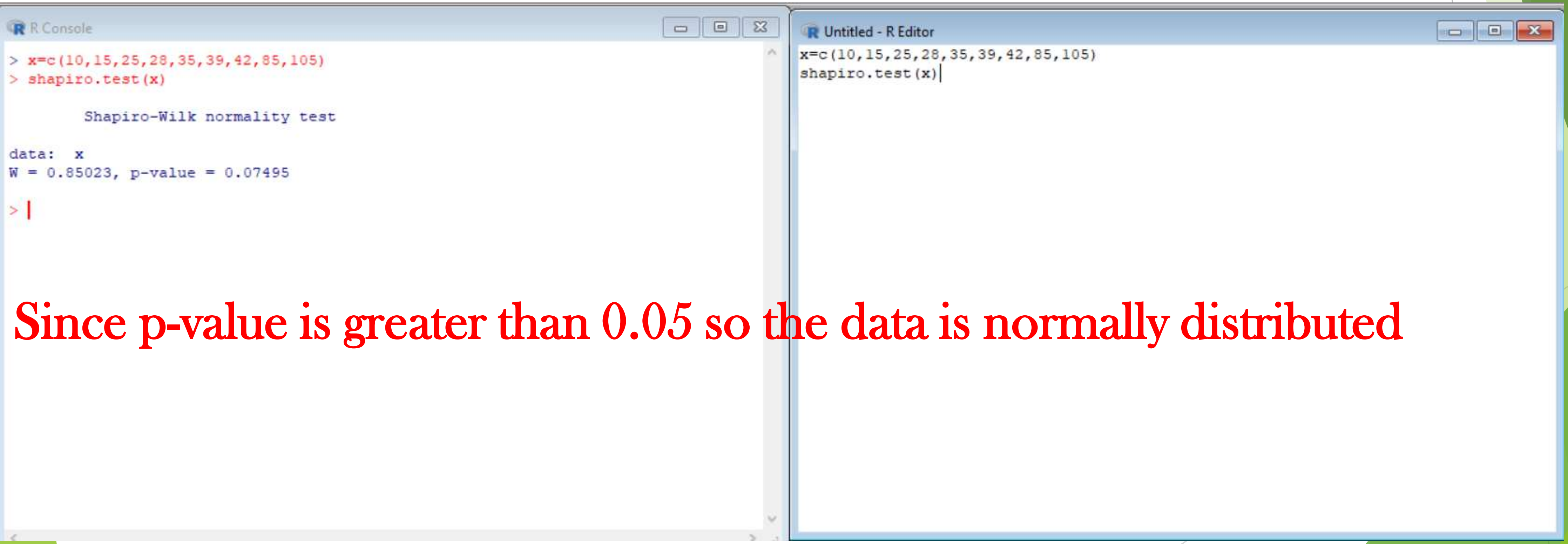


Tests for Normality are:

- Shapiro-Wilk's test : The Shapiro-Wilk test is a way to tell if a random sample comes from a normal distribution. The test gives you a W value; small values indicate your sample is *not* normally distributed (you can reject the null hypothesis that your population is normally distributed if your values are under a certain threshold)
- Normal probability Plot : A normal probability plot is a graphical representation of the data. A normal probability plot is used to check if the given data set is normally distributed or not. It is used to compare a data set with the normal distribution. If a given data set is normally distributed then it will reside in a shape like a straight line.

