NSUT

Netaji Subhas University of Technology

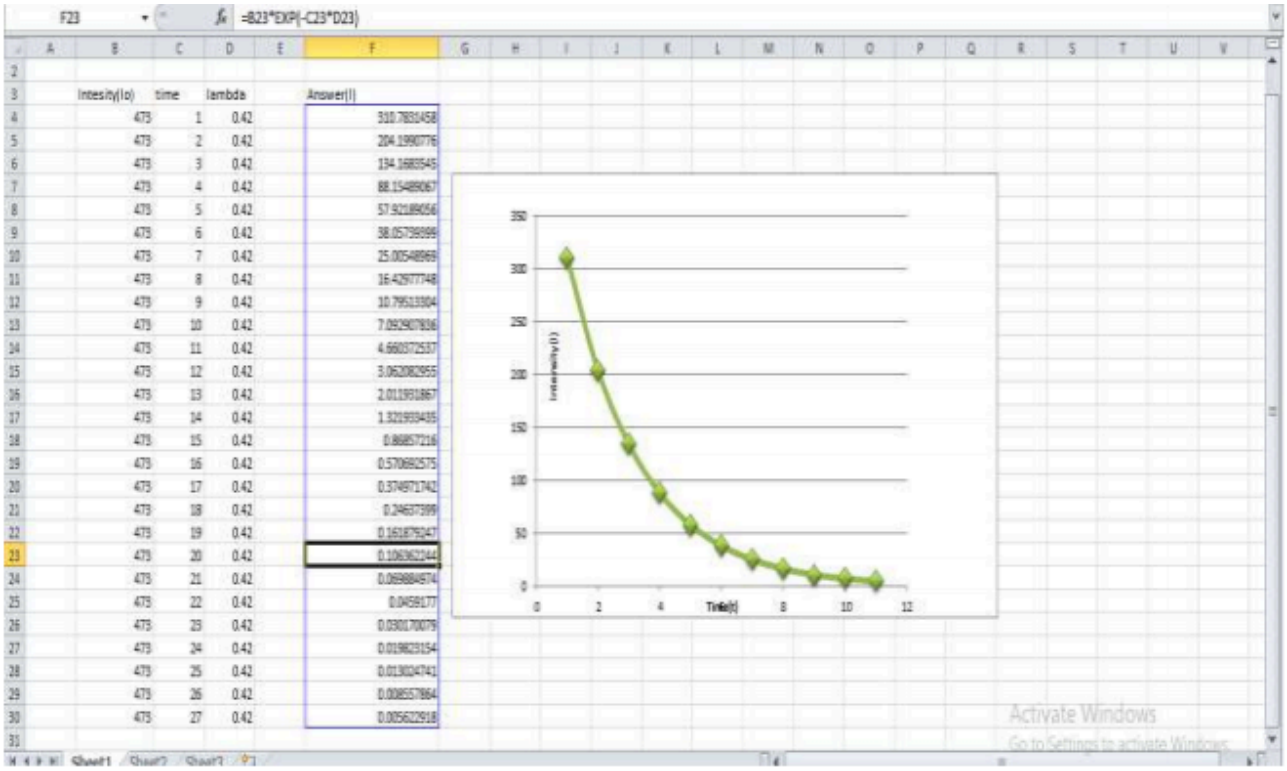
Modeling and Simulation

NAME: Shiv Kumar
CLASS : IT-2
ROLL NO – 2016UIT2563

INDEX

Q1. Simulate exponential decay using MS EXCEL?

$$I = I_o e^{(-t/\tau)}$$



Q2 Simulate CHI SQUARE

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
int ob_freq[2][2];
cin>>ob_freq[0][0];
cin>>ob_freq[0][1];
cin>>ob_freq[1][0];
cin>>ob_freq[1][1];
int row_sum[2];
int col_sum[2];
row_sum[0] = ob_freq[0][0] + ob_freq[0][1];
row_sum[1] = ob_freq[1][0] + ob_freq[1][1];
col_sum[0] = ob_freq[0][0] + ob_freq[1][0];
col_sum[1] = ob_freq[0][1] + ob_freq[1][1];
int total = ob_freq[0][0] + ob_freq[0][1] + ob_freq[1][0] + ob_freq[1][1];
int ex_freq[2][2];
for(int i=0;i<2;i++)
{
for(int j=0;j<2;j++)
{
ex_freq[i][j] = row_sum[i]*col_sum[j]/total;
}

