```
% Geostats: 09/16/2021
clear all;
D = load('elevations.txt');

D = D(:); % creates one big long column
Ix = find(isfinite(D)) %isfinits is putting zeros in a new vector where there are non-
```

```
Ix = 12121×1
     8
     9
    10
    11
    12
    13
    14
    15
    16
    17
     ⋮
```

```
                          % and find is finding the values where there are ones and not

D = D(Ix)
```

```
D = 12121×1
10^3 ×
    2.8803
    2.8788
    2.8774
    2.8759
    2.8738
    2.8657
    2.8576
    2.8496
    2.8436
    2.8402
     ⋮
```

```
     % and find is finding the values where there are ones and not zeros
```

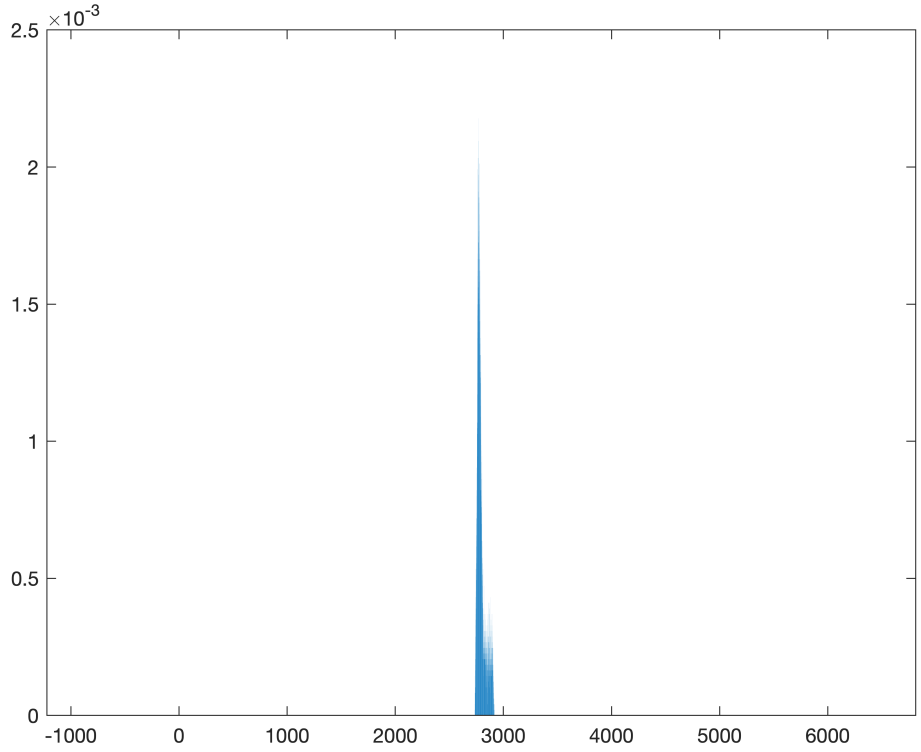
PROBLEM 1

```
D_min = min(D); %min, max, std, and mean find the minimum maximum, standard deviation,
D_max = max(D);
D_std = std(D);
D_mean= mean(D);
nbins = 100; % set number of bins
x0 = (D_mean-100*D_std):D_std/100:(D_mean+100*D_std); %sort the values from least to g
```

PROBLEM 2

```
figure(1)

[distr, xbins]= hist(D, x0); %creates histogram
dx = D_std/10; %bin width
```

```
distr = distr / sum(distr*dx); %length of bins
bar(xbins, distr) %plot the relative  density histrogram problem
```



PROBLEM 3

```
RD = randsample(D,10,true) % draws 10 random numbers from D, the true removes the numb
```

```
RD = 10×1
10^3 ×
    2.7720
    2.7697
    2.7734
    2.7740
    2.9059
    2.7625
    2.8928
    2.8047
    2.8257
    2.8042
```

```
RD_mean = mean(RD)
```

```
RD_mean = 2.8085e+03
```

```
RD_min= min(RD)
```

```
RD_min = 2.7625e+03
```

```
RD_max= max(RD)
```

