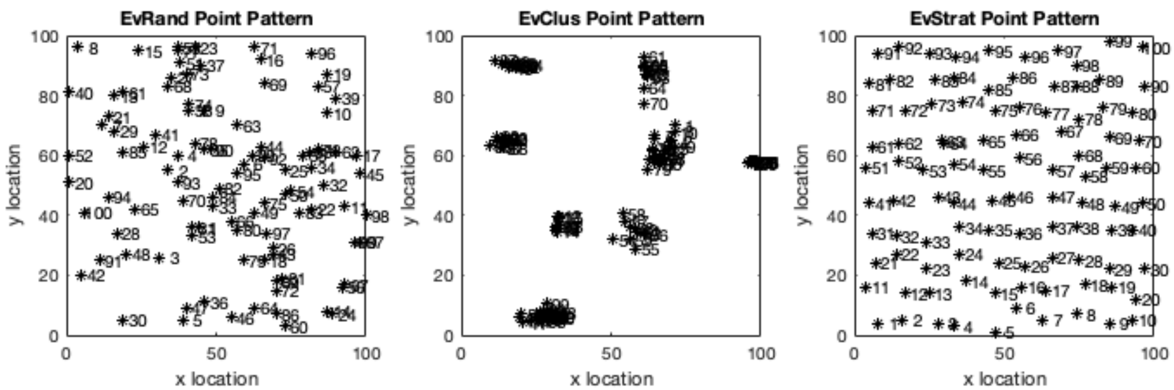
Question 1

```
% Question 1

% Random Point Pattern
[nEvRand , nDimRand] = size(evRand); % event location coordinates
xRand = evRand(:,1);
yRand = evRand(:,2);
idRand = (1:nEvRand)';

% Clustered Point Pattern
[nEvClus , nDimClus] = size(evClus); % event location coordinates
xClus = evClus(:,1);
yClus = evClus(:,2);
idClus = (1:nEvClus)';

% Stratified Point Pattern
[nEvStrat , nDimStrat] = size(evStrat); % event location coordinates
xStrat = evStrat(:,1);
yStrat = evStrat(:,2);
idStrat = (1:nEvStrat)';

% use subplot to organize the 3 separate plots all in one image

% Random Point Pattern
subplot(1,3,1);
plot(xRand,yRand,'k*');
xlabel('x location');
ylabel('y location');
title('EvRand Point Pattern');
axis equal square;
idCharRand = num2str(idRand); % character array with test
text(xRand,yRand,idCharRand);

% Clustered Point Pattern
subplot(1,3,2);
plot(xClus,yClus,'k*');
xlabel('x location');
ylabel('y location');
title('EvClus Point Pattern');
axis equal square;
idCharClus = num2str(idClus); % character array with test
text(xClus,yClus,idCharClus);

% Stratified Point Pattern
subplot(1,3,3);
plot(xStrat,yStrat,'k*');
xlabel('x location');
ylabel('y location');
title('EvStrat Point Pattern');
axis equal square;
idCharStrat = num2str(idStrat); % character array with test
text(xStrat,yStrat,idCharStrat);
```



Question 2

```
% Question 2

% Random Point Pattern
BoundBoxRand = [0 100; 0 100]; % defines study region withmin and max coordinates
CoordsVRand = boundbox2poly(BoundBoxRand); % problem with this line (computes coordinates
of squared vertices
nEventsRand = size(evRand,1);
areaDRand = polyarea(CoordsVRand(:,1), CoordsVRand(:,2)); % defines the area of study
region
lambdaRand = nEventsRand/areaDRand; % defines the global density

% Clustered Point Pattern
BoundBoxClus = [0 100; 0 100]; % defines study region withmin and max coordinates
CoordsVClus = boundbox2poly(BoundBoxClus); % problem with this line (computes coordinates
of squared vertices
nEventsClus = size(evClus,1);
areaDClus = polyarea(CoordsVClus(:,1), CoordsVClus(:,2)); % defines the area of study
region
lambdaClus = nEventsClus/areaDClus; % defines the global density
```

```
% Stratified Point Pattern
BoundBoxStrat = [0 100; 0 100]; % defines study region withmin and max coordinates
CoordsVStrat = boundbox2poly(BoundBoxStrat); % problem with this line (computes
coordinates of squared vertices
nEventsStrat = size(evStrat,1);
areaDStrat = polyarea(CoordsVStrat(:,1), CoordsVStrat(:,2)); % defines the area of study
region
lambdaStrat = nEventsStrat/areaDStrat; % defines the global density
```

