Numerical methods in Fortran: Lec. IV

Devendra Singh Bhakuni

dbhakuni@ictp.it

/afs/ictp/public/d/dbhakuni/NM_course/NM_lec4.pdf

October 27, 2023

Topics to be covered

- Random numbers generators, middle square method, linear congruent method, in-built FORTRAN random number generators.

- Non-uniform random number generators: Transformation method, exponential and Gaussian random number generator.

- Rejection method and Monte-Carlo methods.

- *Non parametric density estimation.*

Outline:

- Get the empirical cumulative distribution function.
  - How to sort a the numbers in a vector? The bubble method.
- A first approximation to the PDF: Histograms.
  - Choosing the number of bins
- More elaborated PDF: Kernel Distribution Function.

Cumulative distribution function

- Footprint of your distribution.

$$C(x) = \int_{x_{\min}}^{x} f(x')dx'$$

- It is easy to obtain numerically from a given set of points:
  1. Given a set of M random numbers of unknown distribution $\{x_i\}$
  2. Sort them in ascending order obtaining $\{x_r\}$.
  3. The empirical cumulative distribution function is

$$C_{emp}(x) = \frac{r}{M}$$

Cumulative distribution function

- Footprint of your distribution.

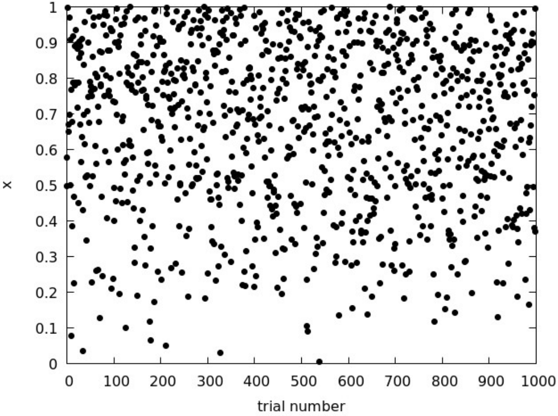
$$C(x) = \int_{x_{\min}}^{x} f(x')dx'$$

- It is easy to obtain numerically from a given set of points:
  1. Given a set of M random numbers of unknown distribution $\{x_i\}$
  2. Sort them in ascending order obtaining $\{x_r\}$.
  3. The empirical cumulative distribution function is

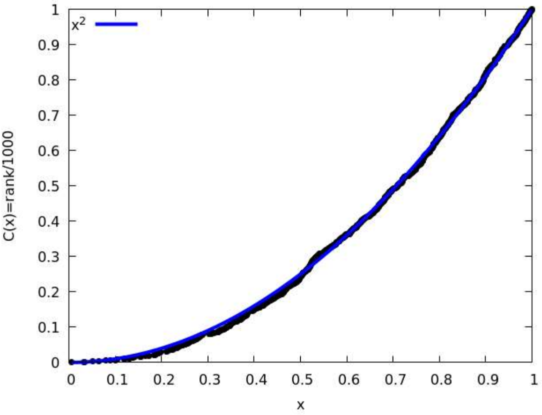
$$C_{emp}(x) = \frac{r}{M}$$

Example

- I generated 1000 points (M=1000) from the PDF f(x)=2x



Example



Sorting (inefficiently)

```
subroutine bubble(v,m)
! m is the number of elements in v
     integer :: i,newn,m,n
     real*8 :: v(m),tmp
     n=m
     do while (n>1)
         newn=0
         do i=2,n
               if (v(i-1)>v(i)) then
                  tmp=v(i)
                  v(i)=v(i-1)
                  v(i-1)=tmp
                  newn=i
               endif
         enddo
        n=newn
     enddo
end subroutine bubble
```

Probability distribution function

- Directly approximate f(x)
- The naïve way is the Histogram:
  - If you have a distribution of a variable x between [xmin,xmax]:
    1. Divide the x range into bins: $\Delta x=(x_{max}-x_{min})/N_{bin}$ N_{bin}: # of bins
    2. Create an array for the histogram H[1:N_{bin}] (initialize to 0)
    3. Each time you generate x check which “bin” it falls into.
    4. H[bin]=H[bin]+1
    5. Normalize,

$$M=\sum_{i=1}^{N_{bin}} H[i]$$

$$H[bin]=H[bin]/(M\Delta x)$$

$$x[i]=x_{\min}+(i-0.5)\Delta x$$

Checking if a point belongs to a bin

Checking if a point belongs to a bin

•With the IF construction: Check that x belongs to the interval between x_{min}+(nbin-1)*Δx and x_{min}+nbin*Δx… How would you program it in FORTRAN?