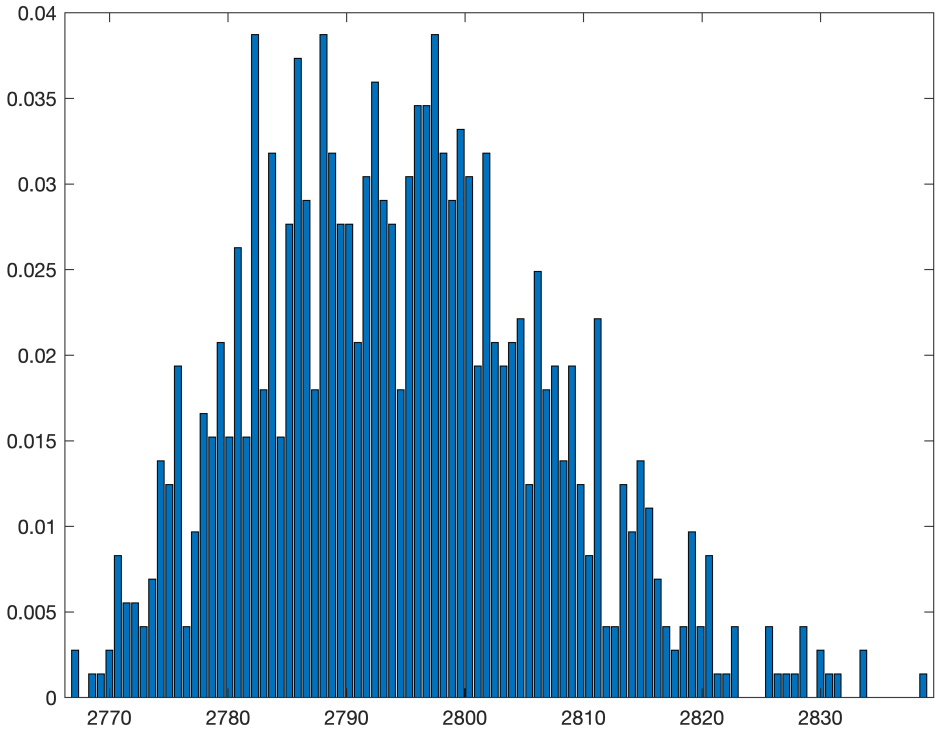
```
RD_max = 2.9059e+03
```

PROBLEM 4

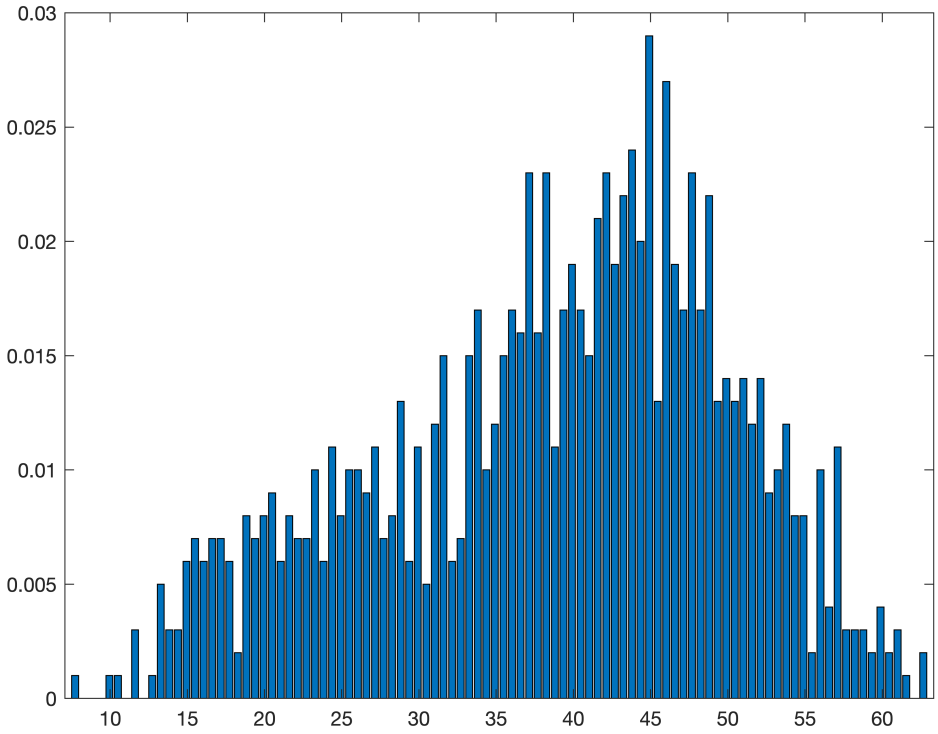
```
for i=1:1000; %index into list the length of 1000 with 1000 samples

   RD = randsample(D,10,true);
   RD_mean(i)= mean(RD);
   RD_min(i) = min(RD);
   RD_max(i) = max(RD);
   RD_std(i) = std(RD);

end % the for loop stores the values for each index (I) and creates arrays for the min
```

PROBLEM 5

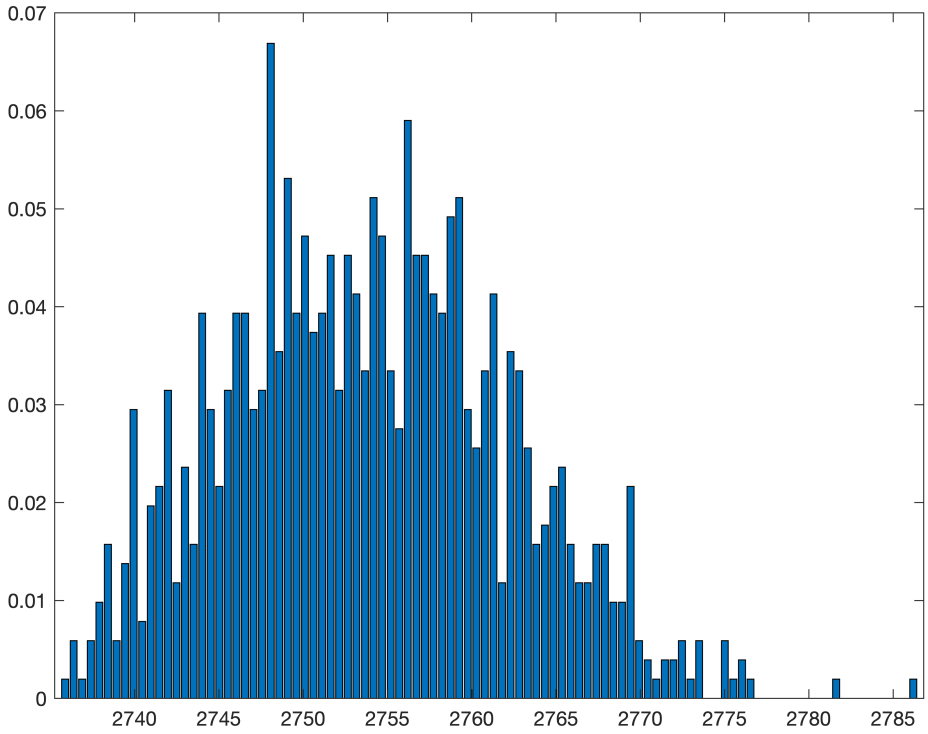
```
figure(2)
[distr_mean, xbins_mean]= hist(RD_mean, nbins); % these are the sane commands as expla
dx_mean = xbins_mean(2)-xbins_mean(1);
distr_mean = distr_mean / (sum(distr_mean.*dx_mean));
bar(xbins_mean, distr_mean) %plot the relative  density histrogram problem
```



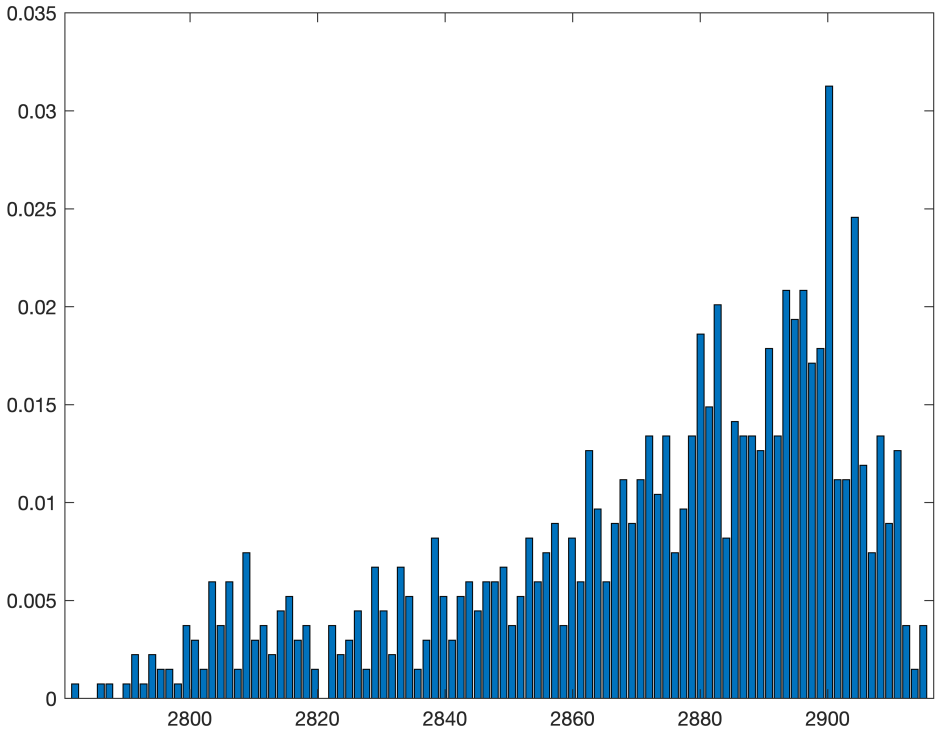

```
figure(3)
[distr_std, xbins_std]= hist(RD_std, nbins);
dx_std = xbins_std(2)-xbins_std(1);
distr_std = distr_std / sum(distr_std);
bar(xbins_std, distr_std) %plot the relative  density histrogram problem
```




```
figure(4)
[distr_min, xbins_min]= hist(RD_min, nbins);
dx_min = xbins_min(2)-xbins_min(1);
distr_min = distr_min / sum(distr_min.*dx_min);
bar(xbins_min, distr_min) %plot the relative  density histrogram problem
```




```
figure(5)
[distr_max, xbins_max]= hist(RD_max, nbins);
dx_max = xbins_max(2)-xbins_max(1);
distr_max = distr_max / sum(distr_max.*dx_max);
bar(xbins_max, distr_max) %plot the relative  density histrogram problem
```



