```
rng('shuffle')
% MEAN and SD are constant in exercise 1 and 3
MEAN = 500;
SD = 100;
```

exercise 1

satscores is (100 x 10) matrix of scores that normally distributed and its mean = 500, sd = 100

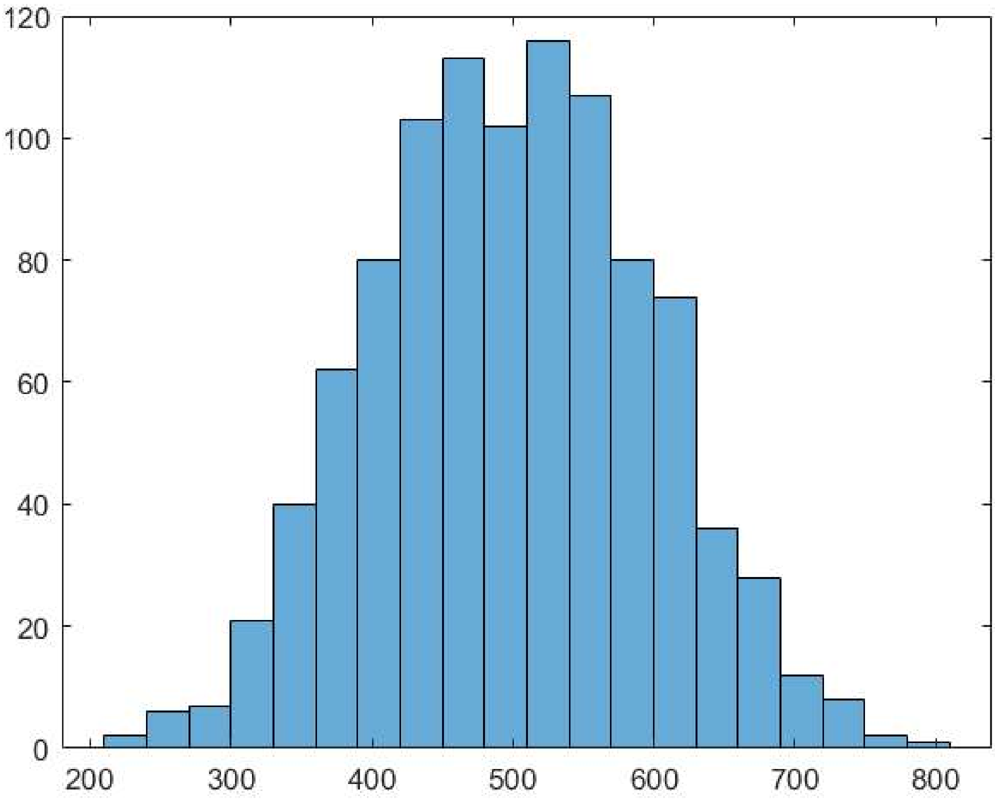
```
satscores = randn(100,10)*SD + MEAN;
mean_satscores = mean(satscores(:))
```

```
mean_satscores = 498.6029
```

```
std_satscores = std(satscores(:))
```

```
std_satscores = 97.4890
```

```
histogram(satscores(:));
```



exercise 2

M is a random integer vector that has 100 elements

```
M = randi(100,1,100);
mean_M = mean(M(:))
```

```
mean_M = 45.5900
```

```
std_M = std(M(:))
```

```
std_M = 27.6047
```

```
idx = randi(100,1,20); % randomly draw idxArray
drawedValues = M(idx(:)); % randomly draw 20 elements from M
mean_drawedValues = mean(drawedValues(:))
```

```
mean_drawedValues = 44.7000
```

```
std_drawedValues = std(drawedValues(:))
```

```
std_drawedValues = 23.2562
```

excercise 3

SATs is (1400 x 2) matrix of multiples of 10 that normally distributed and its mean = 500, sd = 100

```
SATs = round(randn(1400,2)*SD + MEAN,-1);
colmean_SATs = mean(SATs)
```

```
colmean_SATs = 1×2
  499.4214  498.4000
```

```
colstd_SATs = std(SATs)
```

```
colstd_SATs = 1×2
  101.4415   99.8262
```

exercise 4 to 7

mathScores is (100 x 3) matrix that contains NaNs

```
load('mathScores.mat');

% exercise 4
classmean_mathScores = nanmean(mathScores)
```

```
classmean_mathScores = 1×3
   62.7353   85.2002   75.2762
```

```
classstd_mathScores = nanstd(mathScores)
```

```
classstd_mathScores = 1×3
   14.5161    4.7039    9.8834
```

```
mean_mathScores = nanmean(mathScores(:))
```

```
mean_mathScores = 74.8199
```

```
std_mathScores = nanstd(mathScores(:))
```

```
std_mathScores = 13.7884
```

```

% exercise 5
vacancy = numel(mathScores(isnan(mathScores)))
vacancy = 30
```

