Survival Analysis

Survival analysis deals with predicting the time when a specific event is going to occur. It is also known as failure time analysis or analysis of time to death. For example predicting the number of days a person with cancer will survive or predicting the time when a mechanical system is going to fail.

The R package named **survival** is used to carry out survival analysis. This package contains the function **Surv()** which takes the input data as a R formula and creates a survival object among the chosen variables for analysis. Then we use the function **survfit()** to create a plot for the analysis.

## Install Package

install.packages("survival")

### Syntax

The basic syntax for creating survival analysis in R is −

Surv(time,event)

survfit(formula)

Following is the description of the parameters used −

* **time** is the follow up time until the event occurs.
* **event** indicates the status of occurrence of the expected event.
* **formula** is the relationship between the predictor variables.

### Example

We will consider the data set named "pbc" present in the survival packages installed above. It describes the survival data points about people affected with primary biliary cirrhosis (PBC) of the liver. Among the many columns present in the data set we are primarily concerned with the fields "time" and "status". Time represents the number of days between registration of the patient and earlier of the event between the patient receiving a liver transplant or death of the patient.

# Load the library.

library("survival")

# Print first few rows.

print(head(pbc))

When we execute the above code, it produces the following result and chart −

id time status trt age sex ascites hepato spiders edema bili chol

1 1 400 2 1 58.76523 f 1 1 1 1.0 14.5 261

2 2 4500 0 1 56.44627 f 0 1 1 0.0 1.1 302

3 3 1012 2 1 70.07255 m 0 0 0 0.5 1.4 176

4 4 1925 2 1 54.74059 f 0 1 1 0.5 1.8 244

5 5 1504 1 2 38.10541 f 0 1 1 0.0 3.4 279

6 6 2503 2 2 66.25873 f 0 1 0 0.0 0.8 248

albumin copper alk.phos ast trig platelet protime stage

1 2.60 156 1718.0 137.95 172 190 12.2 4

2 4.14 54 7394.8 113.52 88 221 10.6 3

3 3.48 210 516.0 96.10 55 151 12.0 4

4 2.54 64 6121.8 60.63 92 183 10.3 4

5 3.53 143 671.0 113.15 72 136 10.9 3

6 3.98 50 944.0 93.00 63 NA 11.0 3

From the above data we are considering time and status for our analysis.

### Applying Surv() and survfit() Function

Now we proceed to apply the **Surv()** function to the above data set and create a plot that will show the trend.

# Load the library.

library("survival")

# Create the survival object.

survfit(Surv(pbc$time,pbc$status == 2)~1)

Surv(pbc$time,pbc$status == 2)~1

Y~1

# Give the chart file a name.

png(file = "survival.png")

# Plot the graph.

plot(survfit(Surv(pbc$time,pbc$status == 2)~1))

# Save the file.

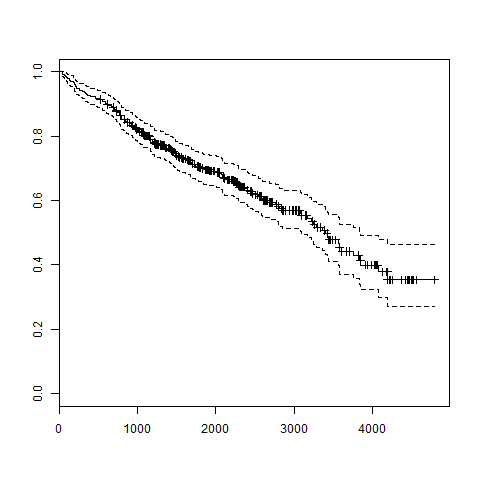
dev.off()

When we execute the above code, it produces the following result and chart −

Call: survfit(formula = Surv(pbc$time, pbc$status == 2) ~ 1)

n events median 0.95LCL 0.95UCL

418 161 3395 3090 3853



# Chi Square Test

**Chi-Square test** is a statistical method to determine if two categorical variables have a significant correlation between them. Both those variables should be from same population and they should be categorical like − Yes/No, Male/Female, Red/Green etc.