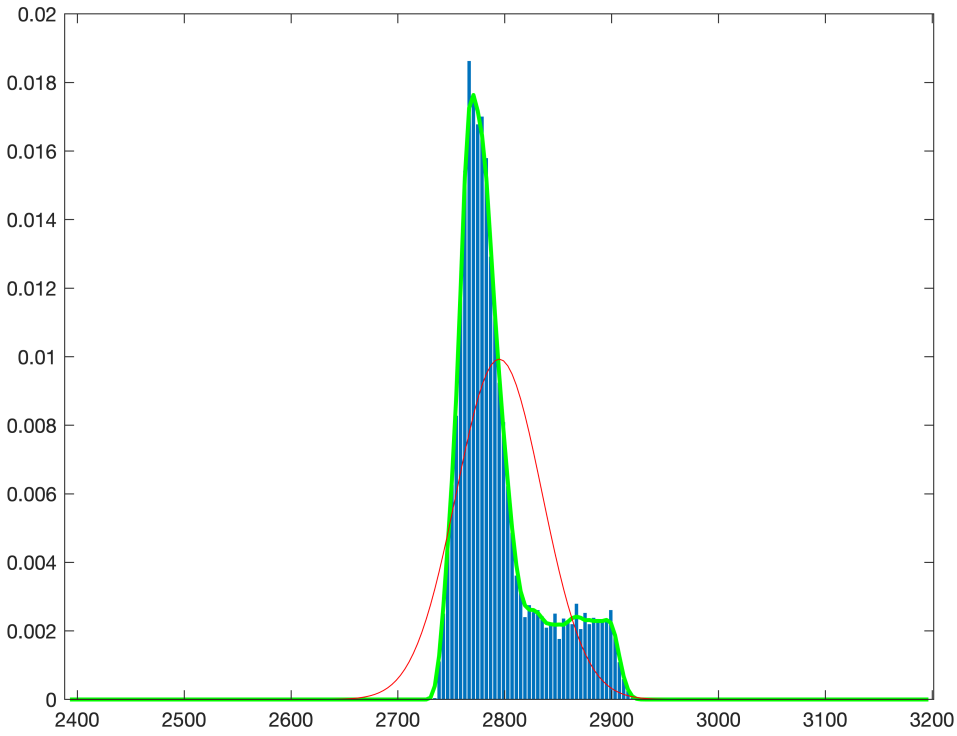
PROBLEM 6

```
ND = load('elevations.txt'); % load elevations.txt data
ND = ND(:); %make it one long column
zx = std(D)/10; %set bin width
z0 = (mean(D)-std(D)*10):zx:(mean(D)+std(D)*10) %create bin spacing
```

```
z0 = 1×201
10^3 ×
    2.3931    2.3971    2.4011    2.4051    2.4091    2.4131    2.4172    2.4212 ···
```

```
[N, xbins]= hist(D,z0); %standard histogram fucntion like before
RDH = N/sum(N*zx);
figure(6);
bar(z0,RDH)
h = 10; %window size
clear f;
for n=1:length(z0)
    dist = (D-z0(n)); %distance from x0 to all other data values
    Idx = find(abs(dist)<h); %finding all data points within h from x0
    w =15/16*(1-(dist(Idx)/h).^2).^2; %weights of all points
    f(n) =sum(w); %sum the weights

end

zx = std(D)/10;
f= 1/sum(f*zx)*f;%normalized pdf
gauss = mypdf(z0,D_mean, D_std);
```

```
figure(6); hold on
 plot(z0, f, 'g', 'linewidth', 2)
plot(z0, gauss, 'r')
```



PROBLEM 7

for ther sample mean

```
figure(7); hold on
bar(xbins_mean, distr_mean)%For mean distribution values
mu_0 = std(RD_mean)/10;
RD_0 = (mean(RD_mean)-std(RD_mean)*10):mu_0:(mean(RD_mean)+std(RD_mean)*10)
```

```
RD_0 = 1×201
10^3 ×
    2.6710    2.6723    2.6735    2.6747    2.6760    2.6772    2.6784    2.6797 ···
```

```
RDmu = mean(RD_mean)
```

```
RDmu = 2.7948e+03
```

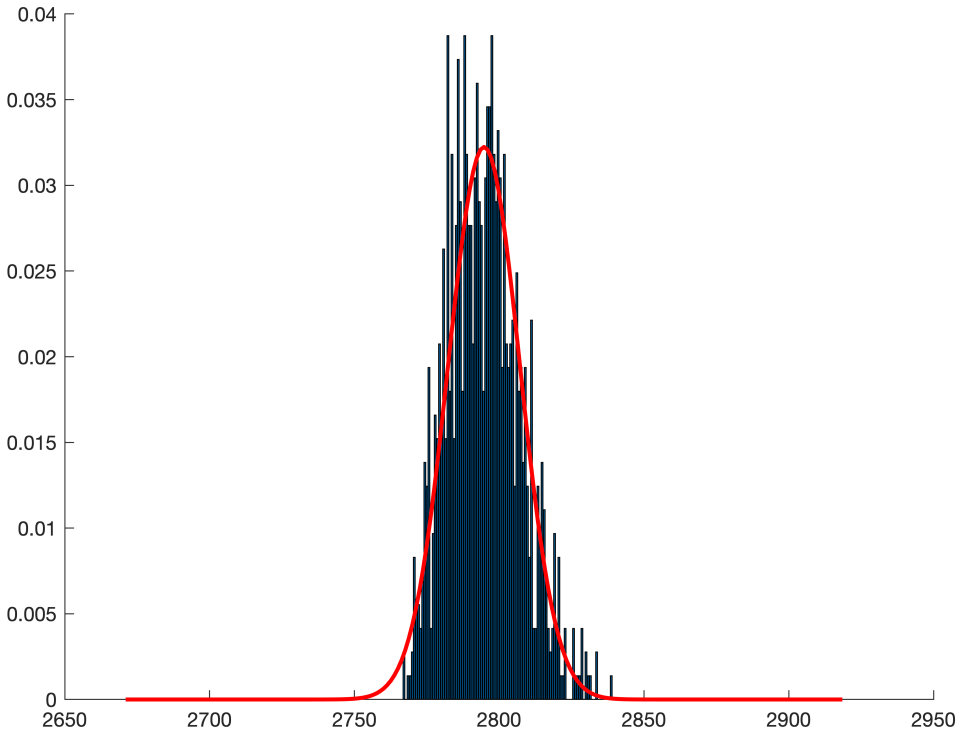
```
RDsig = std(RD_mean)
```

```
RDsig = 12.3741
```

```
mu_g = mypdf(RD_0, RDmu, RDsig) % use mypdf function
```

```
mu_g = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

```
plot(RD_0, mu_g, 'r', 'linewidth', 2) % plot norm distrabution curve
```



for the sample standard deviation