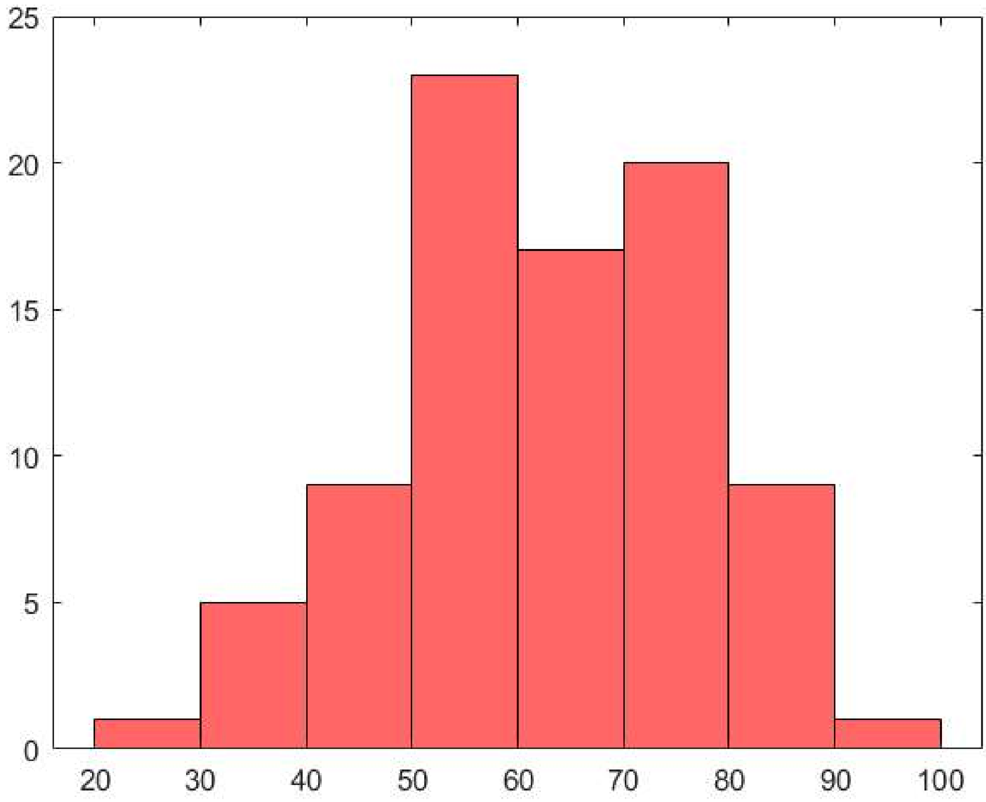
```
% exercise 6
newScores = mathScores; % copy
newScores(isnan(newScores)) = -10; % replace NaN with -10
mean_newScores = mean(newScores(:))
```

```
mean_newScores = 66.3379
```

