Scilab Manual for Managerial Economics via Scilab by Prof Sreenivasa Charyulu Others Sreenidhi Institute Of Science And Technology[1]

Solutions provided by
Prof Sreenivasa Charyulu
Others
Sreenidhi Institute Of Science And Technology

April 27, 2024

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

Experiment: 1

Price Elasticity Of Demand

Scilab code Solution 1.1 1

```
1 //Price Elasticity Of Demand
2 //OS:Windows 10
3 //Scilab 5.5.2
4
5 clear;
6 clc;
7 close;
8
9 //Concept Information
10 //Formula= % Change in Demand/% Change in Price
11 //Ed = ( Q / P ) * (Po/Qo)
12 // Q is the Change in Quantity [Q0-Q1] (or) Initial
       Quantity - Final Quantity
13 // P is the Change in Price [P0-P1] (or) Initial
      Price - Final Price
14 //Q0 stands for Quantity for previous
15
16 //Type 1
17 //Given
18 P0=8          //in rupees
19 P1=4         //in rupees
```

```
Q0=10      //in units
Q1=12      //in units

//Solution
Ed=((Q0-Q1)/(P0-P1)*(P0/Q0))

//Values
mprintf("    Px(in rupees)      Dx(in units)")
mprintf("\n      %d\t\t\t%d",P0,Q0)
mprintf("\n      %d\t\t\t%d",P1,Q1)

//Result
//Note: -ve sign is ignored
mprintf("\n    Ed:%f",-Ed)
if -Ed < 1 then
    mprintf("\nLess than Elasticity Of Demand")
else
    mprintf("Greater than Elasticity Of Demand")
end

//Type 2
//Given
P0=8        //in rupees
P1=6       //in rupees
Q0=10      //in units
Q1=25      //in units

Ed=((Q0-Q1)/(P0-P1)*(P0/Q0))

//Values
mprintf("\n\n    Px(in rupees)      Dx(in units)")
mprintf("\n      %d\t\t\t%d",P0,Q0)
mprintf("\n      %d\t\t\t%d",P1,Q1)

//Result
//Note: -ve sign is ignored
mprintf("\n    Ed:%d",-Ed)
if -Ed < 1 then
```

```
58      mprintf("\nLess than Elasticity Of Demand")
59  else
60      mprintf("\nGreater than Elasticity Of Demand")
61  end
62
63  //Output
64  //     Px(in rupees)      Dx(in units)
65  //       8                10
66  //       4                12
67  //    Ed:0.400000
68  //Less than Elasticity Of Demand
69  //
70  //    Px(in rupees)      Dx(in units)
71  //       8                10
72  //       6                25
73  //    Ed:6
74  // Greater than Elasticity Of Demand
```

Experiment: 2

Income Elasticity Of Demand

Scilab code Solution 2.2 2

```
1 //Income Elasticity Of Demand
2 //OS:Windows 10
3 //Scilab 5.5.2
4
5 clear;
6 clc;
7 close;
8
9 //Concept Information
10 //Formula= % Change in Income/% Change in Q.D.
11 //Ei = ( Q / Y ) * (Yo/Qo)
12 // Q is the Change in Quantity [Q0-Q1] (or) Initial
       Quantity - Final Quantity
13
14 //Given
15 I0=4000          //in rupees
16 I1=5000         //in rupees
17 Q0=20      //in units
18 Q1=25      //in units
19
20 //Solution
```

```
21 Ei=((Q0-Q1)/(I0-I1)*(I0/Q0))
22
23 //Values
24 mprintf("    Income(in rupees)      Commodity Demand(
      in units)")
25 mprintf("\n      %d\t\t\t%d",I0,Q0)
26 mprintf("\n      %d\t\t\t%d",I1,Q1)
27
28 //Result
29 //Note: -ve sign is ignored
30 mprintf("\n      Ei:%d",Ei)
31 if Ei == 1 then
32     mprintf("\nUnity Elasticity Of Demand")
33 elseif Ei > 1 then
34     mprintf("\nGreater than Elasticity Of Demand")
35 else
36     mprintf("\nLess than Elasticity Of Demand")
37 end
38
39 //Output
40 //     Income(in rupees)      Commodity Demand(in
      units)
41 //      4000          20
42 //      5000          25
43 //      Ei:1
44 // Unity Elasticity Of Demand
```