```
figure(8); hold on
bar(xbins_std, distr_std)
std_0 =std(RD_std)/10;
RDstd_0 = (mean(RD_std)-std(RD_std)*10):std_0:(mean(RD_std)+std(RD_std)*10)
```

```
RDstd_0 = 1×201
  -74.8965  -73.7610  -72.6255  -71.4900  -70.3545  -69.2189  -68.0834  -66.9479 ···
```

```
STDmu = mean(RD_std)
```

```
STDmu = 38.6548
```

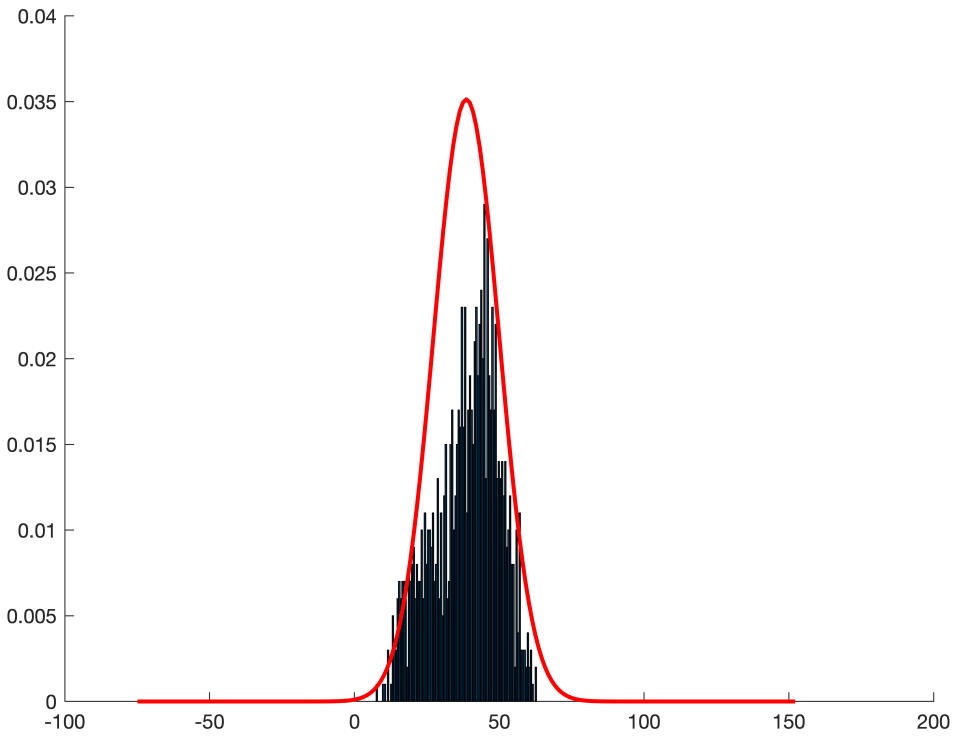
```
STDsig = std(RD_std)
```

```
STDsig = 11.3551
```

```
STD_g = mypdf(RDstd_0, STDmu, STDsig)
```

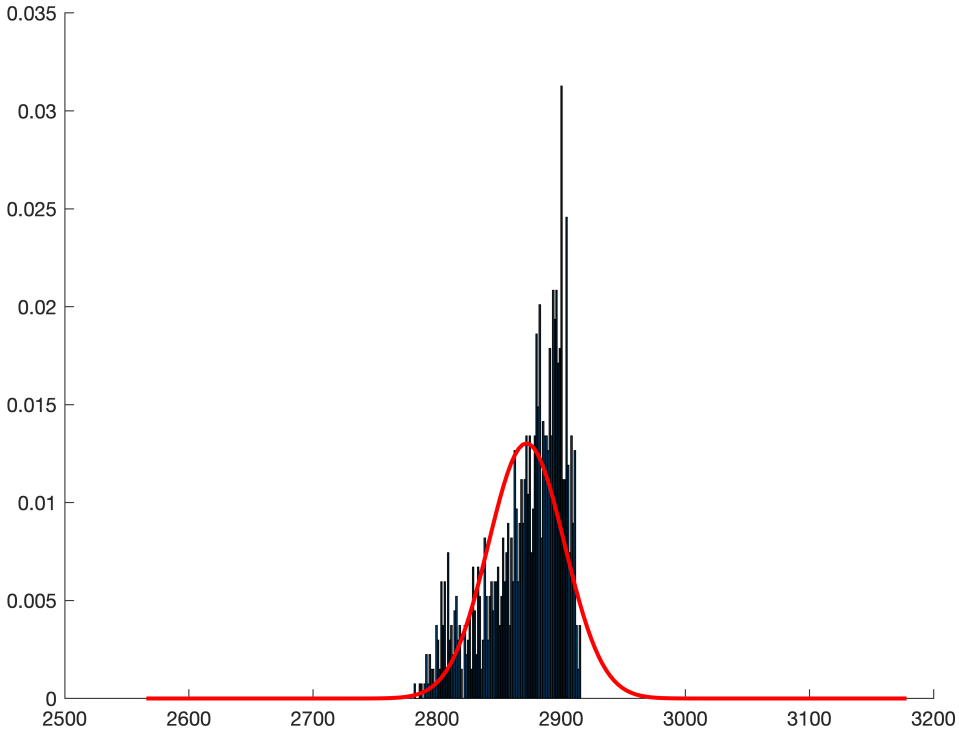
```
STD_g = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

```
plot(RDstd_0, STD_g, 'r', 'linewidth', 2)
```



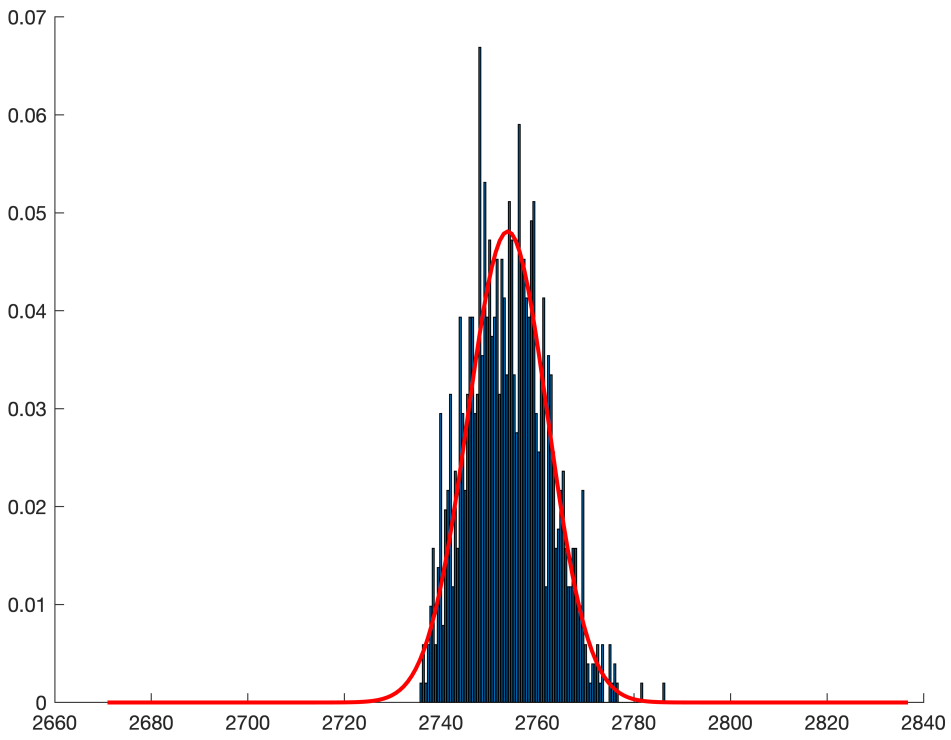
for the sample max

```
clear f;
figure(9); hold on
bar(xbins_max, distr_max);
max0 =std(RD_max)/10;
Max_0 = (mean(RD_max)-std(RD_max)*10):max0:(mean(RD_max)+std(RD_max)*10);
Maxmu = mean(RD_max);
Maxsig = std(RD_max);
max_g = mypdf(Max_0, Maxmu, Maxsig);
plot(Max_0, max_g, 'r', 'linewidth', 2)
```



