Introduction to Statistics

Authors : Bruce Carpenter, Bill Davis, Michael Raschke and Jerry Uhl
Publisher : Math Everywhere, Inc. **Distributor & Translator:** MathMonkeys, LLC
Adapted from Prob/Stat by : Robert Curtis.

STAT.01 Simulations

Tutorials T1

Experience with the starred problems will be useful for understanding developme

Graphics Primitives

The variables (x, s, t, z, y) are independent of each other.

T.1) Monte Carlo estimation of area measurements

John von Neumann and Stanislaw Ulam were great mathematicians of the early to midtwentieth century.
Working together on the original atomic bomb in the 1940's, they devised the Monte Carlo method for doing approximate calculation for problems intractable by hand.
You'll see many Monte Carlo simulations in this course.

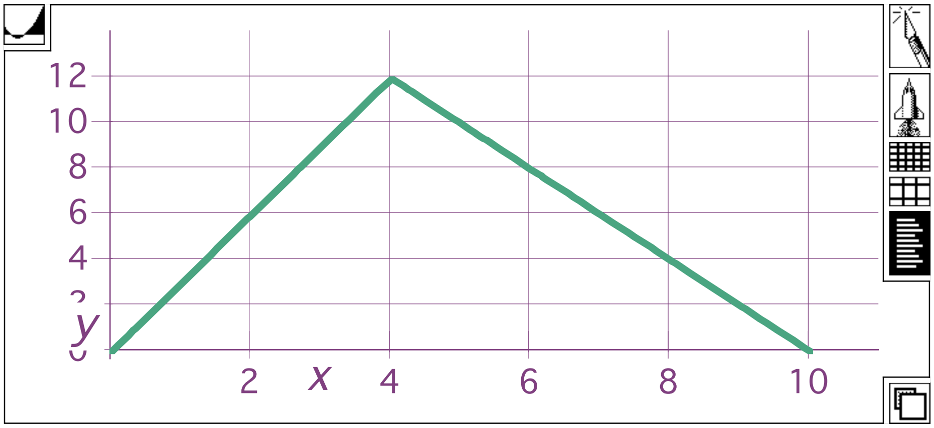
T.1.a) Monte Carlo estimation of the area of geometric objects

Here's a nice triangle, defined via the function f(x) and the x-axis below, whose area we can calculate easily using our geometric area formulas from high school geometry.

xlow = 0
xhigh = 10

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

$y = f(x)$



0 ... 11 = left...right Stretch to Fit

−0.4 ... 14 = bottom...top cropping Moderately

Graph Building Blocks

Curve at $(x, f[x])$ where x = xlow ... xhigh

with a extra heavy line, colored Other... .

Let A be the area of this triangle, computed by via A= 1/2 * base

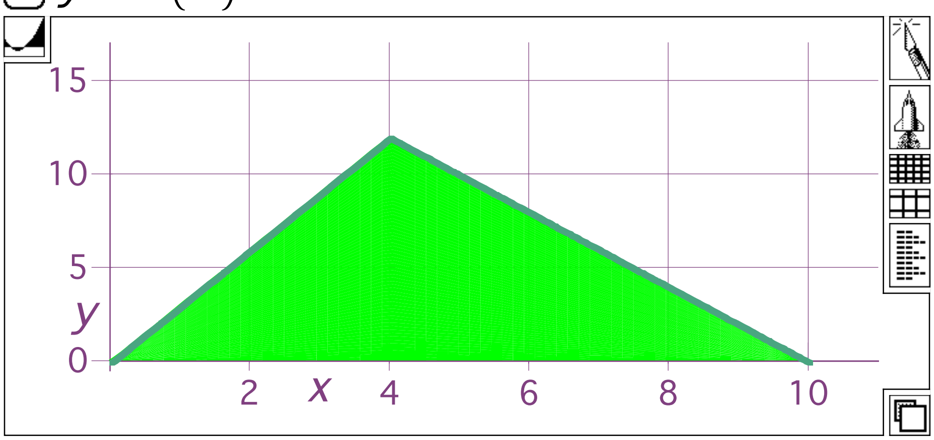
A = 1/2 * (xhigh - xlow) * f(4) = 1/2 * (10) * 12 = 60

xlow = 0

xhigh = 10

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \ge 4) \end{cases}$$

$y = f(x)$



Use uniformly distributed points and the Monte Carlo method to co

the measurement of the area between the plotted curve and the x

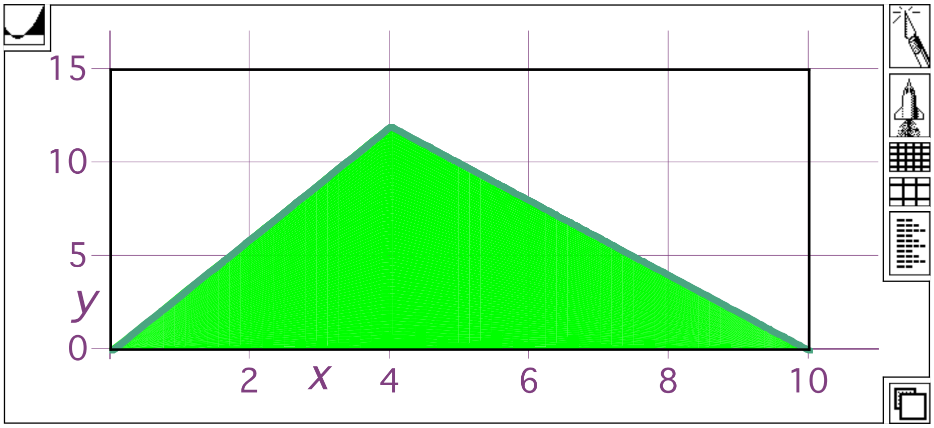
Answer

Put a box around everything:

xlow = 0 xhigh = 10
ylow = 0 yhigh = 15

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

$y = f(x)$



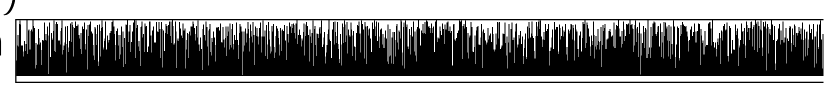
Your box does not have to touch everything in the area under measureme
But, the area under measurement must be inside your box.

Generate 3000 uniformly distributed random points inside the bo

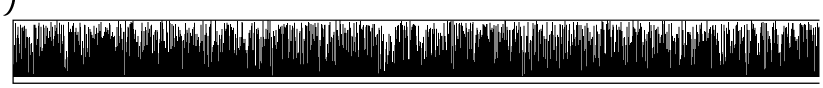
xlow = 0 xhigh = 10
ylow = 0 yhigh = 15

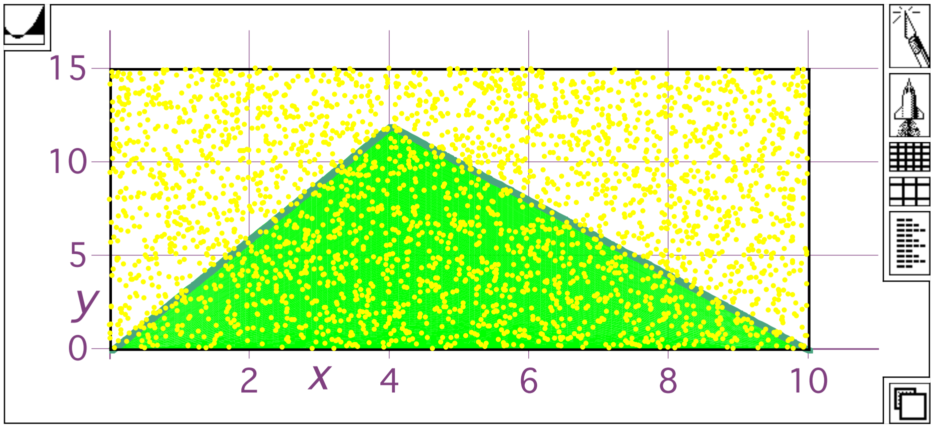
$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

x = Random(xlow , xhigh)

Tabulate xRanNums with 

y = Random(ylow , yhigh)

Tabulate yRanNums with 



Break the random points {x, y} into two separate groups:

1) pointsunder = those points {x, y} under or on the curve

and

2) pointsover = those points {x, y} over the curve

xlow = 0 xhigh = 10

ylow = 0 yhigh = 15

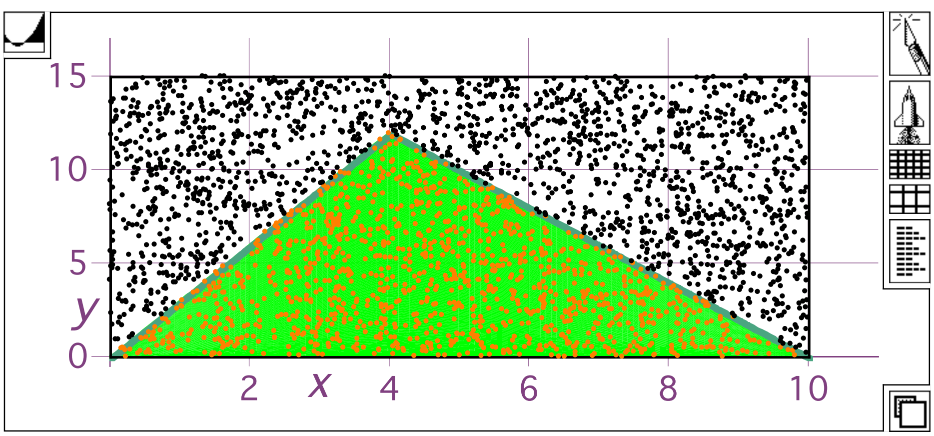
$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

x = Random(xlow , xhigh)

Tabulate xRanNums with

y = Random(ylow , yhigh)

Tabulate yRanNums with



pointsunder consists of the points {x, y} for which y ≤ f[x].

pointsover consists of the points {x, y} for which y > f[x].

The Monte carlo idea:

Because the random points are approximately uniformly distribut

$$\frac{A}{\text{Area enclosed by the box}} \approx \frac{\text{Number of random points under curv}}{\text{Total number of random points inside b}}$$

so that:

$$A \approx \left(\frac{\text{Number of random points under curve}}{\text{Total number of random points inside box}}\right)(\text{Area}$$

Try it out:

$\text{xlow} = 0 \quad \text{xhigh} = 10$

$\text{ylow} = 0 \quad \text{yhigh} = 15$

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

$x = \text{Random}(\text{xlow}, \text{xhigh})$

Tabulate xRanNums with

$y = \text{Random}(\text{ylow}, \text{yhigh})$

Tabulate yRanNums with

$$\text{UnderCount} = \sum_{k=1}^{3000} \left(\text{yRanNums}[k] \leq f[\text{xRanNums}\{k\}]\right)$$

$\text{UnderCount} = 1184$ *Calculate*

$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$

$\text{BoxArea} = 150$ *Calculate*

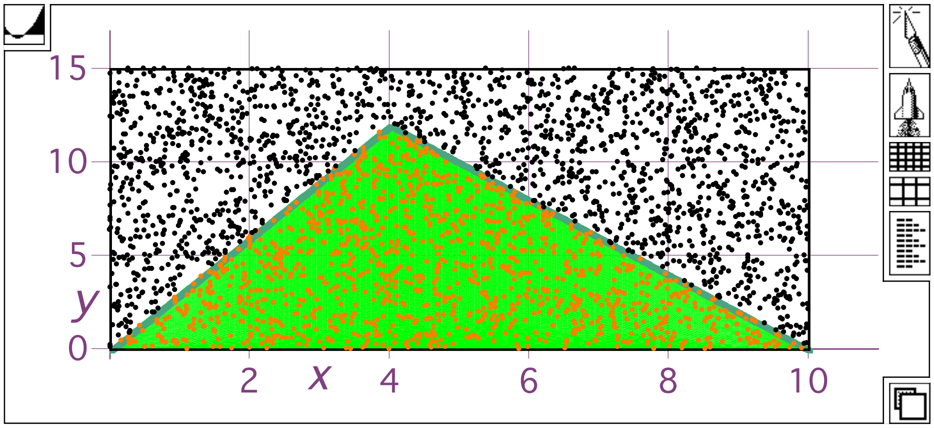
$$\frac{\text{UnderCount}}{3000}\,\text{BoxArea}$$

$\frac{\text{UnderCount}}{3000}\,\text{BoxArea} = \frac{296}{5}$ *Substitute*

$\frac{\text{UnderCount}}{3000}\,\text{BoxArea} = 59.2$ *Calculate*

$\frac{\text{UnderCount}}{3000}\,\text{BoxArea} = \frac{296}{5}$ *Substitute*

$\frac{\text{UnderCount}}{3000}\,\text{BoxArea} = 59.2$ *Calculate*



Run 9 more Monte Carlo experiments, each involving 1000 indep

xlow = 0   xhigh = 10

ylow = 0   yhigh = 15

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

x = Random(xlow , xhigh)

Tabulate $xRanNums_1$ with

Tabulate $xRanNums_2$ with

Tabulate $xRanNums_3$ with

Tabulate $xRanNums_4$ with

Tabulate $xRanNums_5$ with

Tabulate $xRanNums_6$ with

Tabulate $xRanNums_7$ with

Tabulate $xRanNums_8$ with

Tabulate $xRanNums_9$ with

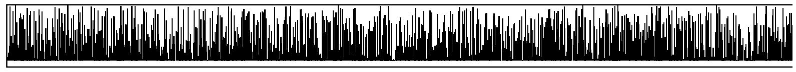
y = Random(ylow , yhigh)

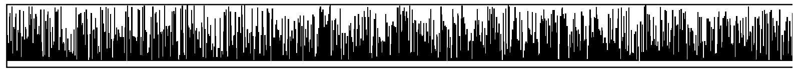
Tabulate $yRanNums_1$ with

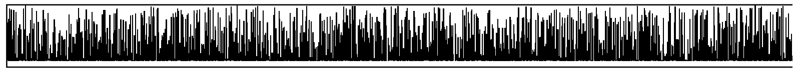
Tabulate $yRanNums_2$ with

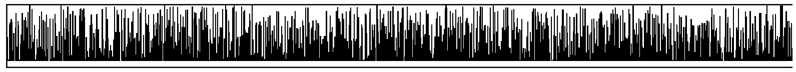
Tabulate $yRanNums_3$ with

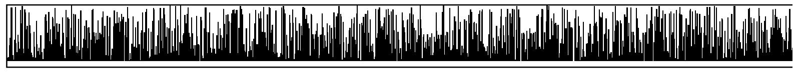
Tabulate $yRanNums_4$ with

Tabulate $yRanNums_5$ with 

Tabulate $yRanNums_6$ with 

Tabulate $yRanNums_7$ with 

Tabulate $yRanNums_8$ with 

Tabulate $yRanNums_9$ with 

$$UnderCount_1 = \sum_{k=1}^{1000} \left(yRanNums_1[k] \le f[xRanNums_1\{k\}] \right)$$

$UnderCount_1 = 413$ *Calculate*

$$UnderCount_2 = \sum_{k=1}^{1000} \left(yRanNums_2[k] \le f[xRanNums_2\{k\}] \right)$$

$UnderCount_2 = 401$ *Calculate*

$$UnderCount_3 = \sum_{k=1}^{1000} \left(yRanNums_3[k] \le f[xRanNums_3\{k\}] \right)$$

$UnderCount_3 = 418$ *Calculate*

$$UnderCount_4 = \sum_{k=1}^{1000} \left(yRanNums_4[k] \le f[xRanNums_4\{k\}] \right)$$

$UnderCount_4 = 390$ *Calculate*

$$UnderCount_5 = \sum_{k=1}^{1000} \left(yRanNums_5[k] \le f[xRanNums_5\{k\}] \right)$$

$UnderCount_5 = 410$ *Calculate*

$$UnderCount_6 = \sum_{k=1}^{1000} \left(yRanNums_6[k] \le f[xRanNums_6\{k\}] \right)$$

$UnderCount_6 = 409$ *Calculate*

$$UnderCount_7 = \sum_{k=1}^{1000} \left(yRanNums_7[k] \le f[xRanNums_7\{k\}] \right)$$

$UnderCount_7 = 387$ *Calculate*

$$UnderCount_8 = \sum_{k=1}^{1000} \left(yRanNums_8[k] \le f[xRanNums_8\{k\}] \right)$$

$UnderCount_8 = 405$ *Calculate*

$$\text{UnderCount}_9 = \sum_{k=1}^{1000}\left(\text{yRanNums}_9[k] \le f[\text{xRanNums}_9\{k\}]\right)$$