Checking if a point belongs to a bin

•With the IF construction: Check that x belongs to the interval between x_{min}+(nbin-1)*Δx and x_{min}+nbin*Δx… How would you program it in FORTRAN?

```
DO j=1,MAXBINS
   IF ((x>xmin+(j-1)*dx).AND.(x<=xmin+j*dx)) H(j)=H(j)+1
ENDDO
```

Checking if a point belongs to a bin

- With the IF construction: Check that x belongs to the interval between x_{min}+(nbin-1)*Δx and x_{min}+nbin*Δx… How would you program it in FORTRAN?

```
DO j=1,MAXBINS
   IF ((x>xmin+(j-1)*dx).AND.(x<=xmin+j*dx)) H(j)=H(j)+1
ENDDO
```

- Is there a wiser manner?

Checking if a point belongs to a bin

- With the IF construction: Check that x belongs to the interval between x_{min}+(nbin-1)*Δx and x_{min}+nbin*Δx… How would you program it in FORTRAN?

```
DO j=1,MAXBINS
   IF ((x>xmin+(j-1)*dx).AND.(x<=xmin+j*dx)) H(j)=H(j)+1
ENDDO
```

- Is there a wiser manner?

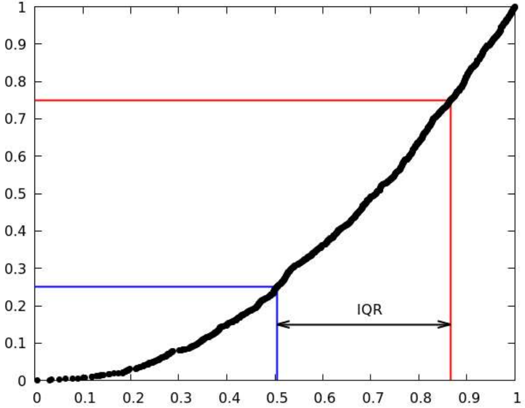
```
j=FLOOR((x-xmin)/dx)+1
H(j)=H(j)+1
```

Freedman-Diaconis rule

$$\Delta x \approx IQR \frac{2}{M^{1/3}}$$

- IQR= Interquartile range. Easy to compute:
  $IQR = C_{emp}^{-1}(0.75) - C_{emp}^{-1}(0.25)$
- It is equivalent to $\mathrm{IQR} = x_{3M/4} - x_{M/4}$ in the sorted vector $\{x_r\}$

IQR



Freedman-Diaconis rule

$$\left(\Delta x\right)_{prox} = IQR\frac{2}{M^{1/3}}$$

- Compute IQR and $(\Delta x)_{prox}$
- Compute the number of bins

$$N_{bin} = floor\left(\frac{\left(x_{\max} - x_{\min}\right)}{\left(\Delta x\right)_{prox}}\right) + 1$$

- Compute real value of $\Delta x = (x_{max} - x_{min}) / N_{bin}$

LET'S DO IT

Doing a histogram

Doing a histogram

1. Obtain the information from your data (this allows you to set the parameters for the histogram).

Doing a histogram

1. Obtain the information from your data (this allows you to set the parameters for the histogram).
2. Perform the counts for the histogram (this is the computational part).

Doing a histogram

1. Obtain the information from your data (this allows you to set the parameters for the histogram).
2. Perform the counts for the histogram (this is the computational part).
3. Normalize (This allows you to compare histograms with different parameters or with other ways of computing the pdf).

Array of random numbers

Obtain the information from your data:

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

Array of random numbers

Obtain the information from your data:

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Obtain the information from your data:

Array of random numbers		
1.5		-2.3
3.2		-1.5
-0.3		-0.7
0.6		-0.5
-2.3		-0.3
-1.5	SORT	0.6
-0.7		1.2
1.2		1.5
2.1		1.7
2.2		2.1
1.7		2.2
-0.5		3.2

M=12

Obtain the information from your data:

Array of random numbers			
1.5		-2.3	x_{min}
3.2		-1.5	
-0.3		-0.7	
0.6		-0.5	
-2.3	SORT	-0.3	
-1.5		0.6	
-0.7		1.2	
1.2		1.5	
2.1		1.7	
2.2		2.1	
1.7		2.2	
-0.5		3.2	x_{max}

M=12

Obtain the information from your data:

Array of random numbers		
1.5	-2.3	x_{min}
3.2	-1.5	
-0.3	-0.7	
0.6	-0.5	
-2.3	-0.3	
-1.5	0.6	
-0.7	1.2	
1.2	1.5	
2.1	1.7	
2.2	2.1	
1.7	2.2	
-0.5	3.2	x_{max}

SORT

$$\mathrm{IQR} = x_{3M/4} - x_{M/4} = x_9 - x_3 = 2.1 - (-0.7) = 2.8$$

M=12

Obtain the information from your data:

Array of random numbers			
1.5		-2.3	x_{min}
3.2		-1.5	
-0.3		-0.7	
0.6		-0.5	
-2.3	SORT	-0.3	
-1.5		0.6	
-0.7		1.2	
1.2		1.5	
2.1		1.7	
2.2		2.1	
1.7		2.2	
-0.5		3.2	x_{max}