Question 3

```
% Question 3

% Random Point Pattern
[idNNRand, distNNRand] = knnsearch([xRand,yRand],[xRand,yRand],'K',2); % compute the
index of the nearest neighbor of each event and corresponding distance
idNNRand(:,1) = []; % delete the first column of both arrays
distNNRand(:,1) = []; % essential because knn function returns two nearest neighbors
where the first column is distance 0

% Clustered Point Pattern
[idNNClus, distNNClus] = knnsearch([xClus,yClus],[xClus,yClus],'K',2); % compute the
index of the nearest neighbor of each event and corresponding distance
idNNClus(:,1) = []; % delete the first column of both arrays
distNNClus(:,1) = []; % essential because knn function returns two nearest neighbors
where the first column is distance 0

% Stratified Point Pattern
[idNNStrat, distNNStrat] = knnsearch([xStrat,yStrat],[xStrat,yStrat],'K',2); % compute
the index of the nearest neighbor of each event and corresponding distance
idNNStrat(:,1) = []; % delete the first column of both arrays
distNNStrat(:,1) = []; % essential because knn function returns two nearest neighbors
where the first column is distance 0
```

Question 4

```
% Question 4

% display the nearest neighbor of some event

% Random Point Pattern
ev2plotRand = 3;
subplot(1,3,1);
plot(xRand,yRand,'*');
hold on;
plot(xRand(ev2plotRand),yRand(ev2plotRand),'ro');
plot(xRand(idNNRand(ev2plotRand)),yRand(idNNRand(ev2plotRand)),'go');
xlabel('x location');
ylabel('y location');
title('EvRand Point Pattern');
hold off;
% add a line segment connecting event 3 with its nearest neighbor
line([xRand(ev2plotRand) xRand(idNNRand(ev2plotRand))],[yRand(ev2plotRand)
yRand(idNNRand(ev2plotRand))])
```

The length of that line segments denotes the distance between event 3 with it's nearest neighbor. We can calculate it using the distance formula.

```
distanceRand = sqrt((xRand(ev2plotRand) - xRand(idNNRand(ev2plotRand)))^2 +
(yRand(ev2plotRand) - yRand(idNNRand(ev2plotRand)))^2); % 11.0454
```

When we compute the distance of this line segment, we get 11.0454.

```
% Clustered Point Pattern
ev2plotClus = 3;
subplot(1,3,2);
plot(xClus,yClus,'*');
hold on;
plot(xClus(ev2plotClus),yClus(ev2plotClus),'ro');
plot(xClus(idNNClus(ev2plotClus)),yClus(idNNClus(ev2plotClus)),'go');
xlabel('x location');
ylabel('y location');
title('EvClus Point Pattern');
hold off;
% add a line segment connecting event 3 with its nearest neighbor
line([xClus(ev2plotClus) xClus(idNNClus(ev2plotClus))],[yClus(ev2plotClus)
yClus(idNNClus(ev2plotClus))])
```

The length of that line segments denotes the distance between event 3 with it's nearest neighbor. We can calculate it using the distance formula.

```
distanceClus = sqrt((xClus(ev2plotClus) - xClus(idNNClus(ev2plotClus)))^2 +
(yClus(ev2plotClus) - yClus(idNNClus(ev2plotClus)))^2); % 1.4139
```