$\text{UnderCount}_9 = 400$ *Calculate*

$$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$$

$\text{BoxArea} = 150$ *Calculate*

$$\text{Estimate}_1 = \frac{\text{UnderCount}_1}{1000}\,\text{BoxArea}$$

$\text{Estimate}_1 = 61.95$ *Calculate*

$$\text{Estimate}_2 = \frac{\text{UnderCount}_2}{1000}\,\text{BoxArea}$$

$\text{Estimate}_2 = 60.15$ *Calculate*

$$\text{Estimate}_3 = \frac{\text{UnderCount}_3}{1000}\,\text{BoxArea}$$

$\text{Estimate}_3 = 62.7$ *Calculate*

$$\text{Estimate}_4 = \frac{\text{UnderCount}_4}{1000}\,\text{BoxArea}$$

$\text{Estimate}_4 = 58.5$ *Calculate*

$$\text{Estimate}_5 = \frac{\text{UnderCount}_5}{1000}\,\text{BoxArea}$$

$\text{Estimate}_5 = 61.5$ *Calculate*

$$\text{Estimate}_6 = \frac{\text{UnderCount}_6}{1000}\,\text{BoxArea}$$

$\text{Estimate}_6 = 61.35$ *Calculate*

$$\text{Estimate}_7 = \frac{\text{UnderCount}_7}{1000}\,\text{BoxArea}$$

$\text{Estimate}_7 = 58.05$ *Calculate*

$$\text{Estimate}_8 = \frac{\text{UnderCount}_8}{1000}\,\text{BoxArea}$$

$\text{Estimate}_8 = 60.75$ *Calculate*

$$\text{Estimate}_9 = \frac{\text{UnderCount}_9}{1000}\,\text{BoxArea}$$

$\text{Estimate}_9 = 60$ *Calculate*

Look at the average of the 10 Monte Carlo estimates:

$$\text{Average} = \frac{\sum_{k=1}^{9} \text{Estimate}_k}{9}$$

$$\text{Average} = \tfrac{1}{9}\,\text{Estimate}_9 + \tfrac{1}{9}\,\text{Estimate}_8 + \tfrac{1}{9}\,\text{Estimate}_7 + \tfrac{1}{9}\,\text{Estimate}$$

$$\text{Average} = 60.55$$ *Calculate*

Compare with the actual value of A
as calculated by LiveMath:

xlow = 0 xhigh = 10
ylow = 0 yhigh = 8.5

$$f(x) = \begin{cases} 3x & (x < 4) \\ -2x + 20 & (x \geq 4) \end{cases}$$

$$A = \tfrac{1}{2}(\text{xhigh} - \text{xlow})\, f(4)$$

$$A = 60$$ *Calculate*

The Law of Large numbers says:

If your experience was typical, then the average of these Monte C
estimates is a close enough approximation of A
for most government work.

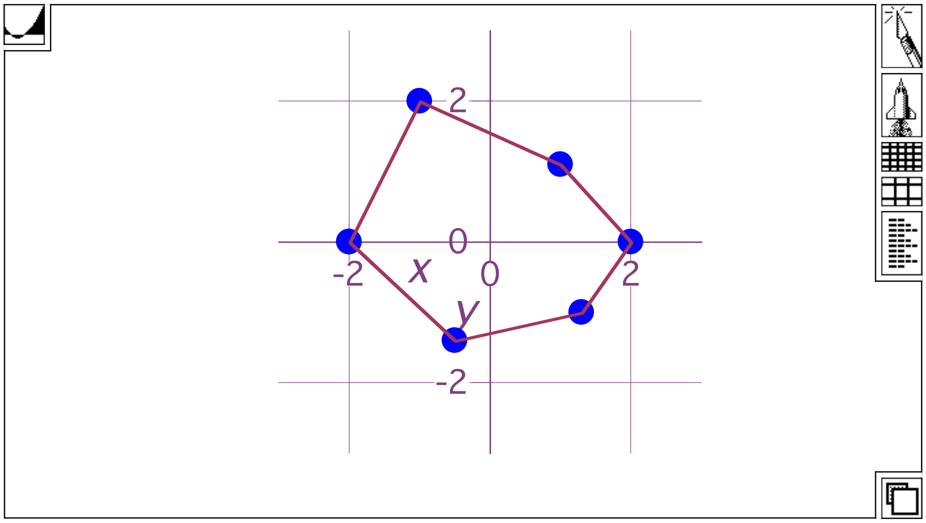
T.1.b) Monte Carlo estimation of another area measurement

Here is a non-regular hexagon. Non-regular refers to the fact that the sides of this hexagon are not equal. A Bee would never made such an ugly honeycomb because they would not stack nicely together.

We can find the area of this non-regular hexagon by slicing up the geometric region into triangles, each of which we can compute the area of easily.

xlow = -2
xhigh = 2

$$V = ([-2,0],[-1,2],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$$



Use uniformly distributed points and the Monte Carlo method to estimate the measurement of the area enclosed by the non-regu

Answer

First we need to measure the area of this non-regular hexagon using basic geometric formulas. The easiest way to accumulate this area is by splitting up this non-regular hexagon into sub-triangles and sub-trapezoids. Remember these basic area formulas from high school geometry

$$\text{AreaTriangle} = \frac{1}{2}(b\,h)$$

$$\text{AreaTrapezoid} = \frac{1}{2}b(h_1 + h_2)$$

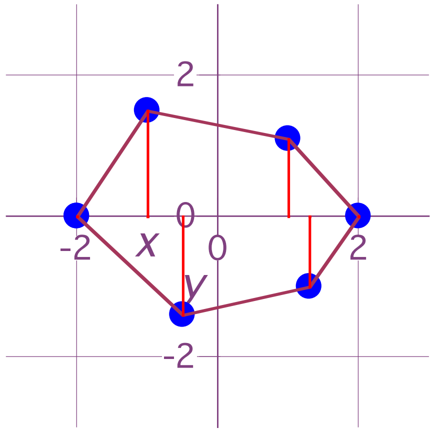
h_1 and h_2 are the heights of each side of the trapezoid

Here is the split up of this non-regular hexagon:

$$\text{xlow} = -2$$

$$\text{xhigh} = 2$$

$$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$$



We manually read off the area formulas for these 6 geometric regions:

1st Triangle

$b = 1$

$h = 1.5$

$A_1 = \frac{1}{2} b\, h$

$A_1 = 0.75$ *Substitute*

2nd Trapezoid

$b = 2$

$h_1 = 1.5$

$h_2 = 1.1$

$A_2 = \frac{1}{2} b\left(h_1 + h_2\right)$

$A_2 = 2.6$ *Substitute*

3rd Triangle

$b = 1$

$h = 1.1$

$A_3 = \frac{1}{2} b\, h$

$A_3 = 0.55$ *Substitute*

4th Triangle

$b = 0.7$

$h = 1$

$A_4 = \frac{1}{2} b\, h$

$A_4 = 0.35$ *Substitute*

5th Trapezoid

$b = 1.3 - (-0.5)$

$h_1 = 1.4$

$h_2 = 1$

$A_5 = \frac{1}{2} b\left(h_1 + h_2\right)$

$A_5 = 2.16$ *Substitute*

6th Triangle

$b = 1.5$

$h = 1.4$

$A_6 = \frac{1}{2} b\, h$

$A_6 = 1.05$ *Substitute*

And then copy/paste those answers into this Case Theory below

$A_1 = 0.75$

$A_2 = 2.6$

$A_3 = 0.55$

$A_4 = 0.35$

$A_5 = 2.16$

$A_6 = 1.05$

Upper Hexagon region

$\sum_{n=1}^{3} A_n$

$\sum_{n=1}^{3} A_n = A_3 + A_2 + A_1$ *Expand*

$\sum_{n=1}^{3} A_n = 3.9$ *Calculate*

Lower Hexagon region

$$\sum_{n=4}^{6} A_n$$

$$\sum_{n=4}^{6} A_n = A_6 + A_5 + A_4 \quad \textit{Expand}$$

$$\sum_{n=4}^{6} A_n = 3.56 \quad \textit{Calculate}$$

Full Hexagon

$$\sum_{n=1}^{6} A_n$$

$$\sum_{n=1}^{6} A_n = A_6 + A_5 + A_4 + A_3 + A_2 + A_1 \quad \textit{Expand}$$

$$\sum_{n=1}^{6} A_n = 7.46 \quad \textit{Calculate}$$

Now, we will use the Monte Carlo Method to try to see if we get near this answer using the Probabilistic method for computing area. First we will deal with the upper part of the hexagon, and we need to get a function f(x) made up of linear segments as the "top hat" of this hexagon. Finding the linear equations involved is easy using the Point-Slope form of a line:

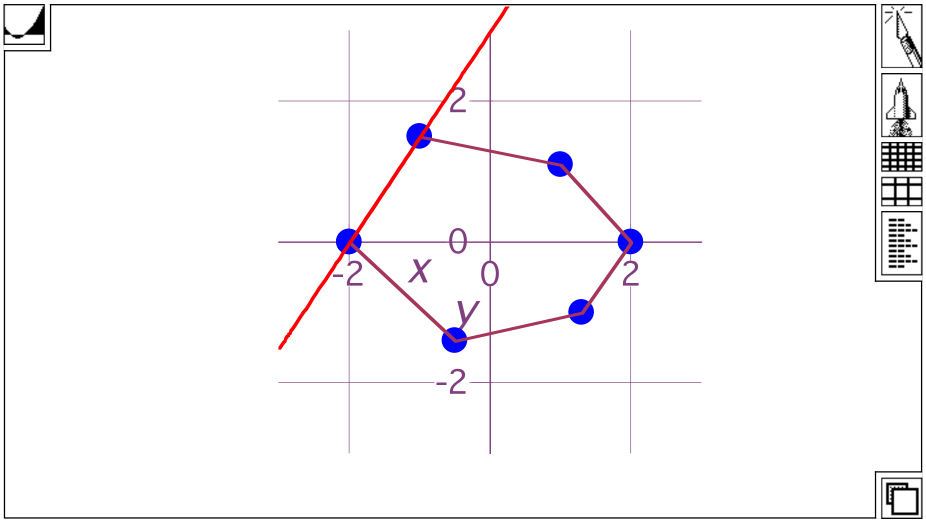
xlow = −2

xhigh = 2

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red



Getting the other line equations in the same way is easy, too:

$xlow = -2$

$xhigh = 2$

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

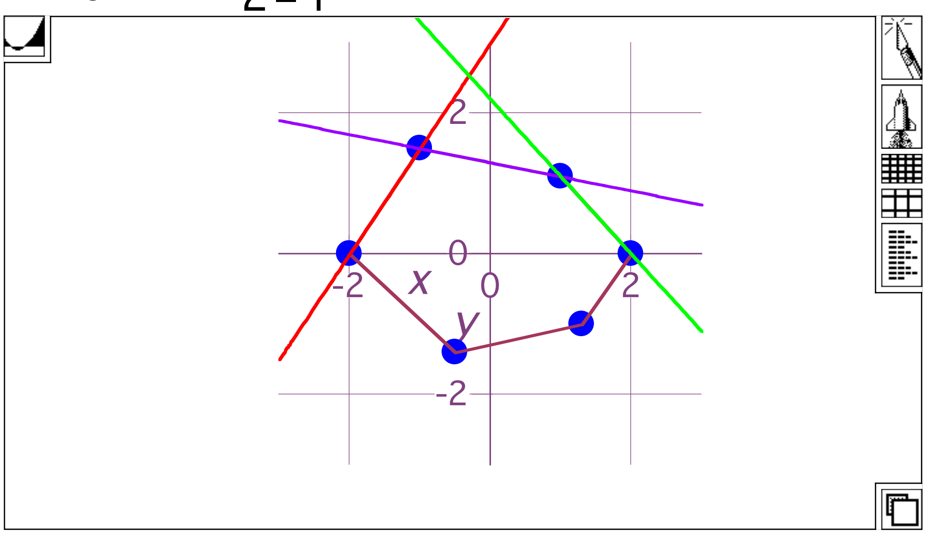
$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

$f_2(x) = \frac{1.1-1.5}{1-(-1)}(x-1)+1.1$ Purple

3rd line segment

$f_3(x) = \frac{0-1.1}{2-1}(x-2)+0$ Green



Then put these 3 linear equations into a piecewise-defined function to "glue them together"

$xlow = -2$

$xhigh = 2$

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \dfrac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

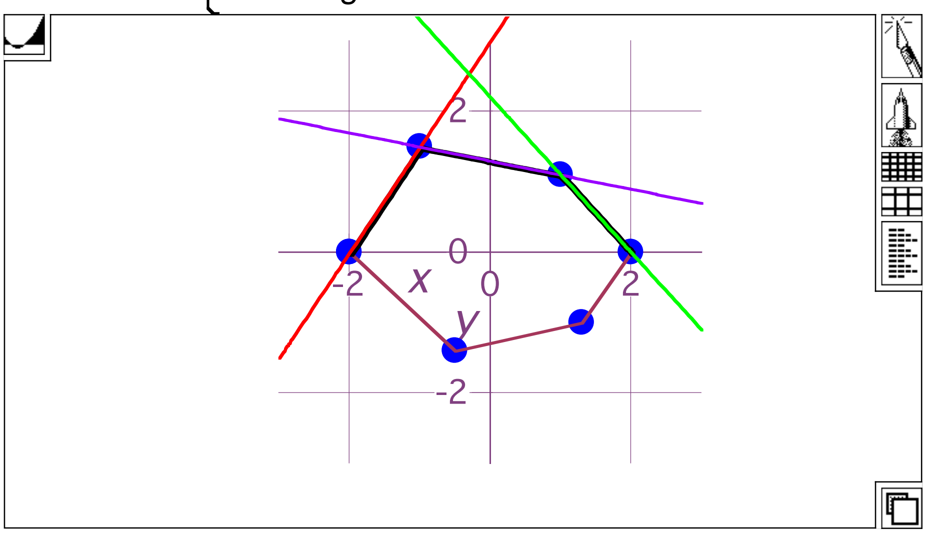
$f_2(x) = \dfrac{1.1-1.5}{1-(-1)}(x-1)+1.1$ Purple

3rd line segment

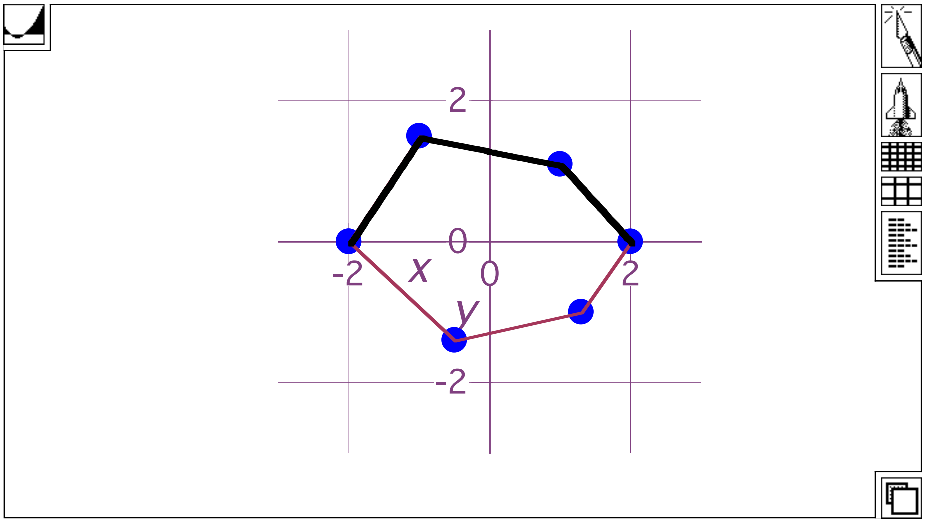
$f_3(x) = \dfrac{0-1.1}{2-1}(x-2)+0$ Green

Piecewise-Defined Glued function

$$f(x) = \begin{cases} f_1(x) & (-2 < x)(x < -1) \\ f_2(x) & (-1 < x)(x < 1) \\ f_3(x) & (1 < x)(x < 2) \end{cases}$$



With original line segment functions removed



Now that we have a "top hat" function, we can proceed with the Monte Carlo Method estimation for the area of this region. First step, put a big box around the entire upper hexagon region.