M=12

$$\text{IQR} = x_{3M/4} - x_{M/4} = x_9 - x_3 = 2.1 - (-0.7) = 2.8$$

$$(\Delta x)_{prox} = IQR\frac{2}{M^{1/3}} = 2.8\frac{2}{12^{1/3}} = 2.446$$

Obtain the information from your data:

Array of random numbers	SORT	
1.5	-2.3	x_{min}
3.2	-1.5	
-0.3	-0.7	
0.6	-0.5	
-2.3	-0.3	
-1.5	0.6	
-0.7	1.2	
1.2	1.5	
2.1	1.7	
2.2	2.1	
1.7	2.2	
-0.5	3.2	x_{max}

M=12

$$\mathrm{IQR} = x_{3M/4} - x_{M/4} = x_9 - x_3 = 2.1 - (-0.7) = 2.8$$

$$(\Delta x)_{prox} = IQR\frac{2}{M^{1/3}} = 2.8\frac{2}{12^{1/3}} = 2.446$$

$$N_{bin} = floor\left(\frac{(x_{\max} - x_{\min})}{(\Delta x)_{prox}}\right) + 1$$

$$N_{bin} = floor\left(\frac{(3.2-(-2.3))}{2.446}\right) + 1 = floor\left(\frac{5.5}{2.446}\right) + 1 = 3$$

Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

x

Array of random numbers

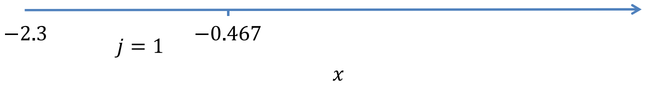
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$



Array of random numbers

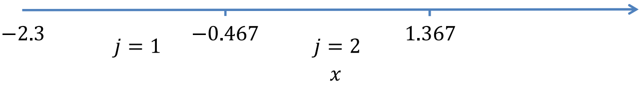
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$



Counts for the histogram:

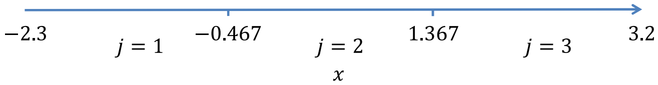
Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$



Array of random numbers

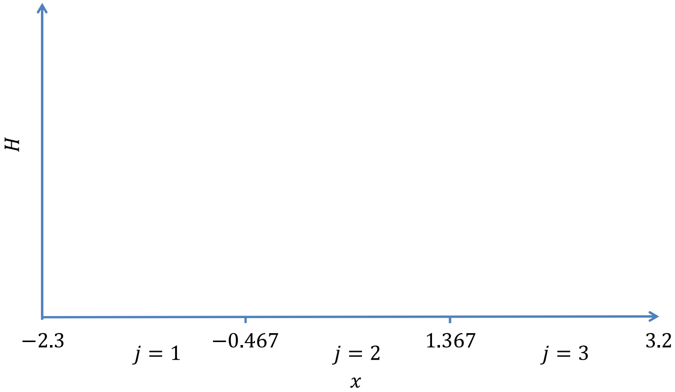
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$



Counts for the histogram:

Array of random numbers

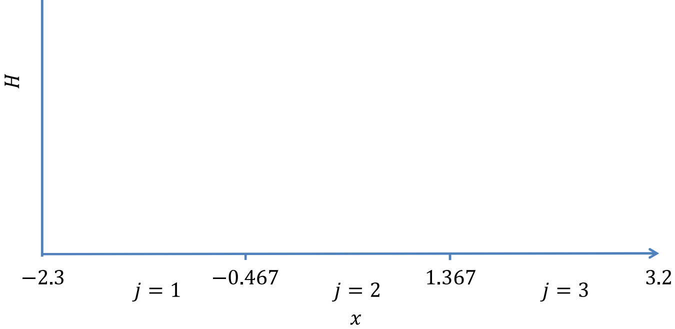
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

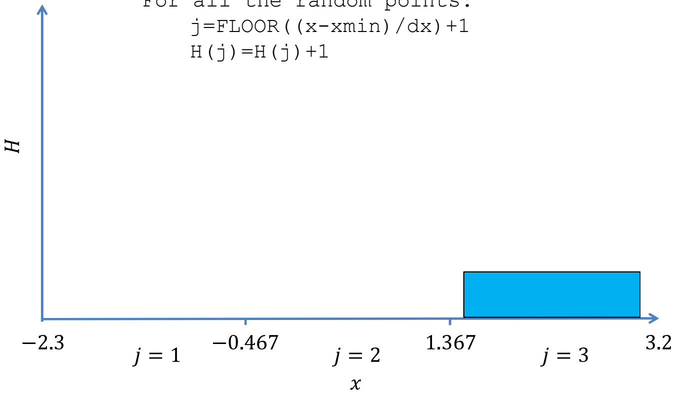
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