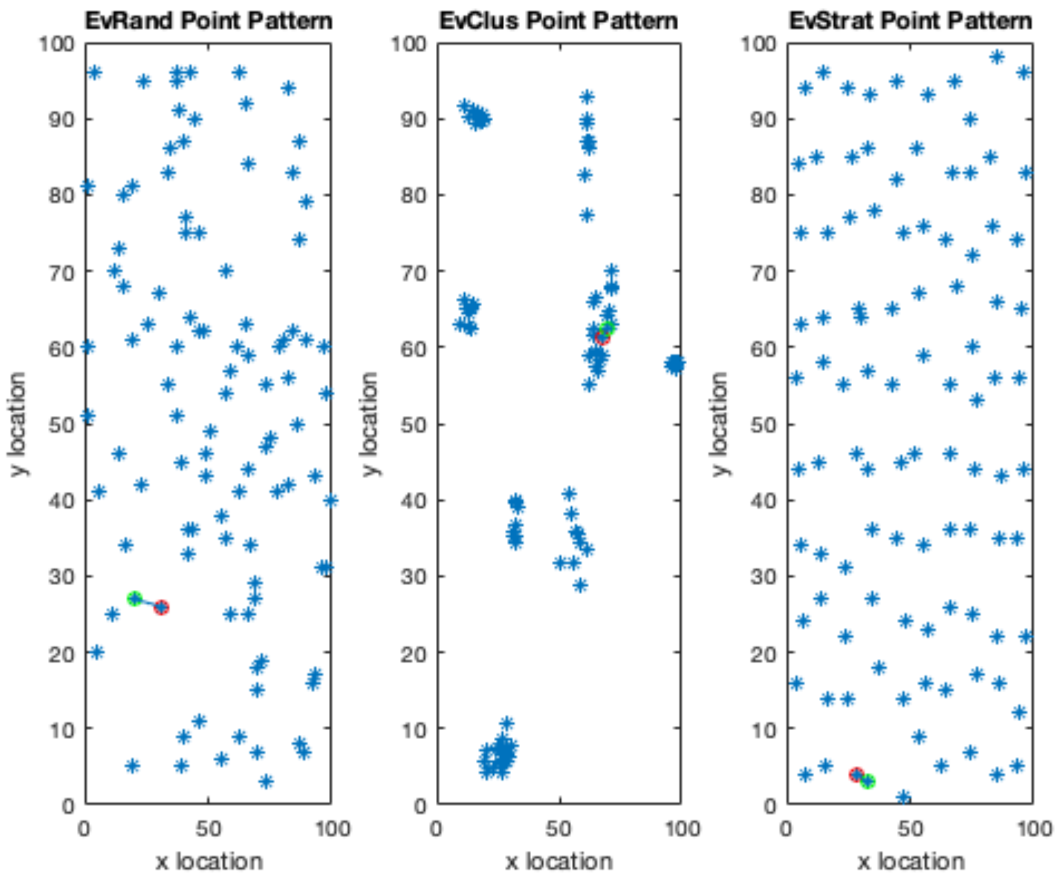
When we compute the distance of this line segment, we get 1.4139.

```
% Stratified Point Pattern
ev2plotStrat = 3;
subplot(1,3,3);
plot(xStrat,yStrat,'*');
hold on;
plot(xStrat(ev2plotStrat),yStrat(ev2plotStrat),'ro');
plot(xStrat(idNNStrat(ev2plotStrat)),yStrat(idNNStrat(ev2plotStrat)),'go');
xlabel('x location');
ylabel('y location');
title('EvStrat Point Pattern');
hold off;
% add a line segment connecting event 3 with its nearest neighbor
line([xStrat(ev2plotStrat) xStrat(idNNStrat(ev2plotStrat))],[yStrat(ev2plotStrat)
yStrat(idNNStrat(ev2plotStrat))])
```

The length of that line segments denotes the distance between event 3 with its nearest neighbor. We can calculate it using the distance formula.

```
distanceStrat = sqrt((xStrat(ev2plotStrat) - xStrat(idNNStrat(ev2plotStrat)))^2 +
(yStrat(ev2plotStrat) - yStrat(idNNStrat(ev2plotStrat)))^2); % 5.0990
```

When we compute the distance of this line segment, we get 5.0990.