$xlow = -2$ $xhigh = 2$

$ylow = 0$ $yhigh = 2$

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

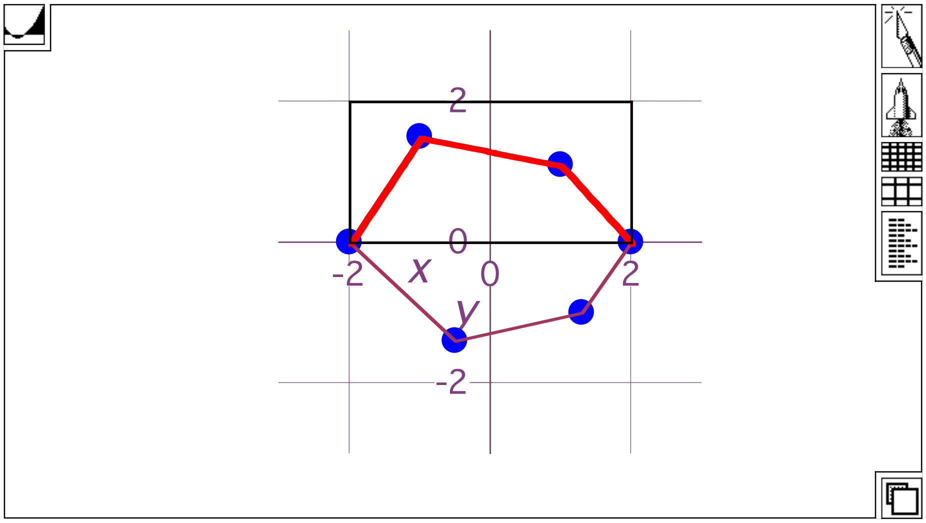
$f_2(x) = \frac{1.1-1.5}{1-(-1)}(x-1)+1.1$ Purple

3rd line segment

$f_3(x) = \frac{0-1.1}{2-1}(x-2)+0$ Green

Piecewise-Defined Glued function

$$f(x) = \begin{cases} f_1(x) & (-2 < x)(x < -1) \\ f_2(x) & (-1 < x)(x < 1) \\ f_3(x) & (1 < x)(x < 2) \end{cases}$$



Generate 2000 uniformly distributed random points inside the bo

xlow = −2 xhigh = 2

ylow = 0 yhigh = 2

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

$f_2(x) = \frac{1.1-1.5}{1-(-1)}(x-1)+1.1$ Purple

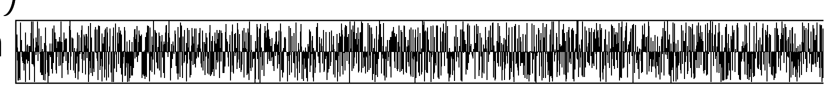
3rd line segment

$f_3(x) = \frac{0-1.1}{2-1}(x-2)+0$ Green

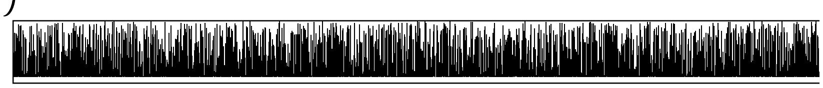
Piecewise-Defined Glued function

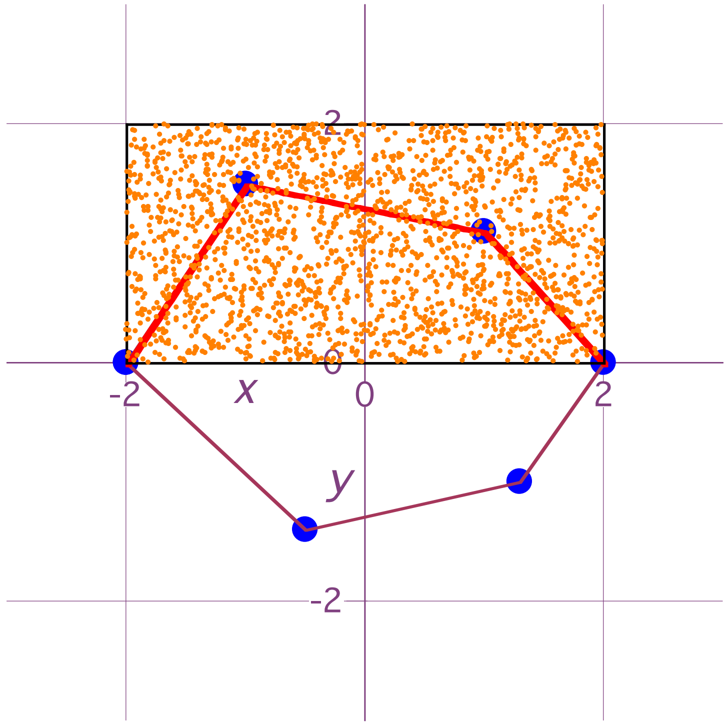
$$f(x) = \begin{cases} f_1(x) & (-2 < x)(x < -1) \\ f_2(x) & (-1 < x)(x < 1) \\ f_3(x) & (1 < x)(x < 2) \end{cases}$$

x = Random(xlow , xhigh)

Tabulate xRanNums with 

y = Random(ylow , yhigh)

Tabulate yRanNums with 



The Monte carlo idea:

Because the points are approximately uniformly distributed, you

$$\frac{\text{Area enclosed by curve}}{\text{Area enclosed by the box}} \approx \frac{\text{Number of random points inside curv}}{\text{Total number of random points inside b}}$$

so that:

$$\text{Area enclosed by curve} \approx \left(\frac{\text{Number of random points inside curve}}{\text{Total number of random points inside box}}\right) * \text{Area enc}$$

Try it out:

$xlow = -2$ $xhigh = 2$

$ylow = 0$ $yhigh = 2$

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

$f_2(x) = \frac{1.1 - 1.5}{1 - (-1)}(x - 1) + 1.1$ Purple

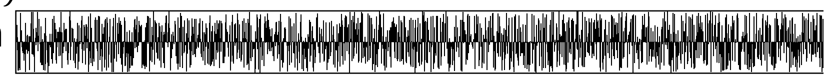
3rd line segment

$f_3(x) = \frac{0 - 1.1}{2 - 1}(x - 2) + 0$ Green

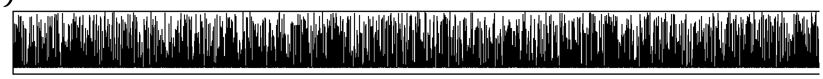
Piecewise-Defined Glued function

$$f(x) = \begin{cases} f_1(x) & (-2 < x)(x < -1) \\ f_2(x) & (-1 < x)(x < 1) \\ f_3(x) & (1 < x)(x < 2) \end{cases}$$

$x = \text{Random}(\text{xlow}, \text{xhigh})$

Tabulate xRanNums with 

$y = \text{Random}(\text{ylow}, \text{yhigh})$

Tabulate yRanNums with 

$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$

$\text{BoxArea} = 8$ *Calculate*

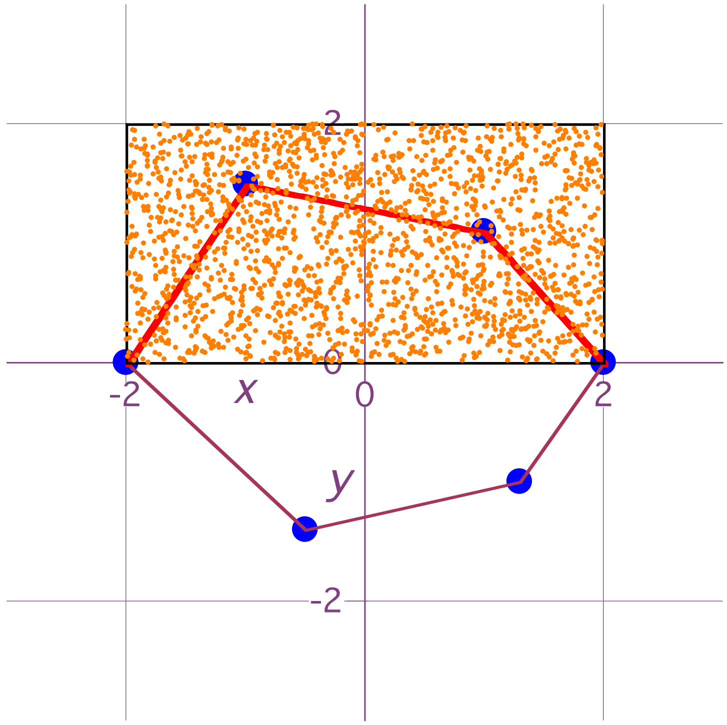
$$\text{Count} = \sum_{k=1}^{2000} (\text{yRanNums}[k] \leq f[\text{xRanNums}\{k\}])$$

$\text{Count} = 968$ *Calculate*

$$\text{AreaEstimate} = \frac{\text{Count}}{2000}\text{BoxArea}$$

$\text{AreaEstimate} = \frac{484}{125}$ *Substitute*

$\text{AreaEstimate} = 3.872$ *Calculate*



That is very good! The area of this upper region of the hexagon was measured previously at 3.9.