Shapiro-Wilk's test

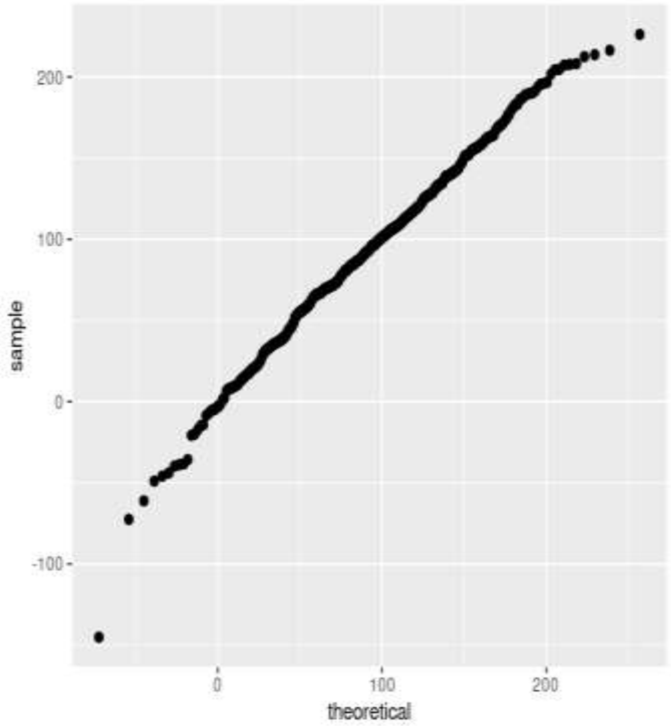
H0:The given data set is Normally distributes
H1:The given data set is not normally distributed



Since p-value is greater than 0.05 so the data is normally distributed

Normal probability Plot

```r
# importing libraries
library(ggplot2)
library(qqplotr)

# creating random data
random_values = rnorm(500, mean = 90, sd = 50)

# ploting data without line and labels
ggplot(mapping = aes(sample = random_values)) + stat_qq_point(size = 2)
```



Plotting different types of distributions

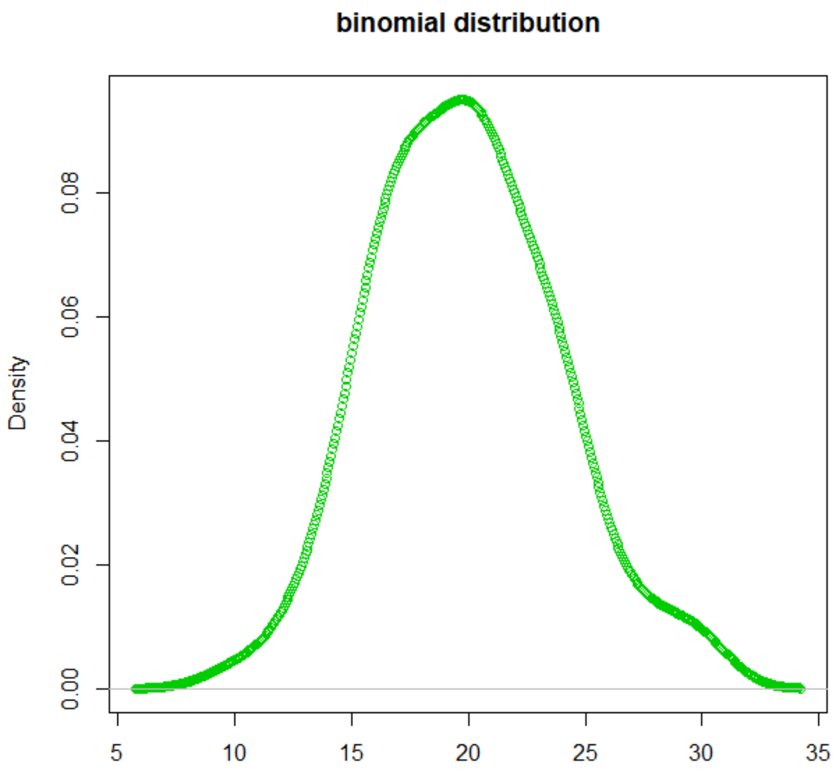
- **Binomial Distribution**

R Console

```
> x=rbinom(100,50,.4)
> plot(density(x),type="b",xlab="x",main="binomial distribution",col="green3")
>
```

Untitled - R Editor

```
x=rbinom(100,50,.4)
plot(density(x),type="b",xlab="x",main="binomial distribution",col="green3")
```



- Poisson Distribution : rpois (n, lambda)
- Exponential Distribution : rexp(n,rate)
- Normal Distribution : rnorm(n,mean,sd)