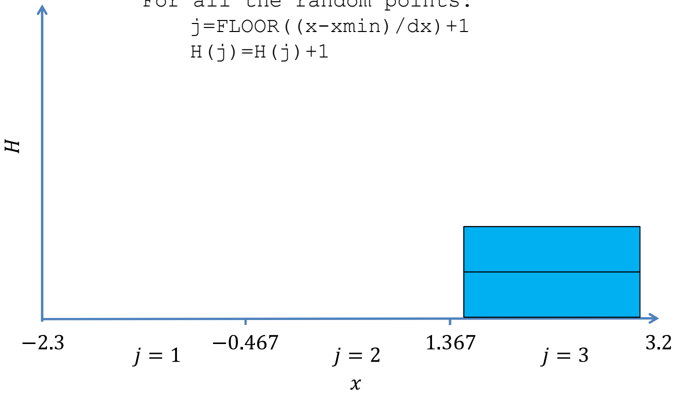
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}}\right) = \left(\frac{(3.2-(-2.3))}{3}\right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

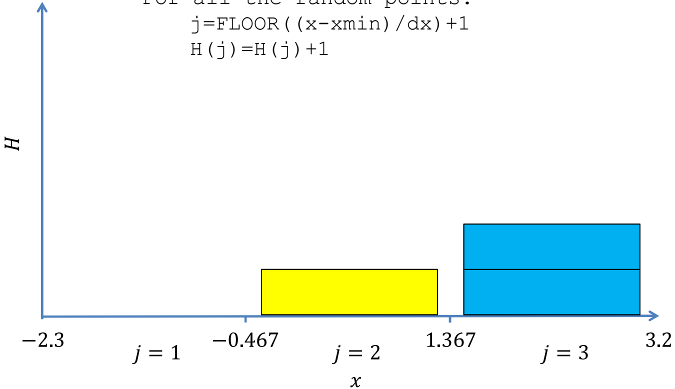
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

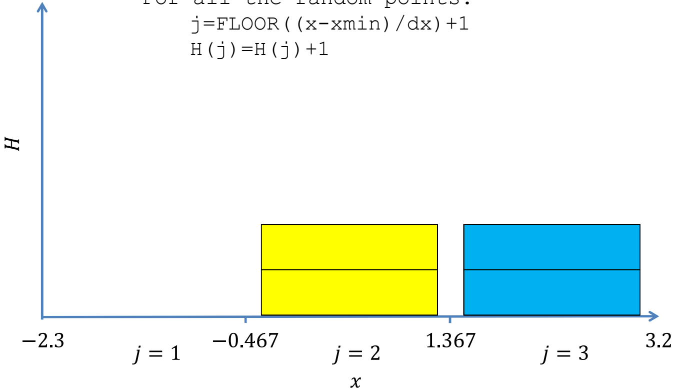
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Counts for the histogram:

Array of random numbers

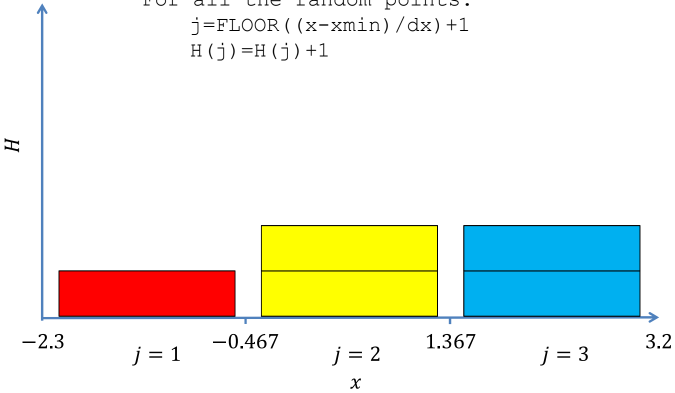
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of
random
numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

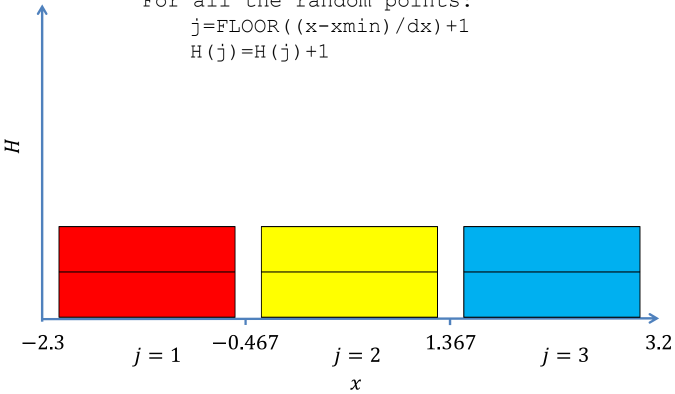
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