Now, average to try to get a better estimate:

xlow = −2 xhigh = 2

ylow = 0 yhigh = 2

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

1st line segment

$f_1(x) = \frac{1.5-0}{-1-(-2)}(x-[-1])+1.5$ Red

2nd line segment

$f_2(x) = \frac{1.1-1.5}{1-(-1)}(x-1)+1.1$ Purple

3rd line segment

$f_3(x) = \frac{0-1.1}{2-1}(x-2)+0$ Green

Piecewise-Defined Glued function

$$f(x) = \begin{cases} f_1(x) & (-2 < x)(x < -1) \\ f_2(x) & (-1 < x)(x < 1) \\ f_3(x) & (1 < x)(x < 2) \end{cases}$$

LiveMath Note: Making 20 Tabulates can get pretty tiring. So here in this example we do the same computations, but with some functional magic. Everytime these functions get Calculated for each input k, the output is a random number. Notice no k on the right-hand -side, because the random number is not dependent upon the k value.

$\text{xRandoms}(k) = \text{Random}(\text{xlow}, \text{xhigh})$

$\text{yRandoms}(k) = \text{Random}(\text{ylow}, \text{yhigh})$

$\text{xRandoms}(3)$

$\text{xRandoms}(3) = 1.21859053899243$ *Calculate*

$$\text{fCounts}(n) = \sum_{k=1}^{2000} (\text{yRandoms}[k] \leq f[\text{xRandoms}\{k\}])$$

$\text{fCounts}(1)$

$\text{fCounts}(1) = 969$ *Calculate*

$\text{fCounts}(2)$

$\text{fCounts}(2) = 962$ *Calculate*

$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$

$$\text{AreaEst}(n) = \frac{\text{fCounts}(n)}{2000}\,\text{BoxArea}$$

$\text{AreaEst}(1)$

$\text{AreaEst}(1) = 4.064$ *Calculate*

$\text{AreaEst}(2)$

$\text{AreaEst}(2) = 3.808$ *Calculate*

$\text{AreaEst}(3)$

$\text{AreaEst}(3) = 3.828$ *Calculate*

$\text{AreaEst}(4)$

$\text{AreaEst}(4) = 3.944$ *Calculate*

$\text{AreaEst}(5)$

$\text{AreaEst}(5) = 3.852$ *Calculate*

$\text{AreaEst}(6)$

$\text{AreaEst}(6) = 3.8$ *Calculate*

AreaEst(7)

AreaEst(7) = 3.932 *Calculate*

AreaEst(8)

AreaEst(8) = 3.864 *Calculate*

AreaEst(9)

AreaEst(9) = 3.828 *Calculate*

AreaEst(10)

AreaEst(10) = 3.856 *Calculate*

The average of the 10 Monte Carlo estimates is:

$$\text{Average} = \frac{\sum_{k=1}^{10} \text{AreaEst}(k)}{10}$$

Average = 3.924 *Calculate*

That is even better! The area of this upper region of the hexagon was measured previously at 3.9.

We can perform the analogous computations for the lower half of the hexagon:

xlow = −2 xhigh = 2

Note: the ylow and yhigh change since we are doing the lower part of the hexagon

ylow = −2 yhigh = 0

$V = ([-2,0],[-1,1.5],[1,1.1],[2,0],[1.3,-1],[-0.5,-1.4],[-2,0])$

4th line segment

$f_4(x) = \frac{-1-0}{1.3-2}(x-1.3)-1$ Red

5th line segment

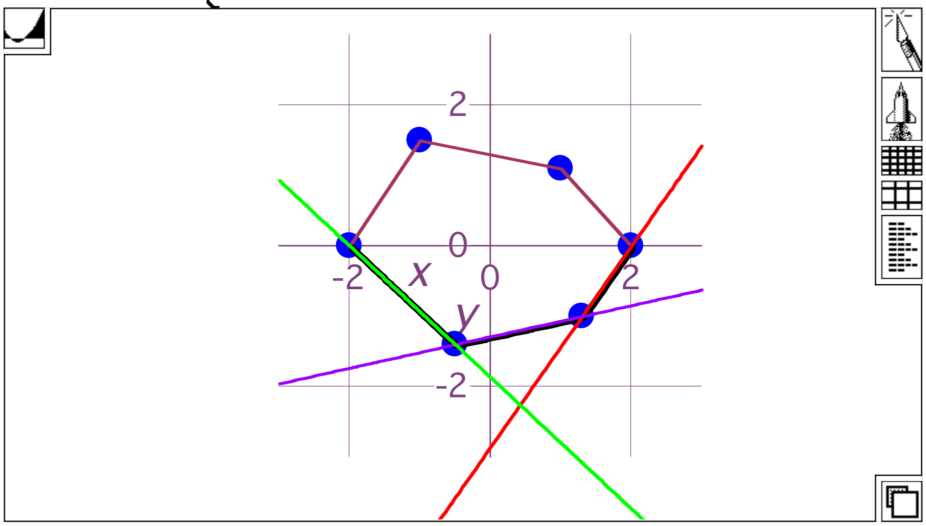
$f_5(x) = \frac{-1.4-(-1)}{-0.5-1.3}(x-[-0.5])-1.4$ Purple

6th line segment

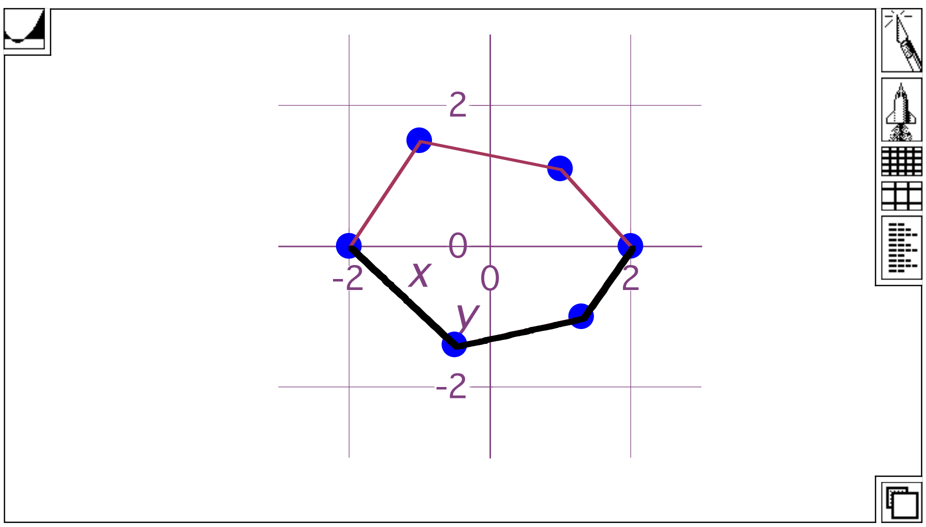
$f_6(x) = \frac{0-(-1.4)}{-2-(-0.5)}(x-[-2])+0$ Green

Piecewise-Defined Glued function

$$f(x) = \begin{cases} f_4(x) & (1.3 < x)(x < 2) \\ f_5(x) & (-0.5 < x)(x < 1.3) \\ f_6(x) & (-2 < x)(x < -0.5) \end{cases}$$



With original line segment functions removed



LiveMath Note: Making 20 Tabulates can get pretty tiring. So here in this example we do the same computations, but with some functional magic. Everytime these functions get Calculated for each input k, the output is a random number. Notice no k on the right-hand -side, because the random number is not dependent upon the k value.

$\text{xRandoms}(k) = \text{Random}(\text{xlow}, \text{xhigh})$

$\text{yRandoms}(k) = \text{Random}(\text{ylow}, \text{yhigh})$

Note: this sum set up below changes because we are below the x-axis now.

$$\text{fCounts}(n) = \sum_{k=1}^{2000} (f[\text{xRandoms}\{k\}] \le \text{yRandoms}[k])$$

$$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$$

$$\text{AreaEst}(n) = \frac{\text{fCounts}(n)}{2000}\,\text{BoxArea}$$

AreaEst(1)

AreaEst(1) = 3.404 *Calculate*

AreaEst(2)

AreaEst(2) = 3.54 *Calculate*

AreaEst(3)

AreaEst(3) = 3.488 *Calculate*

AreaEst(4)

AreaEst(4) = 3.7 *Calculate*

AreaEst(5)

AreaEst(5) = 3.584 *Calculate*

AreaEst(6)

AreaEst(6) = 3.512 *Calculate*

AreaEst(7)

AreaEst(7) = 3.572 *Calculate*

AreaEst(8)

AreaEst(8) = 3.556 *Calculate*

AreaEst(9)

AreaEst(9) = 3.544 *Calculate*

AreaEst(10)

AreaEst(10) = 3.38 *Calculate*

The average of the 10 Monte Carlo estimates is:

$$\text{Average} = \frac{\sum_{k=1}^{10} \text{AreaEst}(k)}{10}$$

Average = 3.5508 *Calculate*

That is very nice! The area of this lower region of the hexagon was measured previously at 3.56.