Experiment: 3

Total Outlay(Expenditure) Method

Scilab code Solution 3.3 3

```
1  //Total Outlay(Expenditure) Method
2  //OS:Windows 10
3  //Scilab 5.5.2
4
5  clear;
6  clc;
7  close;
8
9  //Concept Information
10 //Total Outlay = Price per unit * Demand in units
11
12 price=[8,7,6,5,4]          //per Kg in rupees
13 total_expenditure=[0,7000,12000,15000,16000]
      //in rupees
14
15 mprintf('   Price per kg      Total Expenditure')
16 mprintf('\n    (in rupees)\t      (in rupees)')
17 for i=1:5
18     mprintf('\n       %d\t\t%d',price(i),
```

```
        total_expenditure(i))
end

//Case 1:Price falls from 7/kg to 5/kg
mprintf('\n\n    Price per kg     Total Expenditure')
mprintf('\n    (in rupees)\t      (in rupees)')
mprintf('\n\t%d         \t%d',price(2), total_expenditure(2))
mprintf('\n\t%d         \t%d',price(4), total_expenditure(4))
if total_expenditure(2) > total_expenditure(4) then
    mprintf("\nE > 1")
    mprintf("\nMore than Unity")
    mprintf("\nTotal expenditure increases as a result of decline in price")
else
    mprintf("\nE < 1")
    mprintf("\nLess than Unity")
    mprintf("\nTotal expenditure decreases as a result of increase in price")
end


//Case 2:Price rises from 5/kg to 7/kg
mprintf('\n\n    Price per kg     Total Expenditure')
mprintf('\n    (in rupees)\t      (in rupees)')
mprintf('\n\t%d         \t%d',price(4), total_expenditure(4))
mprintf('\n\t%d         \t%d',price(2), total_expenditure(2))
if total_expenditure(2) > total_expenditure(4) then
    mprintf("\nE > 1")
    mprintf("\nMore than Unity")
    mprintf("\nTotal expenditure increases as a result of decline in price")
else
```

```
47      mprintf("\nE < 1")
48      mprintf("\nLess than Unity")
49      mprintf("\nTotal expenditure decreases as a
           result of increase in price")
50  end
51
52  //Output
53  //     Price per kg      Total Expenditure
54  //     (in rupees)          (in rupees)
55  //        8          0
56  //        7          7000
57  //        6          12000
58  //        5          15000
59  //        4          16000
60  //
61  //     Price per kg      Total Expenditure
62  //     (in rupees)          (in rupees)
63  //   7              7000
64  //   5              15000
65  //E < 1
66  //Less than Unity
67  //Total expenditure decreases as a result of
       increase in price
68  //
69  //     Price per kg      Total Expenditure
70  //     (in rupees)          (in rupees)
71  //   5              15000
72  //   7              7000
73  //E < 1
74  //Less than Unity
75  //Total expenditure decreases as a result of
       increase in price
```