Question 5

```
% Question 5

% Random Point Pattern
R_0Rand = mean(distNNRand); % observed mean nearest neighbor distance
R_eRand = 1/(2*sqrt(lambdaRand)); % expected value of R if pattern is random
RRand = R_0Rand/R_eRand; % R statistic
```

The R statistic is 1.0366. This is very close to the expected value for this R statistic, which is 1, so I assume there isn't enough significant evidence to conclude a difference between the observed and expected mean nearest neighbor distance.

```
% Clustered Point Pattern
R_0Clus = mean(distNNClus); % observed mean nearest neighbor distance
R_eClus = 1/(2*sqrt(lambdaClus)); % expected value of R if pattern is random
RClus = R_0Clus/R_eClus; % R statistic
```

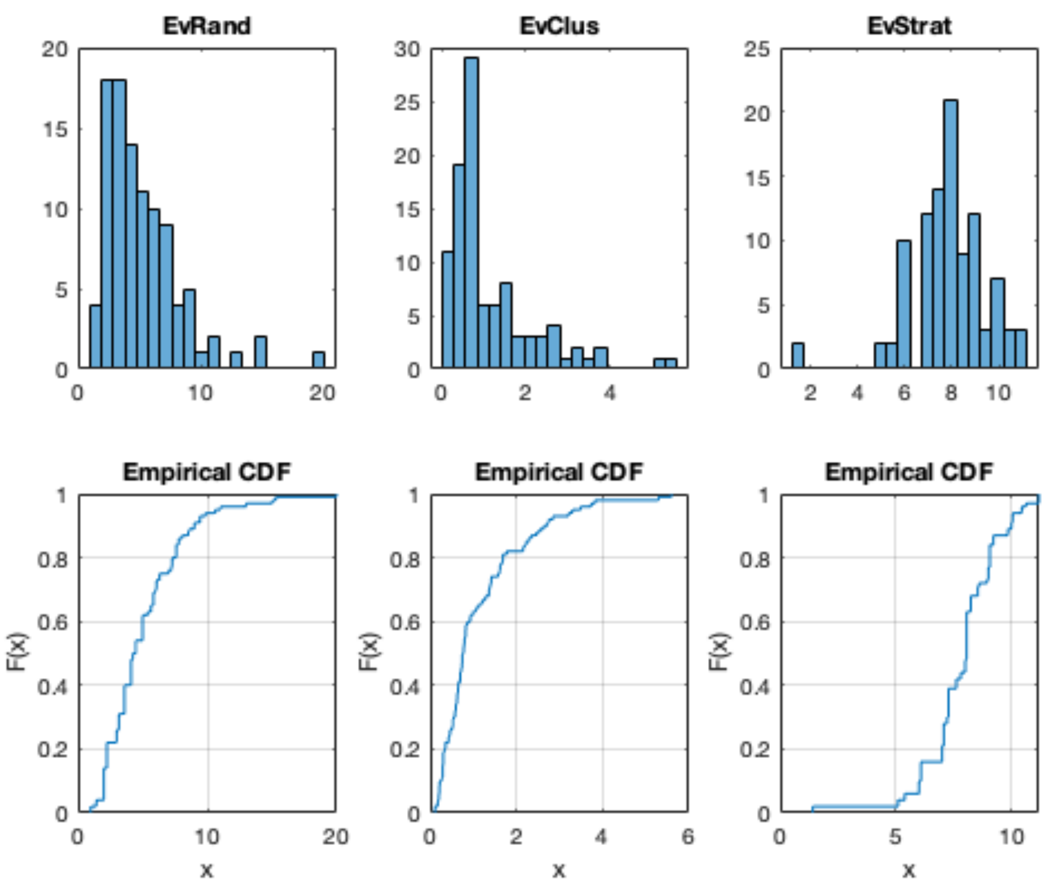
The R statistic is 0.2334. This is significantly less than the expected value for this R statistic, which is 1, so I assume this means that the points are much more clustered than expected.

```
% Stratified Point Pattern
R_0Strat = mean(distNNStrat); % observed mean nearest neighbor distance
R_eStrat = 1/(2*sqrt(lambdaStrat)); % expected value of R if pattern is random
RStrat = R_0Strat/R_eStrat; % R statistic
```

The R statistic is 1.5730. This is more than the expected value for this R statistic, which is 1, so I assume this means that the points are more stratified than expected.