Because of the Law of Large Numbers, the smart money bets that the average was not all that bad.

T.1.c)

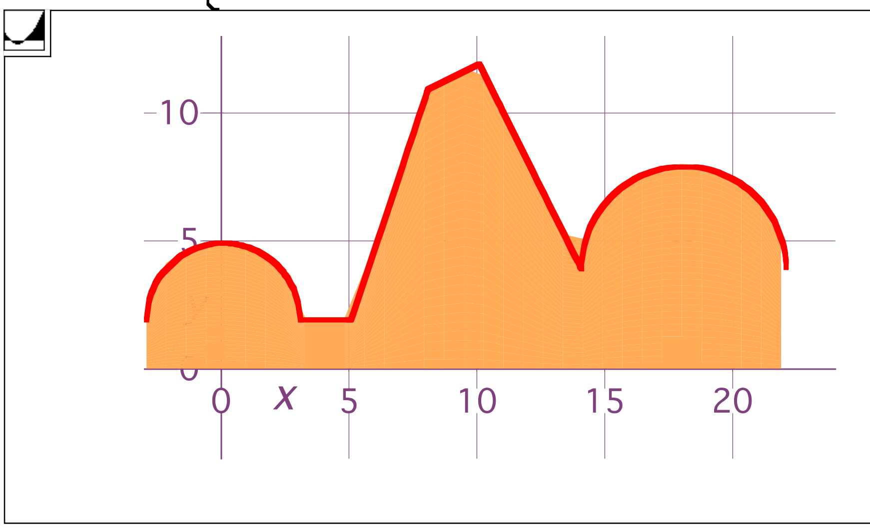
Monte Carlo estimation of another area measurement

Here's a very strange geometric region indeed:

xlow = –3

xhigh = 22

$$f(x) = \begin{cases} \sqrt{9-x^2}+2 & (-3 \le x)(x < 3) \\ 2 & (3 \le x)(x < 5) \\ 3(x-5)+2 & (5 \le x)(x < 8) \\ 0.5(x-8)+11 & (8 \le x)(x < 10) \\ (-2)(x-10)+12 & (10 \le x)(x < 14) \\ \sqrt{16-(x-18)^2}+4 & (14 \le x)(x \le 22) \end{cases}$$



We could break up this geometrical object into circles, rectangles, triangles, trapezoids, and compute the area of each subobject, and then add'em up to get the area.

Instead, let's use uniformly distributed points to estimate the meas of the area enclosed by the curve.

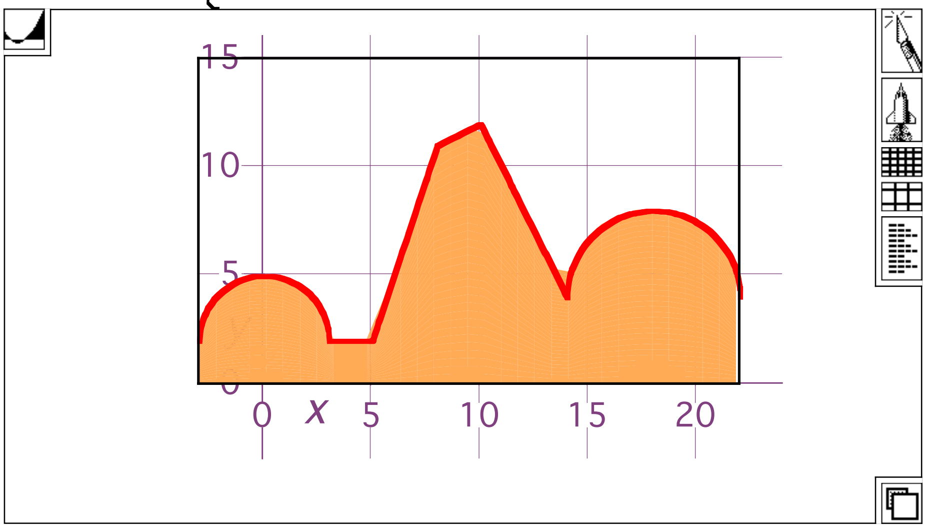
Answer

Put a box around the big geometric object:

$xlow = -3$ $xhigh = 22$

$ylow = 0$ $yhigh = 15$

$$f(x) = \begin{cases} \sqrt{9 - x^2} + 2 & (-3 \le x)(x < 3) \\ 2 & (3 \le x)(x < 5) \\ 3(x-5)+2 & (5 \le x)(x < 8) \\ 0.5(x-8)+11 & (8 \le x)(x < 10) \\ (-2)(x-10)+12 & (10 \le x)(x < 14) \\ \sqrt{16-(x-18)^2} + 4 & (14 \le x)(x \le 22) \end{cases}$$



Generate 2500 uniformly distributed random points inside the bo

$xlow = -3$ $xhigh = 22$

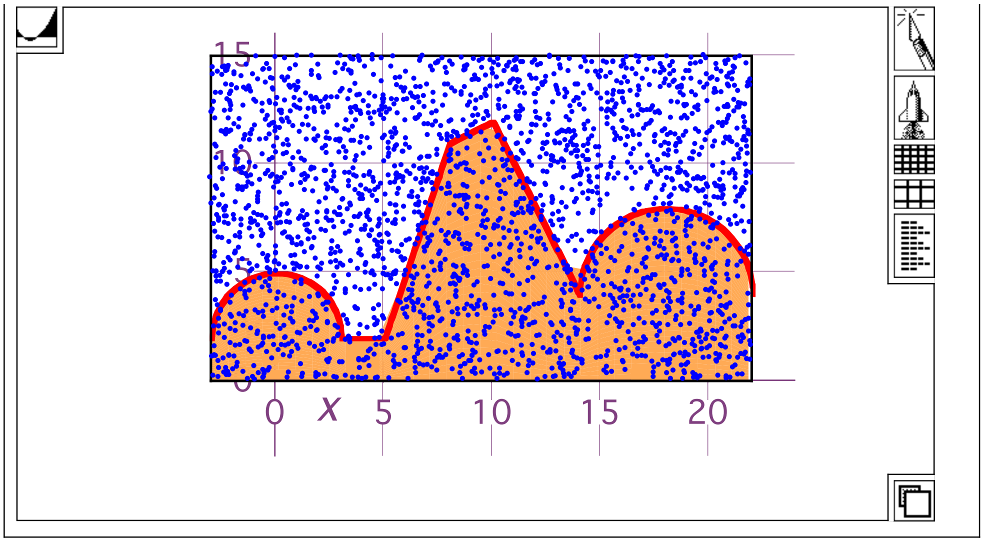
$ylow = 0$ $yhigh = 15$

$$f(x) = \begin{cases} \sqrt{9 - x^2} + 2 & (-3 \le x)(x < 3) \\ 2 & (3 \le x)(x < 5) \\ 3(x-5)+2 & (5 \le x)(x < 8) \\ 0.5(x-8)+11 & (8 \le x)(x < 10) \\ (-2)(x-10)+12 & (10 \le x)(x < 14) \\ \sqrt{16-(x-18)^2} + 4 & (14 \le x)(x \le 22) \end{cases}$$

LiveMath Note: Using the functional approach to generating random numbers as demonstrated in the previous example

$xRandoms(k) = Random(xlow, xhigh)$

$yRandoms(k) = Random(ylow, yhigh)$



The Monte carlo idea:

Because the points are approximately uniformly distributed, you (

$$\frac{\text{Area enclosed by curve}}{\text{Area enclosed by the box}} \approx \frac{\text{Number of random points inside curv}}{\text{Total number of random points inside l}}$$

so that:

$$\text{Area enclosed by curve} \approx \left(\frac{\text{Number of random points inside curve}}{\text{Total number of random points inside box}}\right) * \text{Area enc}$$

Try it out

$xlow = -3$ $xhigh = 22$

$ylow = 0$ $yhigh = 15$

$$f(x) = \begin{cases} \sqrt{9 - x^2} + 2 & (-3 \le x)(x < 3) \\ 2 & (3 \le x)(x < 5) \\ 3(x-5)+2 & (5 \le x)(x < 8) \\ 0.5(x-8)+11 & (8 \le x)(x < 10) \\ (-2)(x-10)+12 & (10 \le x)(x < 14) \\ \sqrt{16-(x-18)^2} + 4 & (14 \le x)(x \le 22) \end{cases}$$

