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.05 Normal and Exponential

Basics B2

Experience with the starred problems will be useful for understanding developme

Graphics Primitives

Accumulating Collection of Stat Functions [v5.2]

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

B.2) Approximately normally distributed data sets: The normal (Gaussian) distribution

B.2.a.i) "Normal Distributions"

The idea of "normally distributed" data sets is a big buzzword in mathematical, physical, biological, and social sciences.

What do folks mean when they say that a data set is *approximately normally distributed*?

Answer:

When they say that a data set is approximately normally distributed, they mean that the *cumulative distribution function* CumDist(x,X) can be described via some basic algebraic formulas that are **completely determined** by the Expected Value μ and the Standard Deviation σ.

Why is this useful? Because if someone walks up to you on the street with a data set X and says, "This data set is approximately normally distributed", the computing just two numbers of that set X -- μ and σ -- will completely determine the CumDist function, and thus the computations of probabilities on the set X.

B.2.a.ii) The Bell Curve Associated to a Data Set X

The
normal
law of error
stands out in the
experience of mankind
as one of the broadest
generalizations of natural
philosophy ~ It serves as the
guiding instrument in researches
in the physical and social sciences and
in medicine, agriculture and engineering ~
It is an indispensable tool for the analysis and the
interpretation of the basic data obtained by observation and experiment.

----This bell shaped design is by statistician W. J. Youden

Let's look at a nice data set X:

$X = (8.9, 8, 12, 8.1, 6.2, 12, 9.6, 9.5, 9.1, 6.7, 9.4, 8, 8.9, 11, 9.4, 13, 9$

And let's compute the Expected Value μ

and Standard Deviation $\sigma = \sqrt{\text{Variance}(X)}$

$\mu = \text{ExpectVal}(X)$

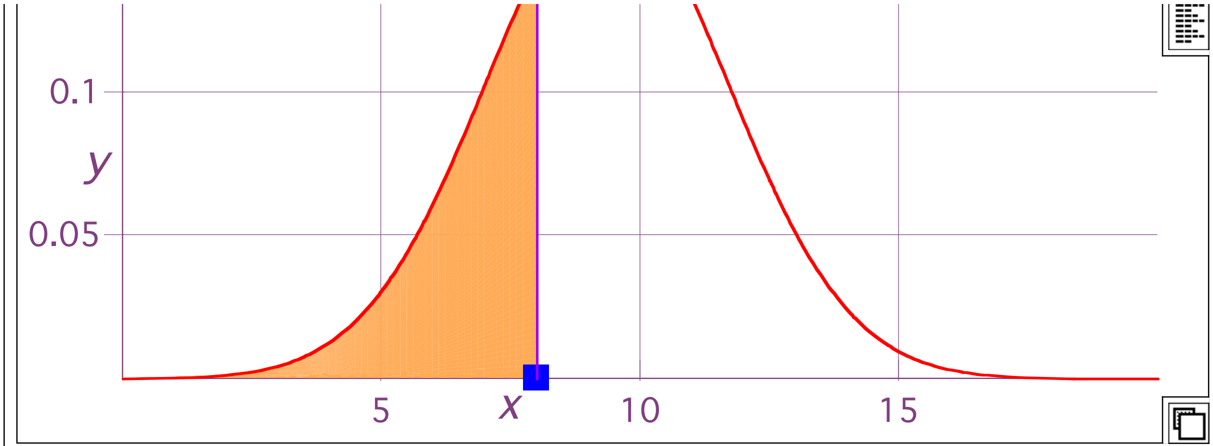
$\mu = 9.3495$ *Calculate*

$\sigma = \sqrt{\text{Var}(X)}$

$\sigma = 2.35355045622566$ *Calculate*

For any data set X, we look at the associated Bell Curve that is defined by the following formula using μ and σ, using our old friend Euler's number

e=2.71828....



What is the area of that yello region under the Bell Curve? We can't use basic geometry to get it, but we can use the Monte Carlo method!