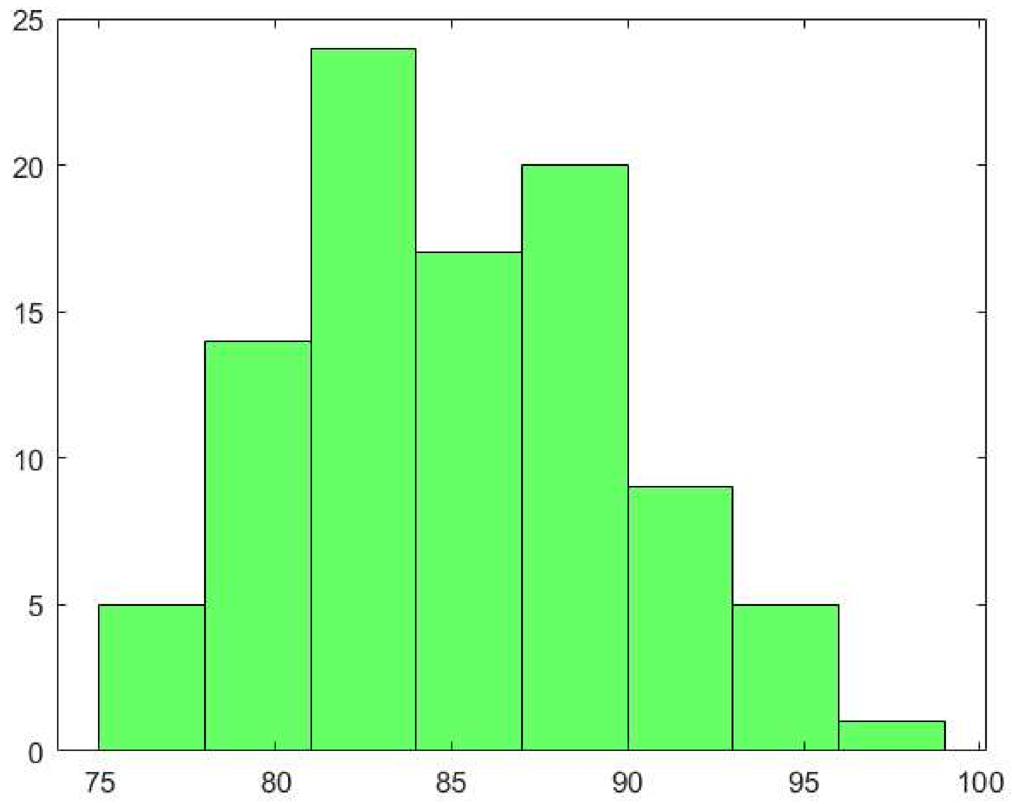
```
std_newScores = std(newScores(:))
```

```
std_newScores = 28.6480
```

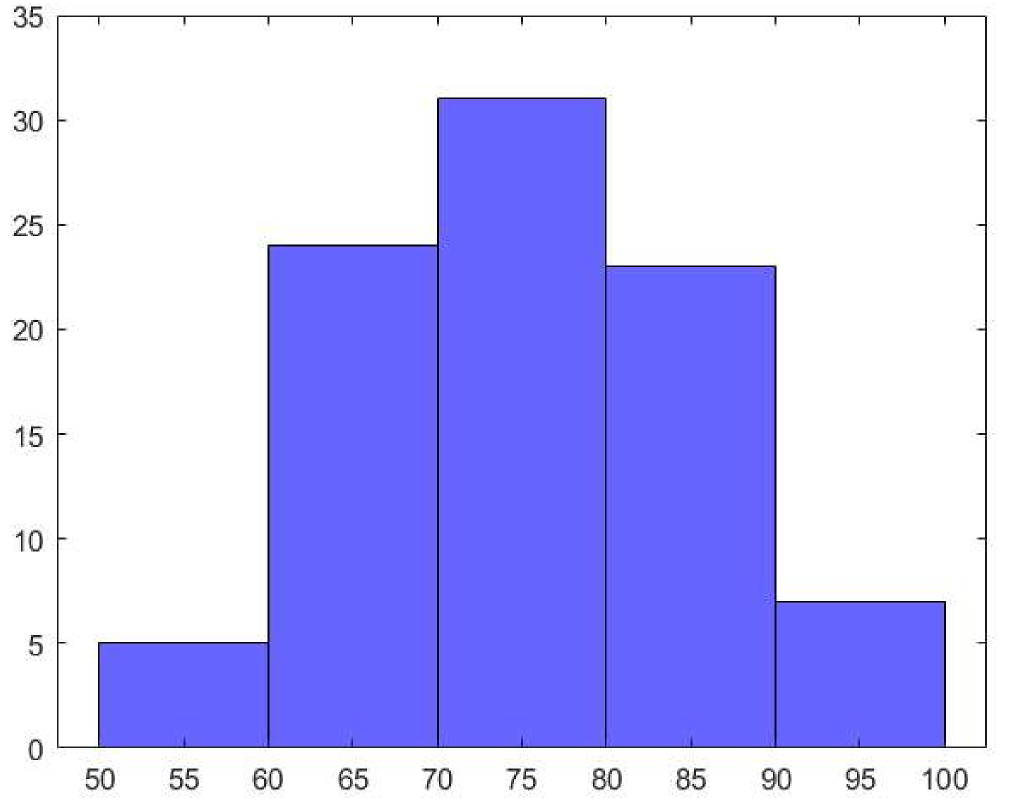
```
% exercise 7
% old scores histograms
histogram(mathScores(:,1),'FaceColor','r');
```