for sample min

```
clear f;
figure(10); hold on
bar(xbins_min, distr_min);
max0 =std(RD_min)/10;
Max_0 = (mean(RD_min)-std(RD_min)*10):max0:(mean(RD_min)+std(RD_min)*10);
Maxmu = mean(RD_min);
Maxsig = std(RD_min);
max_g = mypdf(Max_0, Maxmu, Maxsig);
plot(Max_0, max_g, 'r', 'linewidth', 2)
```



Problem 8

```
IDmu = find(RD_mean<=D_mean)
```

```
IDmu = 1×508
     8    10    11    14    15    16    18    19    22    23    25    27    28 ···
```

```
perc_mu = length(IDmu)/length(RD_mean)
```

```
perc_mu = 0.5080
```

```
perc_mu = perc_mu*100
```

```
perc_mu = 50.8000
```

```
fprintf( '%f chance of finding a value less than the true mean', perc_mu)
```

```
50.800000 chance of finding a value less than the true mean
```

Problem 9

```
tru_max = find(RD_max>(0.99*D_max) & RD_max<(1.01*D_max))
```

```
tru_max = 1×353
     4     7     9    11    12    15    20    23    25    26    28    31    32 ···
```

```
tru_min = find(RD_min>(0.99*D_min) & RD_min<(1.01*D_min))
```

```
tru_min = 1×860
     2     3     4     5     8     9    10    11    12    13    14    15    16 ···
```

```
%range = tru_max(min(tru_max):max(tru_max)):tru_min(min(tru_min):max(tru_min):2)
perc_max = length(tru_max)/length(RD_max)
```

```
perc_max = 0.3530
```

```
perc_min  = length(tru_min)/length(RD_min)
```

```
perc_min = 0.8600
```

```
%Should this be one probabilty
```

Problem 10

```
Elev_Range = find(RD_mean >(D_mean - RDsig) & RD_mean < (D_mean + RDsig))
```

```
Elev_Range = 1×678
     1     2     3     4     8     9    11    12    14    16    17    19    21 ···
```

```
Prob = RD_mean(Elev_Range)
```

```
Prob = 1×678
10^3 ×
    2.8013    2.8043    2.8049    2.8036    2.7934    2.8034    2.7887    2.7987 ···
```

```
Prob_min = min(Prob)
```

```
Prob_min = 2.7824e+03
```

```
Prob_max = max(Prob)
```

```
Prob_max = 2.8071e+03
```

Problem 11

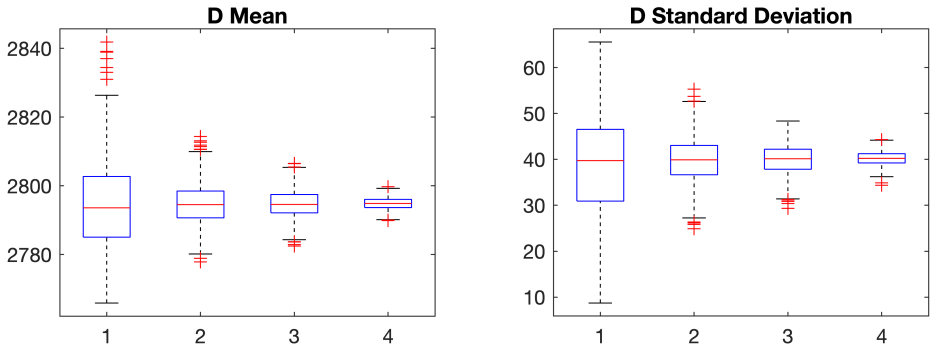
```
clear D;
D = load('elevations.txt');

D = D(:); % creates one big long column
Ix = find(isfinite(D)) %isfinits is putting zeros in a new vector where there are non-
```

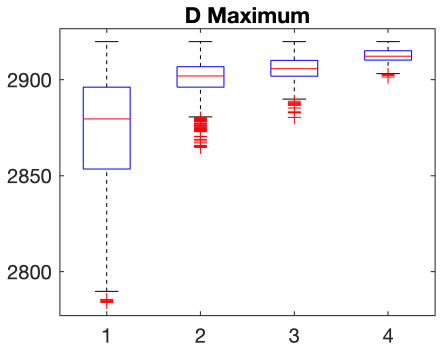
```
Ix = 12121×1
     8
     9
    10
    11
    12
    13
    14
    15
    16
    17
```

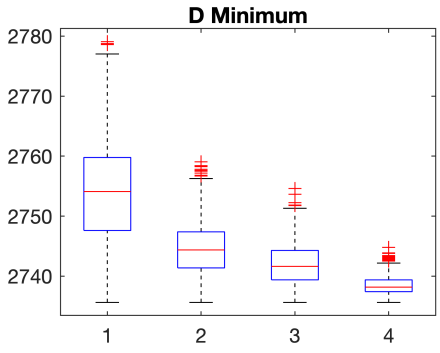
```
                        % and find is finding the values where there are ones and not z

D = D(Ix)
```

```
D = 12121×1
10^3 ×
    2.8803
    2.8788
    2.8774
    2.8759
    2.8738
    2.8657
    2.8576
    2.8496
    2.8436
    2.8402
     ⋮
```

```
samplesize = [10, 50, 100, 500];
for q= 1:length(samplesize)
    for n = 1:1000
      D2 = randsample(D, samplesize(q), true);
      Dmean(n,q) = mean(D2); % mean of D2
      Dmin(n,q) = min(D2) ;
      Dstd(n,q) = std(D2);
      Dmax(n,q) = max(D2) ;
    end
end
figure(11)
subplot(2,2,1)
boxplot(Dmean)
title('D Mean')
subplot(2,2,2)
boxplot(Dstd)
title('D Standard Deviation')
subplot(2,2,3)
boxplot(Dmax)
title('D Maximum')
subplot(2,2,4)
boxplot(Dmin)
title('D Minimum')
```