Remember the Monte Carlo idea:

Because the points are approximately uniformly distributed, yo

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

so that:

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

Try it out

Prob(X≤ 8), so let a=8

$a = 8$

$\text{xlow} = 0$ $\text{xhigh} = a$

$\text{ylow} = 0$ $\text{yhigh} = 0.18$ Choosing yhigh to make sure box encloses the region we want

$$\text{BellCurve}(x) = 0.424889977333939\,\frac{e^{-0.0902657464194174}}{\sqrt{2\pi}}$$

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

STAT.01.T1

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

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

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

$BoxArea = (xhigh - xlow)(yhigh - ylow)$

$BoxArea = 8 \cdot 0.18$ *Calculate Calculate*

$BoxArea = 1.44$ *Calculate*

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

Do a few computations

$AreaEst(1)$

$AreaEst(1) = \frac{1}{2500} fCounts(1) BoxArea$ *Substitute*

$AreaEst(1) = \frac{1}{2500} \cdot 526 \cdot 1.44$ *Calculate Calculate*

$AreaEst(1) = 0.302976$ *Calculate*

$AreaEst(2)$

$AreaEst(2) = 0.298368$ *Calculate*

$AreaEst(3)$

$AreaEst(3) = 0.295488$ *Calculate*

Take 100 averages to get the best estimate:

$\frac{1}{100} \sum_{j=1}^{100} AreaEst(j)$

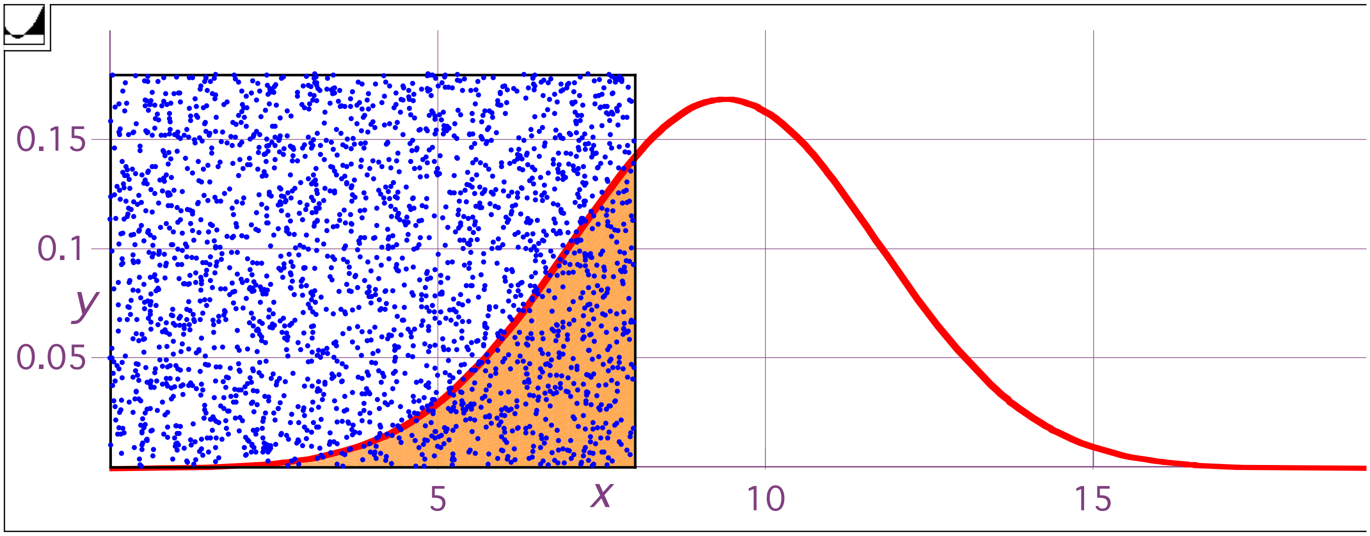
$\frac{1}{100} \sum_{j=1}^{100} AreaEst(j) = 0.28227456$ *Calculate*

Now, remember that Prob(X≤8) = CumDist(8, X)

$CumDist(8, X) = 0.275$

Pretty close. Notice that the computations above did not include the actual data set X - we only used μ, σ, the BellCurve(x) formula, and the Monte

Carlo method.