Question 6

```
% Question 6

% Random Point Pattern
subplot(2,3,1);
histogram(distNNRand,20); % compute the histogram of nearest neighbor distances
title('EvRand');
subplot(2,3,4);
cdfplot(distNNRand); % G function for the point pattern as CDF of distances

% Clustered Point Pattern
subplot(2,3,2);
histogram(distNNClus,20); % compute the histogram of nearest neighbor distances
title('EvClus');
subplot(2,3,5);
cdfplot(distNNClus); % G function for the point pattern as CDF of distances

% Stratified Point Pattern
subplot(2,3,3);
histogram(distNNStrat,20); % compute the histogram of nearest neighbor distances
title('EvStrat');
subplot(2,3,6);
cdfplot(distNNStrat); % G function for the point pattern as CDF of distances
```



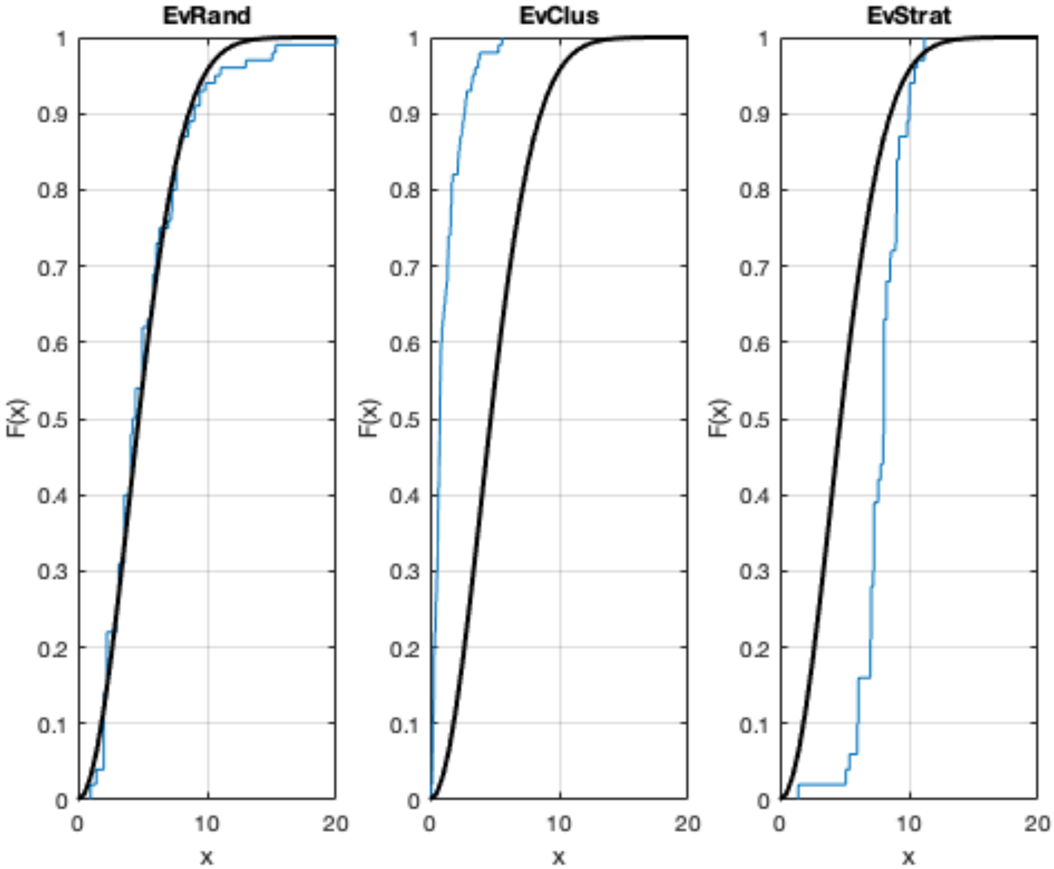
Random Point Pattern Plot: This histogram looks to be positively skewed. This means that nearest neighbors are closer to each other than farther apart, which is expected. Looking at the cdf plot, we see that 50% of the point locations have a nearest neighbor of less than or equal to about 4.5 distance units away. Approximately 94% of all events lie within 10 distance units away from their nearest neighbors.