Experiment: 4

Demand Forecast Technique

Scilab code Solution 4.4 4

```
1  //Demand Forecast Technique
2  //Under Quantitative Technique
3  //OS:Windows 10
4  //Scilab 5.5.2
5
6  clear;
7  clc;
8  close;
9
10 //Concept Information
11 //Time Series Analysis - Least Square Method
12 //y = a + b * x
13 //y indicates future sales
14 //x indicates the year number for which forecast is
       to be made
15 //a is the fixed element of overall sales which is
       not affected by time change
16 //b indicates the rate of change of sales with
       change in time
17 //a =  y /N     b =  xy / n ^2
18
```

```
19 //Given
20 years = [2012,2013,2014,2015,2016]
21 sales = [120,130,135,142,138]      //in rupees
22
23 //Consider the year 2014
24 //Solution
25 X_val=[]
26 for i=1:5
27     X_val($+1) = years(i)-years(3)
28 end
29
30 XY_val=[]
31 for i=1:5
32     XY_val($+1) = X_val(i) * sales(i)
33 end
34
35 X_sqr=X_val^2
36
37 sigma_y = sum(sales)
38 sigma_x = sum(X_val)
39 sigma_xy = sum(XY_val)
40 sigma_xsqr = sum(X_sqr)
41
42 N=length(sales)
43 a = sigma_y/N
44 b = sigma_xy/sigma_xsqr
45
46 //Result
47 //Table
48 mprintf("Year              ")
49 for i=1:5
50     mprintf("    %d",years(i))
51 end
52 mprintf("\n Sales(in rupees)")
53 for i=1:5
54     mprintf("     %d",sales(i))
55 end
56
```

```
57 mprintf("\n\n Year\t  Sales(Y)   X     XY       X^2\
   n")
58 for i=1:5
59     mprintf(" %d\t   %d\t     %d\t   %d\t     %d\n",
       years(i),sales(i),X_val(i),XY_val(i),X_sqr(i)
       )
60 end
61
62 mprintf("\n\t  y=%d\t   x =%d    xy =%d    x ^2=%d\n",
   sigma_y,sigma_x,sigma_xy,sigma_xsqr)
63
64 mprintf("\n a = %d",a)
65 mprintf("\n b = %.2f",b)
66
67 //Future Sales Prediction
68 //For 2017
69 sigma_x = 2017 - 2014 //expected year - current case
       year
70 sigma_y = a + b * sigma_x
71 mprintf("\n\n Estimate : %.2f",sigma_y)
72
73 //For 2018
74 sigma_x = 2018 - 2014 //expected year - current case
       year
75 sigma_y = a + b * sigma_x
76 mprintf("\n Estimate : %.2f",sigma_y)
77
78 //Output
79 // Year                2012   2013   2014   2015
   2016
80 // Sales(in rupees)    120    130    135    142
   138
81 //
82 // Year   Sales(Y)   X     XY       X^2
83 // 2012    120       -2    -240      4
84 // 2013    130       -1    -130      1
85 // 2014    135       0     0         0
86 // 2015    142       1     142       1
```

```
87 // 2016     138         2       276         4
88 //
89 //    y =665      x =0     xy =48     x ^2=10
90 //
91 // a = 133
92 // b = 4.80
93 //
94 // Estimate : 147.40
95 // Estimate : 152.20
```

Experiment: 5

Break Even Point

Scilab code Solution 5.5 5

```
1  //Break Even Point
2  //OS:Windows 10
3  //Scilab 5.5.2
4
5  clear;
6  clc;
7  close;
8
9  //Given
10 fixed_cost = 500          //in rupees
11 variable_cost = 100       //in rupees per unit
12 selling_price = 200       //in rupees per unit
13 units = []
14 total_variable_cost = []
15 total_cost = []
16 total_revenue = []
17
18 //Solution
19 for i = 0:10
20     units($+1) = i
21     total_variable_cost($+1) = i * variable_cost
```

```
22      total_cost($+1) = total_variable_cost(i+1) +
           fixed_cost
23      total_revenue($+1) = units(i+1) * selling_price
24  end
25
26  for i = 1:10
27      if total_cost(i) == total_revenue(i) then
28          BEP_units = i
29      else
30          i = i + 1
31      end
32  end
33
34  //Result
35  mprintf("\n Units  FC(in Rs)  VC(per unit)  TVC
       TC  SP(per unit)  TR")
36  for i = 0:10
37      mprintf("\n    %d\t  %d\t      %d\t %d\t %d\t%d\t
            %d",units(i+1),fixed_cost,variable_cost,
          total_variable_cost(i+1),total_cost(i+1),
          selling_price,total_revenue(i+1))
38  end
39
40  mprintf("\nBreak Even Point occurs at %d units",
       BEP_units-1)
41
42  //Output
43  //
44  // Units  FC(in Rs)  VC(per unit)  TVC     TC  SP(
       per unit)  TR
45  //    0     500          100       0    500     200     0
46  //    1     500          100      100      600     200
       200
47  //    2     500          100      200      700     200
       400
48  //    3     500          100      300      800     200
       600
49  //    4     500          100      400      900     200
```

```
      800
50 //    5      500          100       500       1000     200
      1000
51 //    6      500          100       600       1100     200
      1200
52 //    7      500          100       700       1200     200
      1400
53 //    8      500          100       800       1300     200
      1600
54 //    9      500          100       900       1400     200
      1800
55 //    10     500          100       1000      1500     200
      2000
56 //Break Even Point occurs at 5 units
```