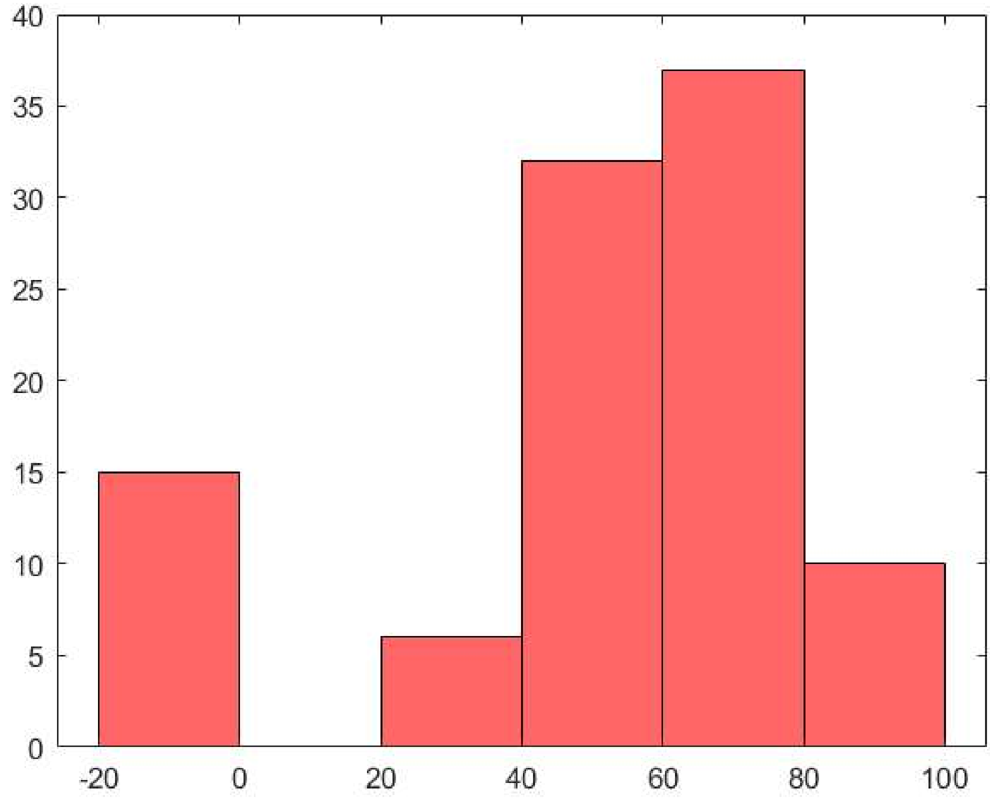
```
histogram(mathScores(:,2),'FaceColor','g');
```



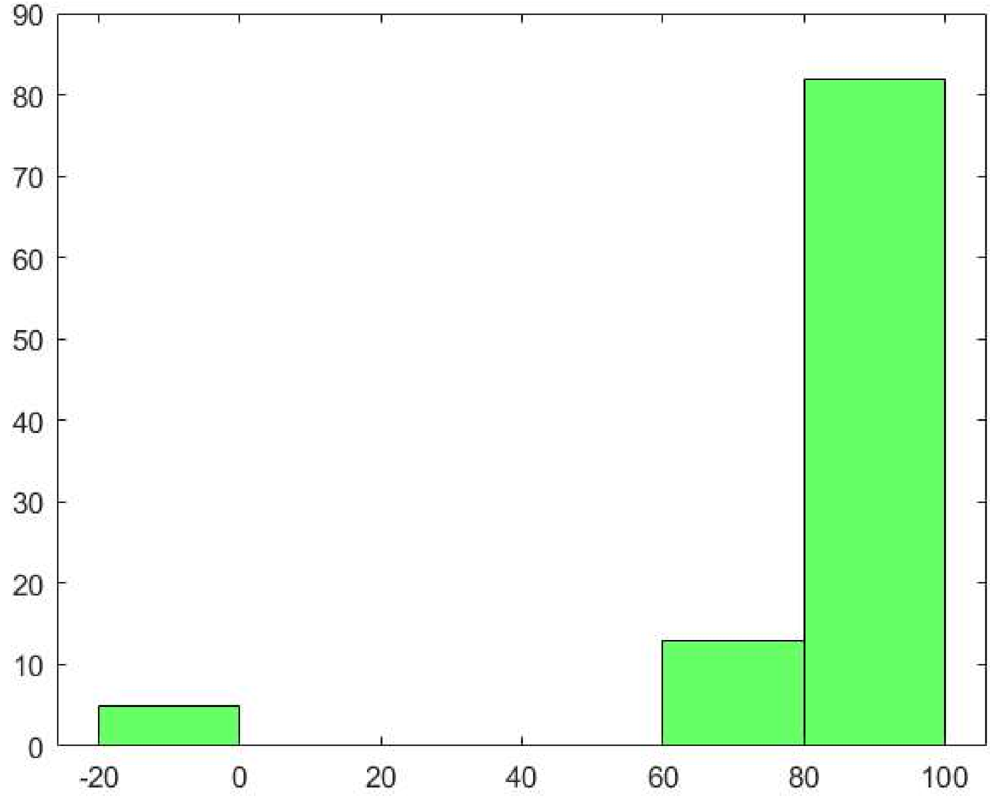
```
histogram(mathScores(:,3),'FaceColor','b');
```