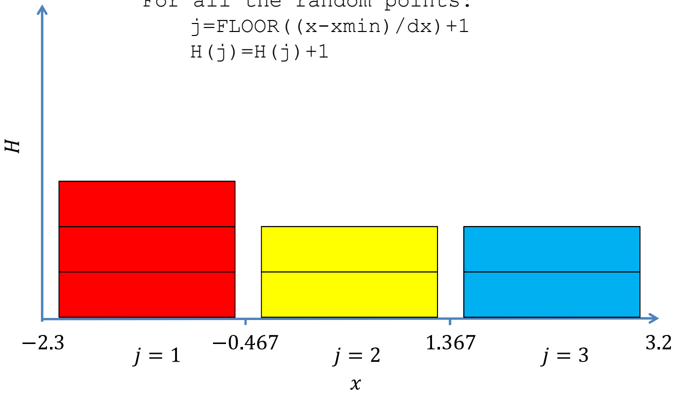
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Counts for the histogram:

Array of random numbers

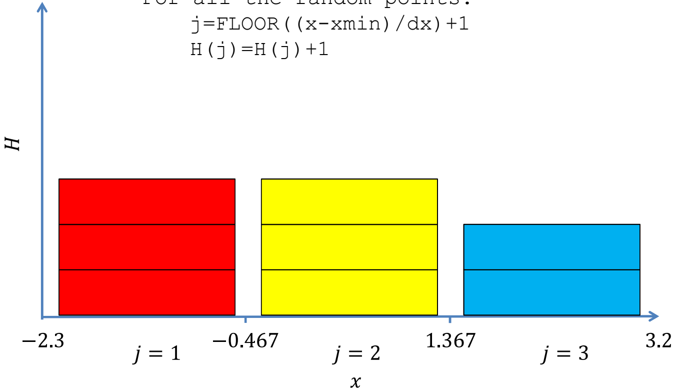
1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

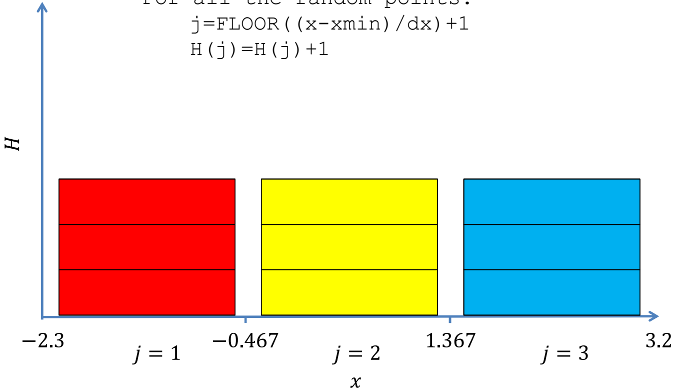
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

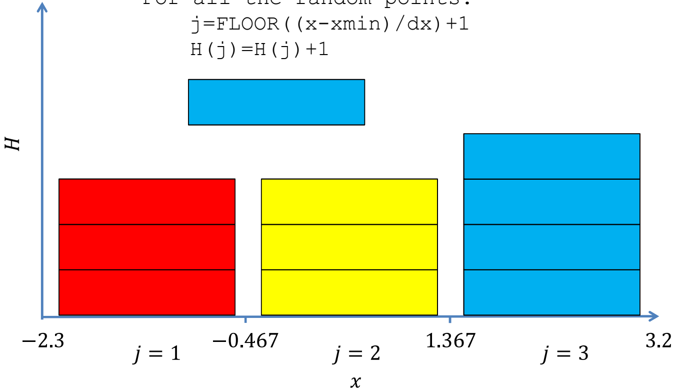
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

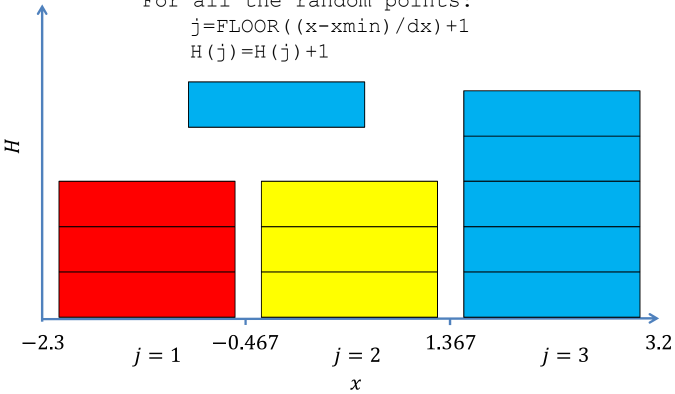
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}}\right) = \left(\frac{(3.2-(-2.3))}{3}\right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