Experiment: 6

Capital Budgeting

Scilab code Solution 6.6 6

```
1 //Capital Budgeting
2 //Pay Back Period
3 //OS:Windows 10
4 //Scilab 5.5.2
5
6 clear;
7 clc;
8 close;
9
10 //Concept Information
11 //Even Model
12 //PBP = Original Cost of the project / Annual Cash
     Inflow
13
14 //Case 1
15 //Solution
16 originalcost = 500000
17 annualcashinflow = 100000
18 pbp = originalcost / annualcashinflow
19
20 //Result
```

```
21 mprintf("Pay Back Period for the project is : %d
      years",pbp)
22
23 //Case 2
24 originalcost = input("\n Enter the original cost of
      the project:")
25 annualcashinflow = input("\n Enter the annual cash
      inflow:")
26 pbp = originalcost / annualcashinflow
27
28 //Result
29 mprintf("\n Pay Back Period for the project is : %.2
      f years",pbp)
30
31 //Output
32 // Pay Back Period for the project is : 5 years
33 //\n Enter the original cost of the project:200000
34 //
35 //\n Enter the annual cash inflow:200000
36 //
37 //
38 // Pay Back Period for the project is : 1.00 years
```

Experiment: 7

Capital Budgeting

Scilab code Solution 7.7 7

```
1 //Capital Budgeting
2 //Payback Period - Project Selection
3 //OS:Windows 10
4 //Scilab 5.5.2
5
6 clear;
7 clc;
8 close;
9
10 //Given
11 project_cost = 200000          //in rupees
12 projecta_cashinflow =
      [50000,50000,50000,50000,50000]      //in years
      1-5
13 projectb_cashinflow =
      [100000,100000,100000,100000,100000]      //in
      years 1-5
14 rcA = 0       //recovering period for Project A
15 rcB = 0       //recovering period for Project B
16
17 //Solution
```

```
18 projecta_cumulative_cashinflow = [0,0,0,0,0]
19 projecta_cumulative_cashinflow(1) =
      projecta_cashinflow(1)              //since 1st
      value of cumulation is always 1st value of
      cashinflow
20 projectb_cumulative_cashinflow = [0,0,0,0,0]
21 projectb_cumulative_cashinflow(1) =
      projectb_cashinflow(1)              //since 1st
      value of cumulation is always 1st value of
      cashinflow
22 for i=2:5
23     projecta_cumulative_cashinflow(i) =
          projecta_cumulative_cashinflow(i-1) +
          projecta_cashinflow(i)
24     projectb_cumulative_cashinflow(i) =
          projectb_cumulative_cashinflow(i-1) +
          projectb_cashinflow(i)
25 end
26
27 //Result
28 mprintf("      Project A")
29 mprintf("\n Year  Cash inflow  Cumulative cash
      inflow")
30 for i=1:5
31     mprintf("\n  %d        %d              %d",i,
          projecta_cashinflow(i),
          projecta_cumulative_cashinflow(i))
32 end
33
34 mprintf("\n\n      Project B")
35 mprintf("\n Year  Cash inflow  Cumulative cash
      inflow")
36 for i=1:5
37     mprintf("\n  %d        %d              %d",i,
          projectb_cashinflow(i),
          projectb_cumulative_cashinflow(i))
38 end
39
```

```
40 for i=1:5
41     if(projecta_cumulative_cashinflow(i) ==
          project_cost)
42         mprintf("\n\nRecovering period is %d years
             in Project A",i)
43         rcA = i
44     end
45 end
46
47 for i=1:5
48     if(projectb_cumulative_cashinflow(i) ==
          project_cost)
49         mprintf("\nRecovering period is %d years in
             Project B",i)
50         rcB = i
51     end
52 end
53
54 if rcA < rcB then
55     mprintf("Project A is accepted")
56 else
57     mprintf("\nProject B is accepted")
58 end
59
60 //Output
61 //      Project A
62 // Year  Cash inflow  Cumulative cash inflow
63 //  1        50000              50000
64 //  2        50000              100000
65 //  3        50000              150000
66 //  4        50000              200000
67 //  5        50000              250000
68 //
69 //      Project B
70 // Year  Cash inflow  Cumulative cash inflow
71 //  1        100000              100000
72 //  2        100000              200000
73 //  3        100000              300000
```

```
74 //  4        100000              400000
75 //  5        100000              500000
76 //
77 //Recovering period is 4 years in Project A
78 //Recovering period is 2 years in Project B
79 //Project B is accepted
```