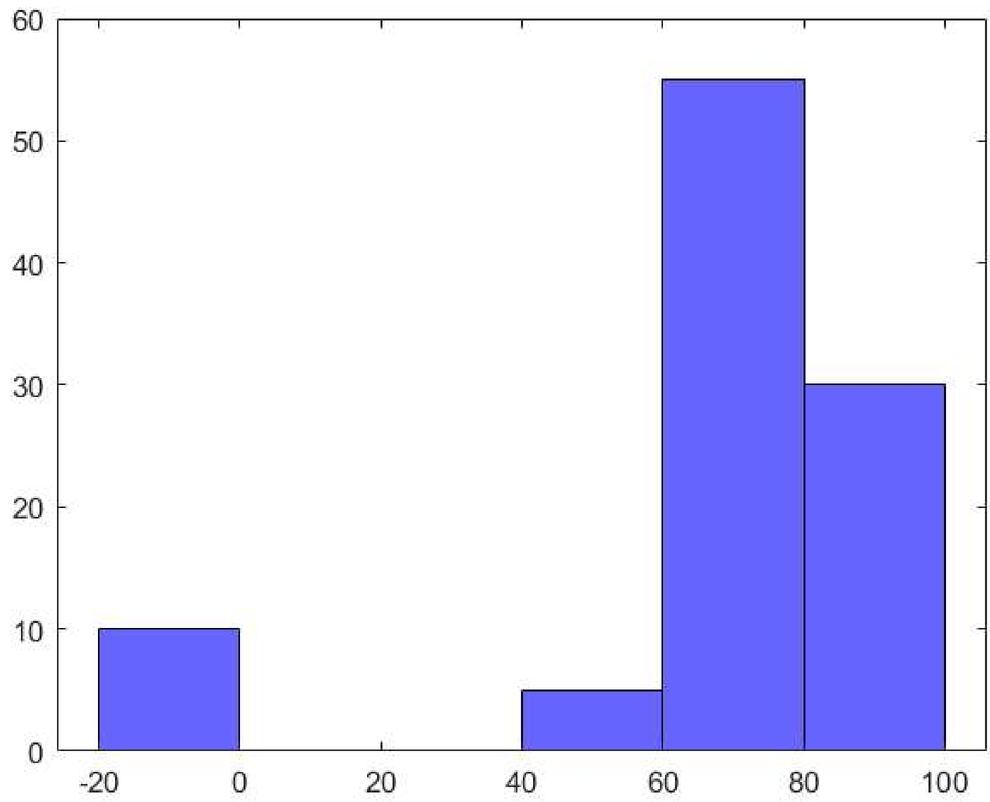
```
% new scores histograms
histogram(newScores(:,1),'FaceColor','r');
```



```
histogram(newScores(:,2),'FaceColor','g');
```



```
histogram(newScores(:,3),'FaceColor','b');
```



exercise 8

floor(x) returns the greatest integer less than or equal to x ceil(x) returns the least integer greater than or equal to x

```
floor_pos = floor(pi);
ceil_pos = ceil(pi);
fix_pos = fix(pi);
table(floor_pos, ceil_pos, fix_pos)
```

```
ans = 1×3 table
```

	floor_pos	ceil_pos	fix_pos
1	3	4	3

```
floor_neg = floor(-pi);
ceil_neg = ceil(-pi);
fix_neg = fix(-pi);
table(floor_neg, ceil_neg, fix_neg)
```

```
ans = 1×3 table
```

	floor_neg	ceil_neg	fix_neg
1	-4	-3	-3

```
% if x is positive, fix(x)==floor(x) as shown in the first table
% if x is negative, fix(x)==ceil(x) as shown in the second table
% Generaly, fix(x)==round(x, 0)
```