Problem 12 Uniform Sampling

```
% clear D,Ix, D2, x0, D_std, D_mu, D_std, Didx_mu_200;

D = load('elevations.txt')
```

```
D = 110×111
10^3 ×
       NaN    2.8961    2.8966    2.8995    2.9029    2.9056    2.9102    2.9023 ···
       NaN    2.8896    2.8996    2.8978    2.9059    2.9122    2.9041    2.9064
       NaN    2.8868    2.9041    2.8959    2.9006    2.9027    2.9044    2.9040
       NaN    2.8868    2.8919    2.8941    2.8975    2.9009    2.9057    2.9041
       NaN    2.8933    2.8861    2.8908    2.8950    2.8991    2.9060    2.9037
       NaN    2.8808    2.8932    2.8880    2.8933    2.8953    2.9014    2.9043
       NaN    2.8815    2.8844    2.9014    2.9091    2.8937    2.8956    2.9073
    2.8803    2.8792    2.8830    2.8916    2.8985    2.8888    2.8967    2.9025
    2.8788    2.8704    2.8735    2.8844    2.8878    2.8961    2.8886    2.8991
    2.8774    2.8797    2.8730    2.8757    2.8848    2.8905    2.8885    2.8993
     ⋮
```

```
nx = 22
```

```
nx = 22
```

```
nr = 111
```

```
nr = 111
```

```
nc = 110
```

```
nc = 110
```

```
Ix=1:nx:nc
```

```
Ix = 1×5
     1    23    45    67    89
```

```
Iy = 1:nx:nr
```

```
Iy = 1×6
     1    23    45    67    89   111
```

```
% this time we leave D as a matrix (110x111)
%plot the realtive density histogram
D_idx = zeros(size(D));
%index D with a vector that describes the rows and clolumns we want
% Ix = find(isfinite(D)) %isfinits is putting zeros in a new vector where there are no
                         % and find is finding the values where there are ones and not

for x = 1:1:22  %This creates the values to be stored in rows
    for y= 1:1:22 %This stores the y-axis values
    Iy1 = y:nx:nr;
    Ix1 = x:nx:nc;
    D_idx = D(Ix1, Iy1); %create the array that stores row and column vectors from Ix1
    D_idx = D_idx(:); %this makes D_idx a column
    Didx_mu_200(x,y) = nanmean(D_idx);
    Didx_min_200(x,y) = nanmin(D_idx(:));
    Didx_max_200(x,y) = nanmax(D_idx(:));
    Didx_std_200(x,y) = nanstd(D_idx(:));
    end
end
%For the standard deviation off of mean
Didx_mu_200c = Didx_mu_200(:);  %same process as before when creating normal distribut
D_muc = mean(Didx_mu_200c);
D_std = std(Didx_mu_200c);
zx =(D_muc-10*D_std):D_std/10:(D_muc+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.7626    2.7629    2.7632    2.7636    2.7639    2.7642    2.7645    2.7649 ···
```

```
gauss_c = mypdf(zx, D_muc, D_std);
%Make Plots
figure(12);
subplot(2,2,1)
histogram(Didx_mu_200c, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Mean')

%for maximum
clear zx, gauss_c, D_std;
```

```
gauss_c = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

```
Didx_max_200c = Didx_max_200(:);
D_mux = mean(Didx_max_200c);
D_std = std(Didx_max_200c);
zx =(D_mux-10*D_std):D_std/10:(D_mux+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.8040    2.8049    2.8059    2.8068    2.8078    2.8087    2.8096    2.8106 ···
```

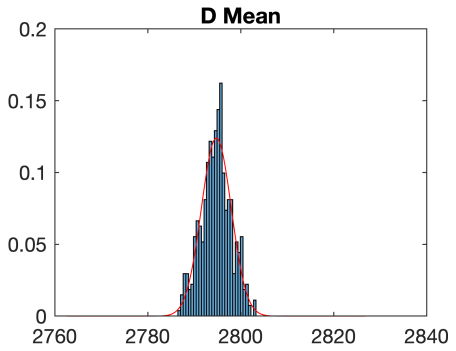
```
gauss_c = mypdf(zx, D_mux, D_std);

subplot(2,2,2)
histogram(Didx_max_200c, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Max')

%for minimum
clear zx, gauss_c, D_std;
```

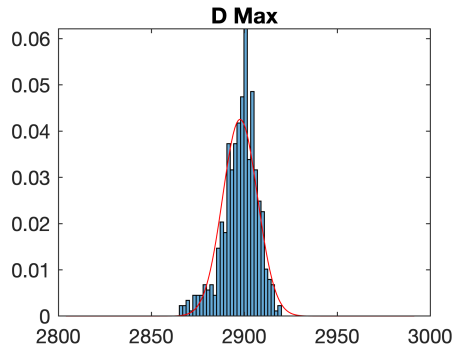
```
gauss_c = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

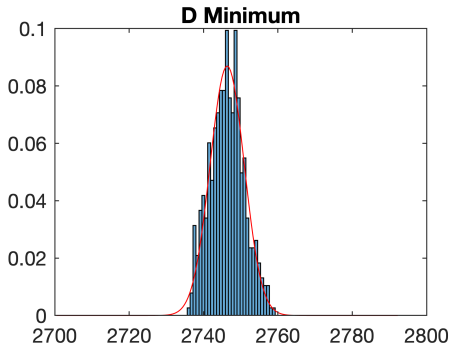
```
Didx_min_200c = Didx_min_200(:);
D_mui = mean(Didx_min_200c);
D_std = std(Didx_min_200c);
zx =(D_mui-10*D_std):D_std/10:(D_mui+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.7005    2.7009    2.7014    2.7019    2.7023    2.7028    2.7032    2.7037 ···
```

```
gauss_c = mypdf(zx, D_mui, D_std);
%Make plots
subplot(2,2,3)
histogram(Didx_min_200c, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Minimum')
```