Experiment: 8

Profitability Index

Scilab code Solution 8.8 8

```
1  //Profitability Index
2  //OS:Windows 10
3  //Scilab 5.5.2
4
5  clear;
6  clc;
7  close;
8
9  //Given
10 projectA_investment = 30000      //in rupees
11 projectB_investment = 50000      //in rupees
12 estimatedlife_A = 5         //in years
13 estimatedlife_B = 5         //in years
14 scrapvalue_A = 2000     //in rupees
15 scrapvalue_B = 4000     //in rupees
16 pv_10 = [0.909,0.826,0.751,0.683,0.621]        //
      discount rate of 10%
17 projectA_CF = [10000,15000,8000,6000,4000]
18 projectB_CF = [40000,30000,10000,5000,4000]
19 pvcf_A = []         //present value
20 pvcf_B = []         //present value
```

```
21
22 //Solution
23 //Profitability Index for Project A
24 for i=1:5
25     pvcf_A($+1) = pv_10(i) * projectA_CF(i)
26 end
27 total_pvcf = sum(pvcf_A)
28 PI_A = total_pvcf / projectA_investment
29
30 //Profitability Index for Project B
31 for i=1:5
32     pvcf_B($+1) = pv_10(i) * projectB_CF(i)
33 end
34 total_pvcf = sum(pvcf_B)
35 PI_B = total_pvcf / projectB_investment
36
37 //Result
38 mprintf("\n  Particulars\t    Project A\t  Project B
     ")
39 mprintf("\n\n  Initial Investment   %d        %d",
     projectA_investment,projectB_investment)
40 mprintf("\n  (in rupees)")
41 mprintf("\n\n  Estimated life")
42 mprintf("          %d\t      %d",estimatedlife_A,
     estimatedlife_B)
43 mprintf("\n  (in years)")
44 mprintf("\n\n  Scrap value")
45 mprintf("          %d\t    %d",scrapvalue_A,
     scrapvalue_B)
46 mprintf("\n  (in rupees)\n")
47 mprintf("\n  Year\tProject A CF\tProject B CF")
48 for i=1:5
49     mprintf("\n   %d\t   %d      \t   %d",i,
         projectA_CF(i),projectB_CF(i))
50 end
51 mprintf("\n\n  Project A")
52 mprintf("\n  Year\tCash inflows(CF)      PV@10percent
     \tPVCF")
```

```
for i=1:5
    mprintf("\n    %d\t    %d    \t          %.3f\t
                %d",i,projectA_CF(i),pv_10(i),pvcf_A(i
        ))
end
mprintf("\n\n  Project B")
mprintf("\n  Year\tCash inflows(CF)      PV@10percent
    \tPVCF")
for i=1:5
    mprintf("\n    %d\t    %d    \t          %.3f\t
                %d",i,projectB_CF(i),pv_10(i),pvcf_B(i
        ))
end
mprintf("\n\n  Profitability Index for Project A: %
    .2f",PI_A)
mprintf("\n  Profitability Index for Project B: %.2f
    ",PI_B)
if PI_A > PI_B then
    mprintf("\n  Project A proposal is accepted
        because its profatibility index is more")
else
    mprintf("\n  Project B proposal is accepted
        because its profatibility index is more")
end

//Output
//
//  Particulars      Project A      Project B
//
//  Initial Investment    30000          50000
//  (in rupees)
//
//  Estimated life          5            5
//  (in years)
//
//  Scrap value           2000         4000
//  (in rupees)
//
```

```
82 //   Year     Project A CF     Project B CF
83 //    1       10000            40000
84 //    2       15000            30000
85 //    3       8000             10000
86 //    4       6000             5000
87 //    5       4000             4000
88 //
89 //   Project A
90 //   Year     Cash inflows(CF)      PV@10percent    PVCF
91 //    1       10000                 0.909           9090
92 //    2       15000                 0.826           12390
93 //    3       8000                  0.751           6008
94 //    4       6000                  0.683           4098
95 //    5       4000                  0.621           2484
96 //
97 //   Project B
98 //   Year     Cash inflows(CF)      PV@10percent    PVCF
99 //    1       40000                 0.909           36360
100 //   2       30000                 0.826           24780
101 //   3       10000                 0.751           7510
102 //   4       5000                  0.683           3415
103 //   5       4000                  0.621           2484
104 //
105 //   Profitability Index for Project A: 1.14
106 //   Profitability Index for Project B: 1.49
107 //   Project B proposal is accepted because its
    profatibility index is more
```