LiveMath Note: Using the functional approach to generating random numbers as demonstrated in the previous example

$\text{xRandoms}(k) = \text{Random}(\text{xlow}, \text{xhigh})$

$\text{yRandoms}(k) = \text{Random}(\text{ylow}, \text{yhigh})$

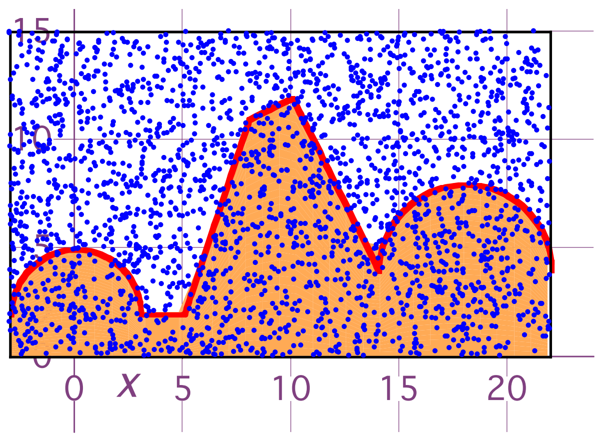
$$\text{fCounts}(n) = \sum_{k=1}^{2500} (\text{yRandoms}[k] \leq f[\text{xRandoms}\{k\}])$$

$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$

$$\text{AreaEst}(n) = \frac{\text{fCounts}(n)}{2500} \text{BoxArea}$$

$\text{AreaEst}(1)$

$\text{AreaEst}(1) = 162.9$ *Calculate*



Now, average to try to get a better estimate:

$\text{xlow} = -3$ $\text{xhigh} = 22$

$\text{ylow} = 0$ $\text{yhigh} = 15$

$$f(x) = \begin{cases} \sqrt{9 - x^2} + 2 & (-3 \leq x)(x < 3) \\ 2 & (3 \leq x)(x < 5) \\ 3(x-5)+2 & (5 \leq x)(x < 8) \\ 0.5(x-8)+11 & (8 \leq x)(x < 10) \\ (-2)(x-10)+12 & (10 \leq x)(x < 14) \\ \sqrt{16-(x-18)^2} + 4 & (14 \leq x)(x \leq 22) \end{cases}$$

LiveMath Note: Using the functional approach to generating random numbers as demonstrated in the previous example

$\text{xRandoms}(k) = \text{Random}(\text{xlow}, \text{xhigh})$

$\text{yRandoms}(k) = \text{Random}(\text{ylow}, \text{yhigh})$

$$\text{fCounts}(n) = \sum_{k=1}^{2500} (\text{yRandoms}[k] \le f[\text{xRandoms}\{k\}])$$

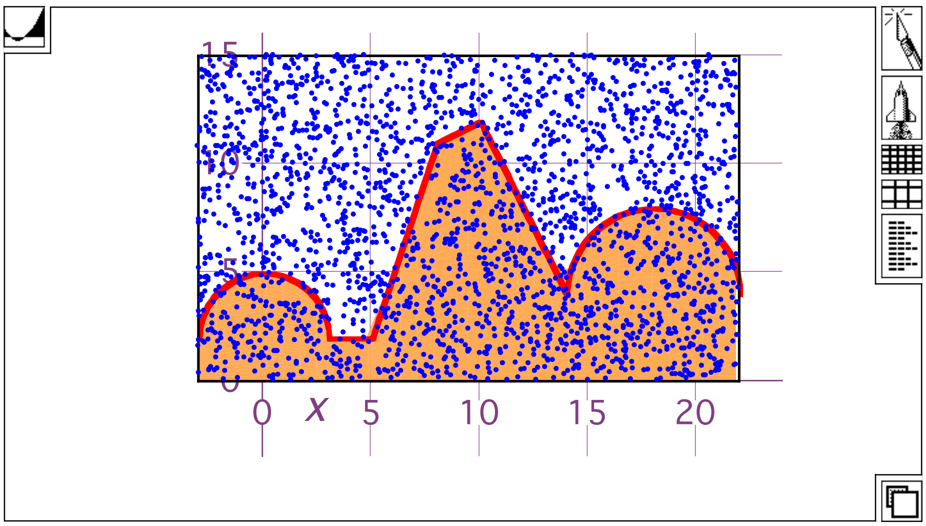
$$\text{BoxArea} = (\text{xhigh} - \text{xlow})(\text{yhigh} - \text{ylow})$$

$$\text{AreaEst}(n) = \frac{\text{fCounts}(n)}{2500}\text{BoxArea}$$

The average of the 10 Monte Carlo estimates is:

$$\text{Average} = \frac{\sum_{k=1}^{10} \text{AreaEst}(k)}{10}$$

$\text{Average} = 160.71$ *Calculate*

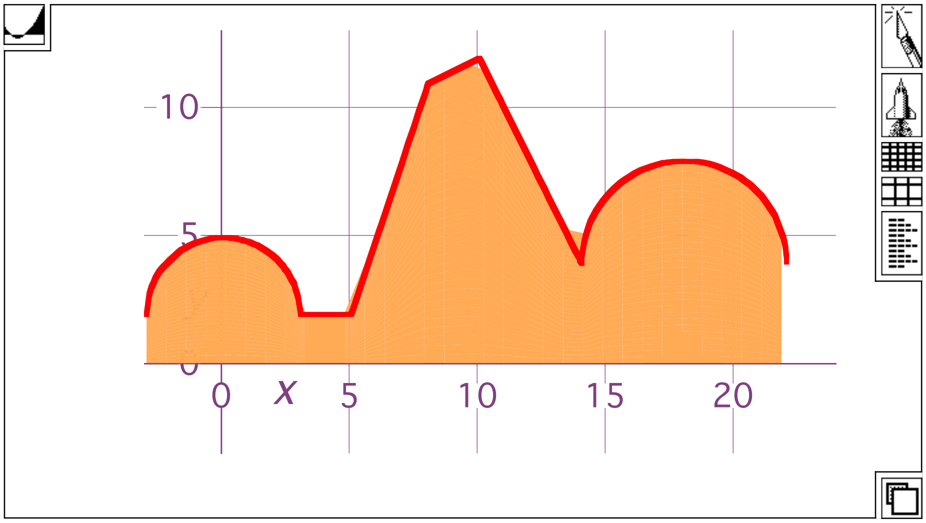


Well, now the no fun part: we should really check the areas of these geometric regions to make sure this magic really works.

$$\text{xlow} = -3$$

$$\text{xhigh} = 22$$

$$f(x) = \begin{cases} \sqrt{9 - x^2} + 2 & (-3 \le x)(x < 3) \\ 2 & (3 \le x)(x < 5) \\ 3(x-5)+2 & (5 \le x)(x < 8) \\ 0.5(x-8)+11 & (8 \le x)(x < 10) \\ (-2)(x-10)+12 & (10 \le x)(x < 14) \\ \sqrt{16-(x-18)^2}+4 & (14 \le x)(x \le 22) \end{cases}$$



First 1/2 circle over [-3, 3]

$A_1 = \frac{1}{2}\pi \cdot 3^2$

Rectangle underneath 1st half circle over [-3, 3]

$A_2 = 6 \cdot 2$

Next rectange over [3,5]

$A_3 = (5-3) \cdot 2$

Next triangle over [5,8]

$A_4 = \frac{1}{2}(8-5)(11-2)$

Next rectange under triangle over [5,8]

$A_5 = 3 \cdot 2$

Next trapezoid over [8,10]

$A_6 = (10-8)\frac{11+12}{2}$

Next triangle over [10,14]

$A_7 = \frac{1}{2}(14-10)(12-4)$

Next rectange under triangle over [10,14]

$A_8 = (14-10) \cdot 4$

Last 1/2 circle over [14, 22]

$A_9 = \frac{1}{2}\pi \cdot 4^2$

Rectangle underneath 1st half circle over [14, 22]

$A_{10} = (22 - 14) \cdot 4$

Sum all of these up

$A = \sum_{k=1}^{10} A_k$

$A = A_{10} + A_9 + A_8 + A_7 + A_6 + A_5 + A_4 + A_3 + A_2 + A_1$ *Expand*

$A = 161.769908169872$ *Calculate*

Because of the Law of Large Numbers, the smart money bets that the average was not all that bad: 161.76 vs. 161.805