Problem 13

```
figure(12)
clear D,D2, x0, D_std, Ix1, Iy1, D_idx, Ix, Iy ;
```

```
D2 = 500×1
10^3 ×
    2.8030
    2.7947
    2.7629
    2.7699
    2.8488
    2.8660
    2.7581
    2.7859
    2.7754
    2.7608
     ⋮
x0 = 1×20001
10^3 ×
   -1.2222   -1.2218   -1.2214   -1.2210   -1.2206   -1.2202   -1.2198   -1.2194 ···
D_std = 4.5881
Ix1 = 1×5
    22    44    66    88   110
Iy1 = 1×5
    22    44    66    88   110
D_idx = 25×1
```

```
10^3 ×
    2.8266
    2.7744
    2.7858
    2.8532
    2.9069
    2.7775
    2.7696
    2.7652
    2.8345
    2.8992
     ⋮
Ix = 1×5
     1    23    45    67    89
```

```
D = load('elevations.txt')
```

```
D = 110×111
10^3 ×
       NaN    2.8961    2.8966    2.8995    2.9029    2.9056    2.9102    2.9023 ···
       NaN    2.8896    2.8996    2.8978    2.9059    2.9122    2.9041    2.9064
       NaN    2.8868    2.9041    2.8959    2.9006    2.9027    2.9044    2.9040
       NaN    2.8868    2.8919    2.8941    2.8975    2.9009    2.9057    2.9041
       NaN    2.8933    2.8861    2.8908    2.8950    2.8991    2.9060    2.9037
       NaN    2.8808    2.8932    2.8880    2.8933    2.8953    2.9014    2.9043
       NaN    2.8815    2.8844    2.9014    2.9091    2.8937    2.8956    2.9073
    2.8803    2.8792    2.8830    2.8916    2.8985    2.8888    2.8967    2.9025
    2.8788    2.8704    2.8735    2.8844    2.8878    2.8961    2.8886    2.8991
    2.8774    2.8797    2.8730    2.8757    2.8848    2.8905    2.8885    2.8993
     ⋮
```

```
nx = 3
```

```
nx = 3
```

```
nr = 111
```

```
nr = 111
```

```
nc = 110
```

```
nc = 110
```

```
Ix=1:nx:nc
```

```
Ix = 1×37
     1     4     7    10    13    16    19    22    25    28    31    34    37 ···
```

```
Iy = 1:nx:nr
```

```
Iy = 1×37
     1     4     7    10    13    16    19    22    25    28    31    34    37 ···
```

```
% this time we leave D as a matrix (110x111)
%plot the realtive density histogram
%index D with a vector that describes the rows and clolumns we want
 %isfinits is putting zeros in a new vector where there are non-finite values
                         % and find is finding the values where there are ones and not
```

```
for x = 1:1:3
    for y= 1:1:3
    Iy1 = y:nx:nr;
    Ix1 = x:nx:nc;
    D_idx = D(Ix1, Iy1);
%     D_idx = D_idx(isfinite(D_idx));
%     D_idx = D_idx(:);
    Didx_mu_30(x,y) = nanmean(D_idx(:));
    Didx_min_30(x,y) = nanmin(D_idx(:));
    Didx_max_30(x,y) = nanmax(D_idx(:));
    Didx_std_30(x,y) = nanstd(D_idx(:));
    end
end
%For the standard deviation off of mean
clear D_muc, D_std, D_mux, D_mui, zx, gauss_c;
```

```
D_std = 4.5881
D_mux = 2.8977e+03
D_mui = 2.7464e+03
zx = 1×201
10^3 ×
    2.7005    2.7009    2.7014    2.7019    2.7023    2.7028    2.7032    2.7037 ···
```

```
Didx_mu_30q = Didx_mu_30(:);
D_muc = mean(Didx_mu_30q);
D_std = std(Didx_mu_30q);
zx =(D_muc-10*D_std):D_std/10:(D_muc+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.7887    2.7887    2.7888    2.7888    2.7889    2.7890    2.7890    2.7891 ···
```

```
gauss_c = mypdf(zx, D_muc, D_std);
%Make Plots
figure(13);
subplot(2,2,1)
histogram(Didx_mu_30q, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Mean')

%for maximum
clear zx, gauss_c, D_std;
```

```
gauss_c = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

```
Didx_max_30q = Didx_max_30(:);
D_mux = mean(Didx_max_30q);
D_std = std(Didx_max_30q);
zx =(D_mux-10*D_std):D_std/10:(D_mux+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.8865    2.8868    2.8871    2.8874    2.8877    2.8880    2.8883    2.8885 ···
```

```
gauss_c = mypdf(zx, D_mux, D_std);
```

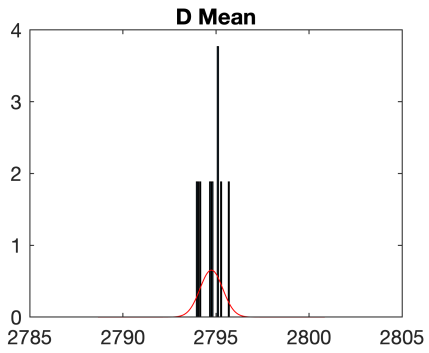
```
subplot(2,2,2)
histogram(Didx_max_30q, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Max')

%for minimum
clear zx, gauss_c, D_std;
```

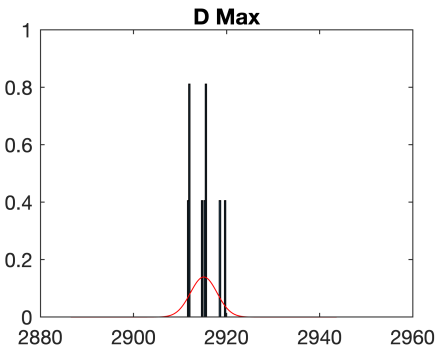
```
gauss_c = 1×201
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000 ···
```

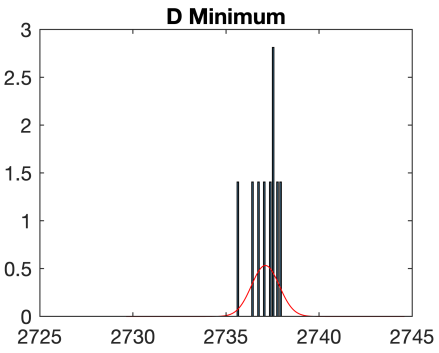
```
Didx_min_30q = Didx_min_30(:);
D_mui = mean(Didx_min_30q);
D_std = std(Didx_min_30q);
zx =(D_mui-10*D_std):D_std/10:(D_mui+10*D_std)
```

```
zx = 1×201
10^3 ×
    2.7296    2.7297    2.7298    2.7299    2.7299    2.7300    2.7301    2.7302 ···
```

```
gauss_c = mypdf(zx, D_mui, D_std);
%Make plots
subplot(2,2,3)
histogram(Didx_min_30q, 30,'Normalization','pdf'); hold on
plot(zx, gauss_c, 'r')
title('D Minimum')
```







Problem 14

```
clear D,Ix, Ix1, Iy1, D_idx, Dstd, Dmax, Dmin, Dmean;
```

```
Ix = 1×37
     1     4     7    10    13    16    19    22    25    28    31    34    37 ···
Ix1 = 1×36
     3     6     9    12    15    18    21    24    27    30    33    36    39 ···
Iy1 = 1×37
     3     6     9    12    15    18    21    24    27    30    33    36    39 ···
D_idx = 36×37
10^3 ×
    2.9041    2.9027    2.9099    2.8985    2.8915    2.8773    2.8795    2.8673 ···
    2.8932    2.8953    2.9087    2.8997    2.8910    2.8974    2.8665    2.8730
    2.8735    2.8961    2.8913    2.8952    2.8948    2.8863    2.8754    2.8679
    2.8671    2.8765    2.8765    2.8825    2.8822    2.8707    2.8755    2.8704
    2.8611    2.8611    2.8736    2.8621    2.8600    2.8578    2.8693    2.8578
    2.8454    2.8522    2.8475    2.8584    2.8508    2.8375    2.8536    2.8437
    2.8285    2.8393    2.8363    2.8276    2.8258    2.8254    2.8335    2.8256
    2.8135    2.8366    2.8220    2.8184    2.8155    2.8155    2.8105    2.8115
    2.8047    2.8105    2.8129    2.8136    2.8321    2.8234    2.8019    2.8013
    2.7852    2.8003    2.7994    2.8020    2.8126    2.8084    2.7985    2.8018
     ⋮
Dstd = 1000×4
   33.5266   43.7335   41.6958   40.3003
   50.3870   37.2108   39.8736   41.4775
   35.1664   36.5607   38.0851   39.9353
   39.2778   51.1845   39.3311   39.7442
   48.3159   41.1038   42.1791   39.2359
   36.4414   33.3949   37.4042   38.7412
   43.4336   39.9271   39.3422   40.2807
   37.0121   42.6038   40.0143   37.9971
   47.8008   46.0650   40.5435   41.6068
   28.0713   37.7360   37.2027   39.3399
     ⋮
Dmax = 1000×4
10^3 ×
    2.8858    2.8984    2.9050    2.9156
    2.8951    2.8981    2.8991    2.9156
    2.8740    2.9005    2.9084    2.9095
    2.8858    2.9069    2.9187    2.9097
    2.9011    2.9009    2.9075    2.9156
    2.8562    2.8894    2.9045    2.9014
    2.9112    2.8857    2.9060    2.9187
    2.9010    2.9079    2.9072    2.9153
    2.8944    2.9121    2.9048    2.9122
    2.8520    2.8879    2.8957    2.9112
     ⋮
Dmin = 1000×4
10^3 ×
    2.7723    2.7421    2.7437    2.7380
    2.7599    2.7472    2.7428    2.7373
    2.7596    2.7430    2.7426    2.7405
    2.7433    2.7455    2.7392    2.7398
    2.7577    2.7501    2.7466    2.7405
    2.7476    2.7426    2.7408    2.7389
    2.7646    2.7389    2.7469    2.7392
    2.7765    2.7446    2.7463    2.7394
    2.7550    2.7438    2.7375    2.7392
```