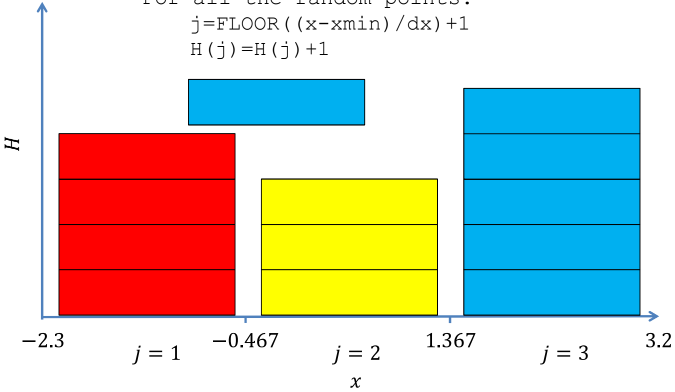
M=12

Counts for the histogram:

$$N_{bin} = 3$$

$$\Delta x = \left(\frac{(x_{\max} - x_{\min})}{N_{bin}} \right) = \left(\frac{(3.2 - (-2.3))}{3} \right) = 1.833$$

```
For all the random points:
    j=FLOOR((x-xmin)/dx)+1
    H(j)=H(j)+1
```

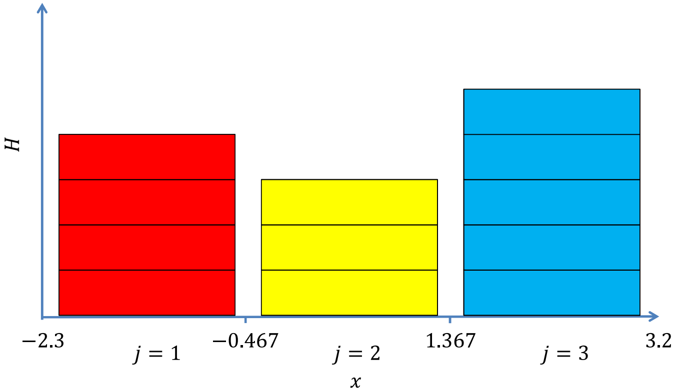


Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

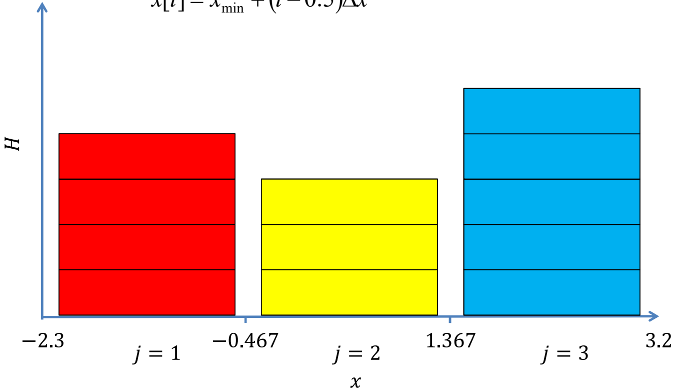
M=12

Normalize:

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

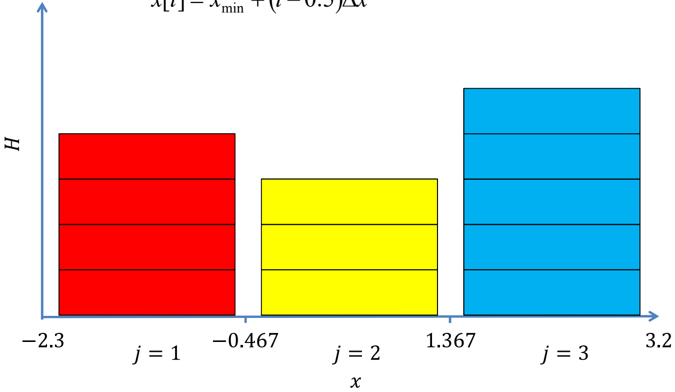
Normalize:

$$H[1] = \frac{4.}{12.* 1.83333}$$

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:

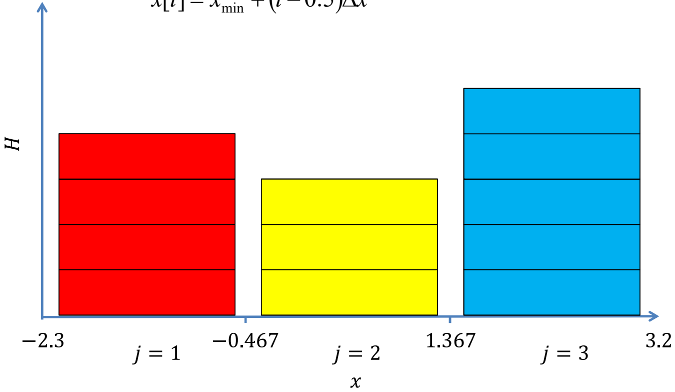
$$H[1] = \frac{4.}{12.* 1.83333}$$

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i-0.5)\Delta x$$

$$H[2] = ?$$



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

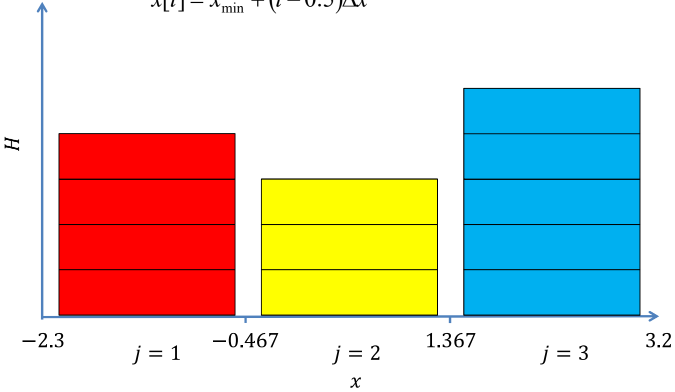
$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$

