

Statistics library

Michael V Koldayev

November 7, 2015

In this report I would like to show the statistics library that we were creating in class. The program consists of functions and all of them are related to each other. At the end we can see as the output as neat diagram with Gauss distribution which widely used in different test results.

PYTHON CODE

```
1 import random #import library that generates random numbers
2 from graphics import* #import all the graphics from graphics.py program in order to draw.
3
4 def zeros(n): #function with parameter n
5     a=[] #empty array
6     for i in range(n): #creates for in loop which goes until function's parameter value is reached
7         a+=[] #creates an array till for in loop runs
8     return a #returns an array that is fullfilled with empty arrays
9
10 def sum_array(a): #function that will calculate sum of array
11     s=0 #initial sum value is equal to 0
12     for i in range(len(a)): #creates for in loop which loops the same amount of time as length of array - a
13         s+=a[i] #each time it loops, program adds a number to the sum until the length of array is reached.
14     return s #returns the sum value
15
16 def rand_array(n,mini,maxi): #function that fulfills array with random values
17     a=[] #empty array
18     for i in range(n): #for in loop that runs until reaches n
19         a+=[] #append random number between minimum and maximum values
20     return a #returns array
21
22 def avg(a): #function that calculates and average of an array
23     return sum_array(a)/len(a) #program takes sum of an array a and divides it by the amount of numbers stored in that array.
24
25 def var(a): #function that calculates variance
26     s=0 #initial value is 0
27     for i in range(len(a)): #loop until reaches the last value of array a
28         s+=a[i]**2 #
29     m=avg(a) #m is equal to an average number of an array a
30     return (s/len(a)-m**2) #return sum divided by the squared average value of an array a
31
32 def maximum(a): #function look for the maximum value in the array
33     m=0 #initially, maximum value is 0
34     for i in range(len(a)): #we want to run this program till we reach the last number of an array a
35         if a[i]>m: #checks each value in the array whether it is bigger than number m
36             m=a[i] #if it is bigger, then it changes current m value on a new, bigger value from an array a
37     return m #by the end, it returns maximum value
38
39 def histogram(mini,maxi,bins,a): #histogram function includes min, max values, number of bins, and array a
40     h=zeros(bins) #run function zeros with parameter bins
41     w=(maxi-mini)/bins #create new variable which is equal to max value minus min value dived by number of bins
42     for i in range(len(a)): #loop in range of length of array a
43         for j in range(bins): #loop in range of number of bins
44             if a[i]>(mini+j*w) and a[i]<(mini+(j+1)*w): #if statement that checks whether we should add more 45
45                 h[j]=h[j]+1 #if it is true add 1 to it
46     return h #return h
```



Figure 1: Wonderful Kitty))

```

48
49def bargraph(a, window): #function that draws
50     win=GraphWin("BarGraph", 500, 500) #Graph with name "BarGraph", width 500, height 500
51     window.setCoords(-1,-1,len(a)+1, maximum(a)+1) #set coordinates of window bigger than current values of a.
52     for i in range(len(a)): #for in loops runs until the last number of array a is reached.
53         rec=Rectangle(Point(i,0), Point(i+1,a[i])) #Draw rectangle that is equal to our array's values
54         rec.draw(window) #draw window
55
56def main(): main function that includes all other functions
57     gauss=zeros(1000) #amount of numbers we want to check
58     for i in range(len(gauss)): #run loop with the amount we
59         gauss[i]=sum_array(rand_array(10, 0, 1)) #gauss number is equal to sum of 10 random arrays
60     histo=histogram(0., 10., 1700, gauss) #show 1700 values between 0-10
61     win=GraphWin("BarGraph'",500,500) #creates a windows
62     bargraph(histo,win) #draws histogram inside the window
63
64main() #calls main function in runs everything we wrote down.

```

Output:



The window with this output appeared. As you clearly see, it is a Gauss distribution