Experiment: 9

Average rate of return

Scilab code Solution 9.9 9

```
1 //Average rate of return
2 //OS:Windows 10
3 //Scilab 5.5.2
4
5 clear;
6 clc;
7 close;
8
9 //Concept Information
10 //ARR = (Average annual profits after taxes /
     Average Investment original investment) * 100
11 //Average investment = 1/2 (Total investment - scrap
     value)
12
13 //Given
14 projectA_investment = 500000        //in rupees
15 projectA_scrapvalue = 20000         //in rupees
16 profit_A = [40000,60000,70000,50000,20000]    //in
     rupees
17
18 //Solution
```

```
total_earnings = sum(profit_A)
N = length(profit_A)
average_earnings = total_earnings / N
average_investment = 1/2 * (projectA_investment -
   projectA_scrapvalue) + projectA_scrapvalue
average_rateofreturn = (average_earnings /
   average_investment) * 100

//Result
mprintf("\n  Total Earnings: %d rupees",
   total_earnings)
mprintf("\n  Average Earnings: %d rupees",
   average_earnings)
mprintf("\n  Average Investment: %d rupees",
   average_investment)
mprintf("\n  Average Rate of Return: %.2f percent",
   average_rateofreturn)

//Output
//
//  Total Earnings: 240000 rupees
//  Average Earnings: 48000 rupees
//  Average Investment: 260000 rupees
//  Average Rate of Return: 18.46 percent
```