```
    2.7479    2.7435    2.7422    2.7373
    ⋮
```

```
D = load('elevations.txt')
```

```
D = 110×111
10^3 ×
       NaN    2.8961    2.8966    2.8995    2.9029    2.9056    2.9102    2.9023 ···
       NaN    2.8896    2.8996    2.8978    2.9059    2.9122    2.9041    2.9064
       NaN    2.8868    2.9041    2.8959    2.9006    2.9027    2.9044    2.9040
       NaN    2.8868    2.8919    2.8941    2.8975    2.9009    2.9057    2.9041
       NaN    2.8933    2.8861    2.8908    2.8950    2.8991    2.9060    2.9037
       NaN    2.8808    2.8932    2.8880    2.8933    2.8953    2.9014    2.9043
       NaN    2.8815    2.8844    2.9014    2.9091    2.8937    2.8956    2.9073
    2.8803    2.8792    2.8830    2.8916    2.8985    2.8888    2.8967    2.9025
    2.8788    2.8704    2.8735    2.8844    2.8878    2.8961    2.8886    2.8991
    2.8774    2.8797    2.8730    2.8757    2.8848    2.8905    2.8885    2.8993
    ⋮
```

```
 %isfinits is putting zeros in a new vector where there are non-finite values
                         % and find is finding the values where there are ones and not

nr = 111
```

```
nr = 111
```

```
nc = 110
```

```
nc = 110
```

```
samplesize = [50, 100, 200, 400,500]
```

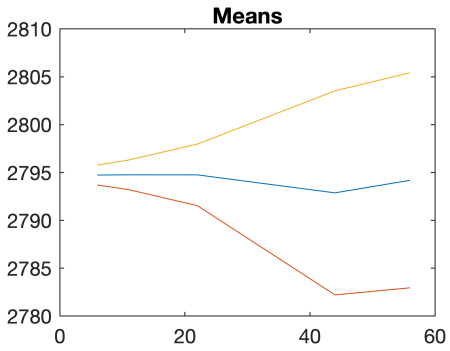
```
samplesize = 1×5
    50   100   200   400   500
```

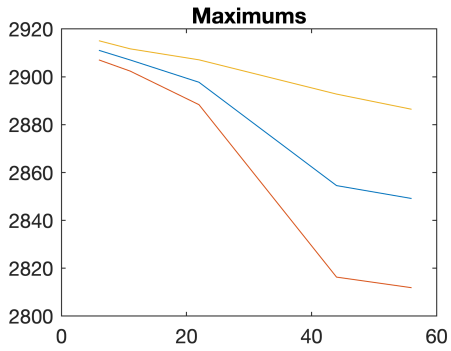
```
for q= 1:length(samplesize)
    nx = round(samplesize(q)/9);

    for x = 1:1:nx  %same for loop as before except now it will vary with sample size
        for y = 1:1:nx
        Iy1 = y:nx:nr;
        Ix1 = x:nx:nc;
        D_idx = D(Ix1, Iy1);
        Didx_mu(x,y) = nanmean(D_idx(:));
        Didx_min(x,y) = nanmin(D_idx(:));
        Didx_max(x,y) = nanmax(D_idx(:));
        Didx_std(x,y) = nanstd(D_idx(:));
        end

    end
    x_mu(q) = nanmean(Didx_mu(:));  %store the mean of the mean values for each sample
    x_std(q) = nanstd(Didx_mu(:)) %store the standard deviation of mean values for each
    len(q) = length(Didx_mu) % number of samples for each sample size 'q'
    x_mux(q) = nanmean(Didx_max(:));
    x_mui(q) = nanmean(Didx_min(:));
```

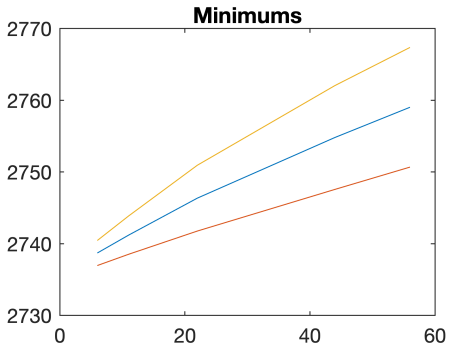
```
    x_stx(q) = nanstd(Didx_max(:));
    x_sti(q) = nanstd(Didx_min(:));
end
```

```
x_std = 1.0383
len = 6
x_std = 1×2
    1.0383    1.5468
len = 1×2
     6    11
x_std = 1×3
    1.0383    1.5468    3.2154
len = 1×3
     6    11    22
x_std = 1×4
    1.0383    1.5468    3.2154   10.6589
len = 1×4
     6    11    22    44
x_std = 1×5
    1.0383    1.5468    3.2154   10.6589   11.2325
len = 1×5
     6    11    22    44    56
```

```
figure(14)
subplot(2,2,1)
plot(len, x_mu); hold on %plot length versus mean
plot(len, x_mu-x_std); hold on %plot length versus mean minus the std
plot(len, x_mu+x_std);  %plot length versus mean plus std
title('Means')
subplot(2,2,2)
plot(len, x_mux); hold on %plot length versus mean
plot(len, x_mux-x_stx); hold on %plot length versus mean minus the std
plot(len, x_mux+x_stx);
title( 'Maximums')
subplot(2,2,3)
plot(len, x_mui); hold on %plot length versus mean
plot(len, x_mui-x_sti); hold on %plot length versus mean minus the std
plot(len, x_mui+x_sti);
title( 'Minimums')
```







GAUSSIAN FUNCTION

```
function f =mypdf(z,mu,sig)

A = 1/(sig*sqrt(2*pi));
B = (z-mu).^2;
C = 2*sig.^2;
f = A*exp(-B./C);
end
```