Clustered Pattern Plot:
This histogram looks to be positively skewed. This means that nearest neighbors are closer to each other than farther apart, which is expected. Compared to evRand, this histogram has a much higher peak and density at lower values. This means that there are just more samples that have closer near distances than far in this dataset than the random one.
Looking at the cdf plot, we see that 50% of the point locations have a nearest neighbor of less than or equal to about 1.75 distance units away. Since 100% of the point locations have a nearest neighbor distance of less than 6 distance units, then the proportion of events that lie at most 10 distance units away from their nearest neighbor is 100%.

Stratified Pattern Plot:
This histogram looks to be normally distributed. This means that every point's nearest neighbor is the same distance away. This has a higher peak than the randomly scattered dataset but a lower peak than the clustered dataset.
Looking at the cdf plot, we see that 50% of the point locations have a nearest neighbor of about 8 distance units away. About 90% of the points in this dataset lies within 10 distance units away from their nearest neighbor. This is less than the randomly scattered points but significantly more than the clustered point patterns.

Question 7

```
% Question 7
% Random Point Pattern
DRand = 0:0.5:20;
GRand = 1 - exp(-lambdaRand*pi*DRand.^2); % G function needs lambda to work
subplot(1,3,1);
cdfplot(distNNRand);
hold on;
plot(DRand, GRand, '-k', 'LineWidth',2);% compare the G function under a random pattern
with observations
title('EvRand');

% Clustered Point Pattern
DClus = 0:0.5:20;
GClus = 1 - exp(-lambdaClus*pi*DClus.^2); % G function needs lambda to work
subplot(1,3,2);
cdfplot(distNNClus);
hold on;
plot(DClus, GClus, '-k', 'LineWidth',2); % compare the G function under a random pattern
with observations
title('EvClus');
% Stratified Point Pattern
DStrat = 0:0.5:20;
GStrat = 1 - exp(-lambdaStrat*pi*DStrat.^2); % G function needs lambda to work
subplot(1,3,3);
cdfplot(distNNStrat);
hold on;
plot(DStrat, GStrat, '-k', 'LineWidth',2); % compare the G function under a random
pattern with observations
title('EvStrat');
```



Random Point Pattern Plot: *I think the G function follows the cdf function well. It looks like the G function overestimated the cdf at the beginning and the end of the plots (i.e., where the F(x) = 0 and F(x) = 1. However, when the function is increasing to a cdf of 1, there are more often when it underestimates the cdf (i.e. where the curve lies under the step function).*

Clustered Pattern Plot: *Based on the plot above, it looks like the G function does not fit the empirical cdf well. It looks like the G function underestimates the cdf of distance units throughout the whole sample space. The empirical cdf increases at a much faster rate than what is expected and expressed through the G function plot.*

Stratified Pattern Plot: *Based on the plot above, it looks like the G function does not fit the empirical cdf well. It looks like the G function overestimates the cdf of distance units throughout the whole sample space. The empirical cdf starts out increasing at a much slower rate than what is expected and expressed through the G function plot; however the slope becomes steeper at around 7 distance units and catches up to the G function at around 11 distance units.*

In General, when the point pattern is random, the G function fits/follows the ecdf function well. When the point pattern is clustered, the G function underestimates (lies under) the ecdf function. And finally when the point pattern is stratified, the G function overestimates (lies above) the ecdf function.