For example, we can build a data set with observations on people's ice-cream buying pattern and try to correlate the gender of a person with the flavor of the ice-cream they prefer. If a correlation is found we can plan for appropriate stock of flavors by knowing the number of gender of people visiting.

## Syntax

The function used for performing chi-Square test is **chisq.test()**.

The basic syntax for creating a chi-square test in R is −

chisq.test(data)

Following is the description of the parameters used −

* **data** is the data in form of a table containing the count value of the variables in the observation.

## Example

We will take the Cars93 data in the "MASS" library which represents the sales of different models of car in the year 1993.

library("MASS")

print(str(Cars93))

When we execute the above code, it produces the following result −

'data.frame': 93 obs. of 27 variables:

$ Manufacturer : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4 4 4 5 ...

$ Model : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1 6 24 54 74 73 35 ...

$ Type : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3 2 ...

$ Min.Price : num 12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33 ...

$ Price : num 15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7 ...

$ Max.Price : num 18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36.3 ...

$ MPG.city : int 25 18 20 19 22 22 19 16 19 16 ...

$ MPG.highway : int 31 25 26 26 30 31 28 25 27 25 ...

$ AirBags : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1 2 2 2 2 2 2 ...

$ DriveTrain : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3 2 2 ...

$ Cylinders : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4 4 4 5 ...

$ EngineSize : num 1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...

$ Horsepower : int 140 200 172 172 208 110 170 180 170 200 ...

$ RPM : int 6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 ...

$ Rev.per.mile : int 2890 2335 2280 2535 2545 2565 1570 1320 1690 1510 ...

$ Man.trans.avail : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...

$ Fuel.tank.capacity: num 13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...

$ Passengers : int 5 5 5 6 4 6 6 6 5 6 ...

$ Length : int 177 195 180 193 186 189 200 216 198 206 ...

$ Wheelbase : int 102 115 102 106 109 105 111 116 108 114 ...

$ Width : int 68 71 67 70 69 69 74 78 73 73 ...

$ Turn.circle : int 37 38 37 37 39 41 42 45 41 43 ...

$ Rear.seat.room : num 26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...

$ Luggage.room : int 11 15 14 17 13 16 17 21 14 18 ...

$ Weight : int 2705 3560 3375 3405 3640 2880 3470 4105 3495 3620 ...

$ Origin : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ...

$ Make : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3 5 6 7 9 8 10 ...

The above result shows the dataset has many Factor variables which can be considered as categorical variables. For our model we will consider the variables "AirBags" and "Type". Here we aim to find out any significant correlation between the types of car sold and the type of Air bags it has. If correlation is observed we can estimate which types of cars can sell better with what types of air bags.

# Load the library.

library("MASS")

# Create a data frame from the main data set.

car.data <- data.frame(Cars93$AirBags, Cars93$Type)

# Create a table with the needed variables.

car.data = table(Cars93$AirBags, Cars93$Type)

print(car.data)

# Perform the Chi-Square test.

print(chisq.test(car.data))

When we execute the above code, it produces the following result −

Compact Large Midsize Small Sporty Van

Driver & Passenger 2 4 7 0 3 0

Driver only 9 7 11 5 8 3

None 5 0 4 16 3 6

Pearson's Chi-squared test

data: car.data

X-squared = 33.001, df = 10, p-value = 0.0002723

Warning message:

In chisq.test(car.data) : Chi-squared approximation may be incorrect

### Conclusion

The result shows the p-value of less than 0.05 which indicates a significant correlation.

install.packages("survival")

library(survival)

#Surv(time,event)

#survfit(formula)

pbc

str(pbc)

help(pbc)

View(pbc)

head(pbc)

pbc$time

pbc$status

Surv(pbc$time,pbc$status == 2)

survfit(Surv(pbc$time,pbc$status == 2)~1)

plot(survfit(Surv(pbc$time,pbc$status == 2)~1))