$$H[1] = \frac{4.}{12.* 1.83333}$$

$$H[2] = ?$$

$$x[1] = ?$$



Array of
random
numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

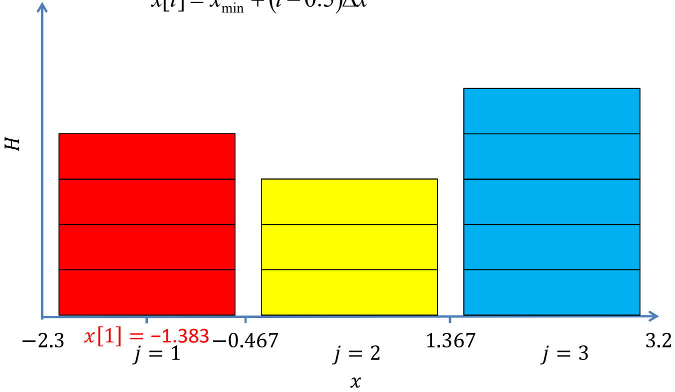
$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$

$$H[1] = \frac{4.}{12.* 1.83333}$$

$$H[2] = ?$$

$$x[1] = ?$$



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

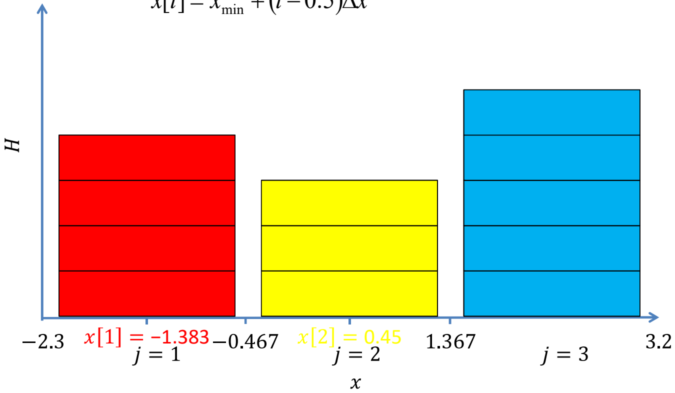
$$H[bin] = H[bin] / (M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$

$$H[1] = \frac{4.}{12.* 1.83333}$$

$$H[2] = ?$$

$$x[1] = ?$$



Array of random numbers

1.5
3.2
-0.3
0.6
-2.3
-1.5
-0.7
1.2
2.1
2.2
1.7
-0.5

M=12

Normalize:

$$M = \sum_{i=1}^{N_{bin}} H[i]$$

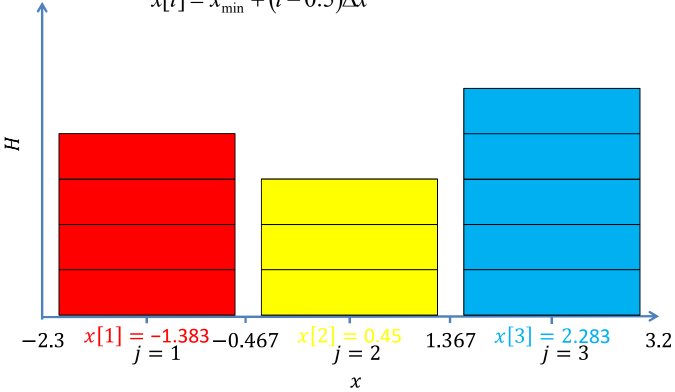
$$H[bin] = H[bin]/(M\Delta x)$$

$$x[i] = x_{\min} + (i - 0.5)\Delta x$$

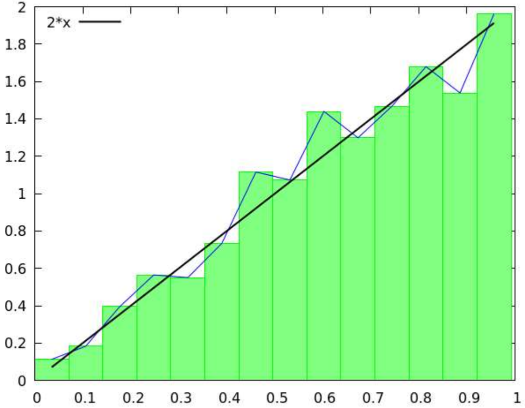
$$H[1] = \frac{4.}{12.* 1.83333}$$

$H[2] = ?$

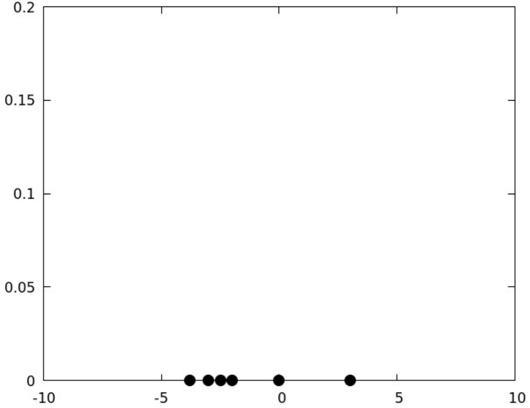
$x[1] = ?$



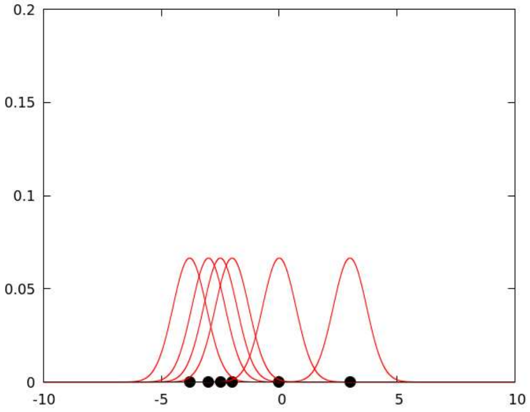
Histogram



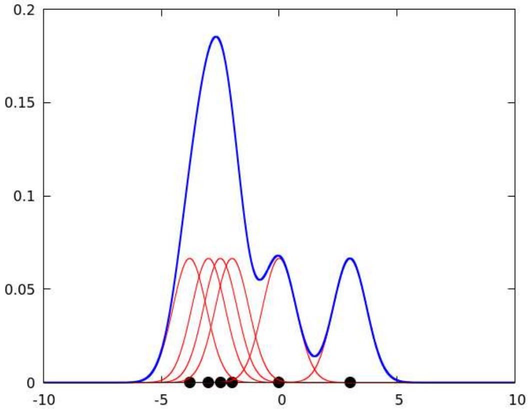
Kernel Density Estimation



Kernel Density Estimation



Kernel Density Estimation



Kernel Density Estimation

- $p(x) = \frac{1}{M}\sum_{i=1}^{M} K(x, s, x_i)$
- If the kernel is Gaussian, $K(x, s, x_i) = \mathcal{N}(x, s, x_i) = \frac{1}{s\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-x_i}{s}\right)^2}$
- s is the smoothing parameter, not easy to choose, as rule of thumb we will use:

$$s = \frac{0.9A}{M^{1/5}}, A = \min\left(\sigma, \frac{IQR}{1.34}\right)$$

Kernel density estimation

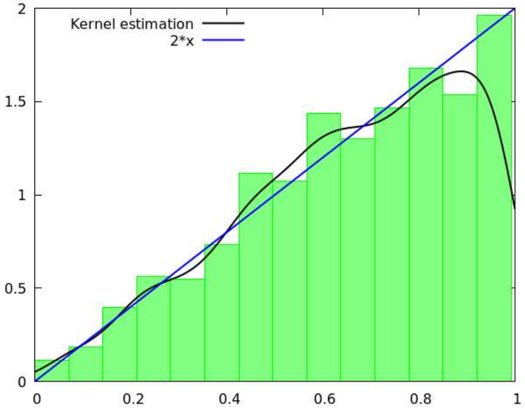
- We obtain a function defined as

$$p(x) = \frac{1}{M}\sum_{i=1}^{M} K(x, s, x_i)$$

- Remember, in computational terms, a function corresponds to a table!

```
-1.           p(-1)
-0.999        p(-0.999)
….            …
0.999         p(0.999)
1.            p(1.)
```

Kernel Density Estimation



Assignment-15

1) Generates 10^4 points (x_i) in the interval $x \in [-10, 10]$ from the dist. ($\mathcal{N}(x, s, x_i)$ is the normal dist. with mean x_i and std. dev. s)
$f(x) = 15x^2\mathcal{N}(x, 0.25, -0.5) + 13\mathcal{N}(x, 0.3, -1.5) + 7\mathcal{N}(x, 1.0, 3.0)$
using the rejection method and compute -

- The empirical cumulative distribution function.

- The histogram representation using the Freedman-Diaconis rule.

- The value of the Gaussian kernel density estimation $(p(x))$ using the rule of thumb for the smoothing. In this case, build a table with 100000 entries for the x between -10 and 10 (Note this 10000 is not related with the size of the sample that is still 10000).

Submit your code as ***surname_assignment_15.f90*** to ***dbhakuni@ictp.it*** before next Thursday.

Function assignment

Use f_{max}=17.5