Experiment: 10

Capital Budgeting

Scilab code Solution 10.10 10

```
1 //Capital Budgeting
2 //Net Present Value - Project Selection
3 //OS:Windows 10
4 //Scilab 5.5.2
5
6 clear;
7 clc;
8 close;
9
10 //Given
11 projectA_investment = 60000      //in rupees
12 projectB_investment = 80000      //in rupees
13 estimatedlife_A = 7         //in years
14 estimatedlife_B = 7         //in years
15 scrapvalue_A = 9000     //in rupees
16 scrapvalue_B = 11000     //in rupees
17 pv_10 = [0.909,0.826,0.751,0.683,0.621]        //
      discount rate of 10%
18 projectA_CF = [20000,19000,18000,12500,9000]
19 projectB_CF = [68000,42000,15000,14000,11000]
20 projectA_CF($+1) = scrapvalue_A  //last value is
```

```
    always scrap value
21 projectB_CF($+1) = scrapvalue_B  //last value is
    always scrap value
22 pvcf_A = []        //present value
23 pvcf_B = []        //present value
24
25 //Solution
26 //Project A Present Value
27 for i=1:5
28     pvcf_A($+1) = pv_10(i) * projectA_CF(i)
29 end
30 total_pvcf = sum(pvcf_A)
31 NPV_A = total_pvcf - projectA_investment
32
33 //Project B Present Value
34 for i=1:5
35     pvcf_B($+1) = pv_10(i) * projectB_CF(i)
36 end
37 total_pvcf = sum(pvcf_B)
38 NPV_B = total_pvcf - projectB_investment
39
40 //Result
41 mprintf("\n  Particulars\t    Project A\t  Project B
    ")
42 mprintf("\n\n  Initial Investment   %d        %d",
    projectA_investment,projectB_investment)
43 mprintf("\n  (in rupees)")
44 mprintf("\n\n  Estimated life")
45 mprintf("          %d\t      %d",estimatedlife_A,
    estimatedlife_B)
46 mprintf("\n  (in years)")
47 mprintf("\n\n  Scrap value")
48 mprintf("          %d\t    %d",scrapvalue_A,
    scrapvalue_B)
49 mprintf("\n  (in rupees)\n")
50 mprintf("\n  Year\tProject A CF\tProject B CF")
51 for i=1:5
52     mprintf("\n   %d\t   %d      \t   %d",i,
```

```
        projectA_CF(i),projectB_CF(i))
end
mprintf("\n\n  Project A")
mprintf("\n  Year\tCash inflows(CF)      PV@10percent
   \tPVCF")
for i=1:5
    mprintf("\n   %d\t   %d   \t         %.3f\t
             %d",i,projectA_CF(i),pv_10(i),pvcf_A(i
        ))
end
mprintf("\n\n  Project B")
mprintf("\n  Year\tCash inflows(CF)      PV@10percent
   \tPVCF")
for i=1:5
    mprintf("\n   %d\t   %d   \t         %.3f\t
             %d",i,projectB_CF(i),pv_10(i),pvcf_B(i
        ))
end
mprintf("\n\n   NPV for Project A: %.2f",NPV_A)
mprintf("\n   NPV for Project B: %d",NPV_B)

if NPV_A > NPV_B then
    mprintf("\n   Project A is accepted as its NPV
       is higher")
else
    mprintf("\n   Project B is accepted as its NPV
       is higher")
end

//Output
//    Particulars          Project A      Project B
//
//    Initial Investment    60000          80000
//    (in rupees)
//
//    Estimated life          7            7
//    (in years)
//
```

```
82 //    Scrap value          9000      11000
83 //    (in rupees)
84 //
85 //    Year    Project A CF    Project B CF
86 //     1     20000            68000
87 //     2     19000            42000
88 //     3     18000            15000
89 //     4     12500            14000
90 //     5     9000             11000
91 //
92 //    Project A
93 //    Year    Cash inflows(CF)      PV@10percent    PVCF
94 //     1     20000                  0.909           18180
95 //     2     19000                  0.826           15694
96 //     3     18000                  0.751           13518
97 //     4     12500                  0.683           8537
98 //     5     9000                   0.621           5589
99 //
100 //    Project B
101 //    Year    Cash inflows(CF)      PV@10percent    PVCF
102 //     1     68000                  0.909           61812
103 //     2     42000                  0.826           34692
104 //     3     15000                  0.751           11265
105 //     4     14000                  0.683           9562
106 //     5     11000                  0.621           6831
107 //
108 //     NPV for Project A: 1518.50
109 //     NPV for Project B: 44162
110 //     Project B is accepted as its NPV is higher
```