}
double ob_chi_Sq = 0;
for(int i=0;i<2;i++)
for(int j=0;j<2;j++)
ob_chi_Sq += (double)(ex_freq[i][j] - ob_freq[i][j])*(ex_freq[i][j] - ob_freq[i][j])/ex_freq[i][j];
cout<<ob_chi_Sq<<endl;
cout<<"Enter tabulated value of chi_sq:"<<endl;
double ex_chi_Sq;
cin>> ex_chi_Sq;
if(ob_chi_Sq < ex_chi_Sq)
cout<< "Accepted"<<endl;
else
cout << "Rejected"<<endl;
return 0;
}
```

Q3 Simulate Kolmoghrov – Smirnov Test

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
vector<float>random;
random.push_back(0.0f);
cout<<"10 Random numbers are: "<<endl;
srand(9);
for(int i=0;i<10;i++){
random.push_back(rand()%100);
}
sort(random.begin(),random.end());
for(auto &x:random){
x/=100;
cout<<x<<" ";
}
cout<<endl<<endl<<endl;
float N=10,mxdp=-1,mxdm=-1;
cout<<"i: "<<"\t";
for(int i=1;i<=random.size()-1;i++) cout<<i<<"\t";
cout<<endl;
cout<<"R: "<<"\t";
for(int i=1;i<=random.size()-1;i++) cout<<random[i]<<"\t";
cout<<endl;
cout<<"i/N: "<<"\t";
for(int i=1;i<=random.size()-1;i++) cout<<i/N<<"\t";

cout<<endl;
cout<<"D+: "<<"\t";
for(int i=1;i<=random.size()-1;i++){
cout<<max(i/N - random[i],0.0f)<<"\t";
mxdp=max(mxdp,max(i/N - random[i],0.0f));
}
cout<<endl;
cout<<"D-: "<<"\t";
for(int i=1;i<=random.size()-1;i++){
cout<<max(random[i] - (i-1)/N ,0.0f)<<"\t";
mxdm=max(mxdm,max(random[i] - (i-1)/N ,0.0f));
}
cout<<endl<<endl<<endl;
float mxd=max(mxdm,mxdp);
cout<<"D = "<<max(mxdm,mxdp)<<endl;
cout<<"Enter Given D_aplha: ";
float d;
cin>>d;
if(d>mxd)
cout<<"Reject the hypothesis!"<<endl;
else
cout<<"Accept the Hypothesis!!"<<endl;
return 0;
}
```

Q4. Value of PIE using Monte Carlo

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
const int nrolls = 100000; // number of experiments
const int nstars = 95; // maximum number of stars to distribute
default_random_engine generator;
uniform_int_distribution<int> x(0,1000),y(0,1000);
int p[10]={};
float ni=0;
for(int i=0;i<nrolls;i++){
float _x=x(generator);
float _y=y(generator);
_x/=1000.0f;
_y/=1000.0f;
float d= _x*_x + _y*_y;
//cout<<d<<endl;
if(d<=1) ni += 1;
float pi= 4*ni/(i+1);
if(i%1000 == 0) cout<<pi<<endl;
}
return 0;
}
```

Q5 Generate random number using Mid Square Method:

```
#include<bits/stdc++.h>
using namespace std;
#define MOD 1000000009
int findLength(long long int n){
int cnt=0;
while(n){
cnt++;
n/=10;
}
return cnt;
}
long long int crop(long long int seed,int l,int n){
//cout<<seed<<"^^^^^^^"<<endl;
while(l--){
seed/=10;
}
long long int temp=0;
int g=1;
//cout<<seed<<"######"<<n<<endl;
for(int i=0;i<n;i++){
g*=10;
}
temp=seed%g;
int len=findLength(temp);
int q=n-len;
while(q--){

temp*=10;
if(len==0){
temp+=1;
len=1;
}
}
return temp;
}
vector<long long int> midSqaure(long long int seed,int n)
{
vector<long long int>v;
while(n--){
int l=findLength(seed);
seed=(seed*seed)%MOD;
int b=findLength(seed);
int t=(b-l+1)/2;
seed=crop(seed,t,l);
v.push_back(seed);
}
return v;
}
int main()
{
cout<<"Enter A Number: "<<endl;
long long int seed;
cin>>seed;
cout<<"Enter the number of random numbers you want"<<endl;
int n;
```

```
cin>>n;
vector<long long int>v=midSqaure(seed,n);
for(int i=0;i<v.size();i++){
cout<<v[i]<<" ";
}
return 0;
}
```

Q6. Generate random number using Linear Congruential method(Residue method)

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
long long int x,c,m,a;
cin>>x>>c>>a>>m;
int t;
cin>>t;
t++;
while(t--){
cout<<x%m<<endl;
x=((a*x)+c)%m;
}
return 0;
}
```

Q7 Simulation of single server queuing system

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
int seed1,seed2;
vector<int>arrival,service;
cout<<"Enter 2 Seed Values: ";
cin>>seed1>>seed2;
cout<<"=========================================================================
==========
==="<<endl<<endl;;
cout<<"Arrival times of patient: "<<endl;
srand(seed1);
arrival.push_back(0);
service.push_back(0);
for(int i=0;i<10;i++){
arrival.push_back(arrival[i] + rand()%16);
}
srand(seed2);
for(int i=0;i<10;i++){
service.push_back(rand()%10);
if(service[service.size()-1] == 0) service[service.size()-1] ++ ;
}
for(int i=1;i<=10;i++) cout<<arrival[i]<<"\t";
cout<<endl;
cout<<"Service time for each patient: "<<endl;

for(int i=1;i<=10;i++) cout<<service[i]<<"\t";
cout<<endl;
int time=0;
int cnt=1;
vector<int>st,en;
st.push_back(arrival[1]);
while(cnt<=10){
if(time<arrival[cnt]){
time+=1;
continue;
}
service[cnt]-=1;
time+=1;
if(service[cnt] == 0){
en.push_back(time);
cnt+=1;
if(cnt<=10) st.push_back(max(time,arrival[cnt]));
continue;
}
}
cout<<"Starting of the service: "<<endl;
for(auto &x: st) cout<<x<<"\t";
cout<<endl;
cout<<"Ending of service: "<<endl;
for(auto &x: en) cout<<x<<"\t";
cout<<endl;
cout<<"Waiting by each Patient: "<<endl;
for(int i=1;i<=10;i++){
cout<<max(0,st[i-1]-arrival[i])<<"\t";
```

```
}
return 0;
}
```