Computation #2: Prob(X≤ 11.5)

Since we know the data set X here, and we have LiveMath, we can compute this probability using the CumDist(x,X) funciton:

$X = (8.9, 8, 12, 8.1, 6.2, 12, 9.6, 9.5, 9.1, 6.7, 9.4, 8, 8.9, 11, 9.4,$

Prob(X≤ 11.5) = CumDist(11.5, X)

$\text{CumDist}(11.5, X)$

$\text{CumDist}(11.5, X) = 0.800000000000001$ *Calculate*

So the probability of a pull from X being ≤ 11.5 is 80%. Easy peasy.

But now let's look at something interesting with the associated Bell curve graph:

$\mu = 9.3495$

$\sigma = 2.35355045622566$

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.424889977333939\ \frac{e^{-0.090265746419417}}{\sqrt{2\pi}}$$

Prob(X≤ 11.5), so let a=11.5

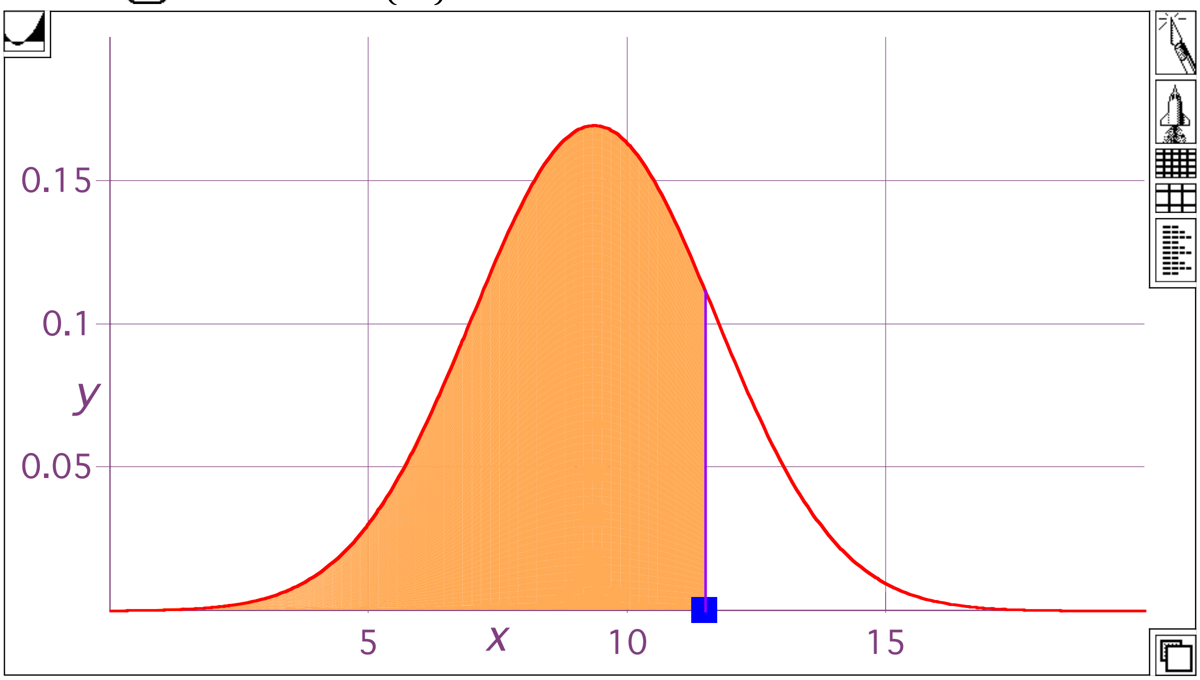
$a = 11.5$

BellCurve(a)

$\text{BellCurve}(a) = \text{BellCurve}(8)$

$$\text{BellCurve}(a) = 0.424889977333939\ \frac{1}{e^{0.16438748665815 8}\sqrt{2}}$$

$\text{BellCurve}(a) = 0.14381161856011$



What is the area of that yello region under the Bell Curve? We can't use basic geometry to get it, but we can use the Monte Carlo method!

Remember the Monte Carlo idea:

Because the points are approximately uniformly distributed, yo

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

so that:

Area enclosed by curve $\approx$

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

Try it out

Prob(X≤ 11.5), so let a=11.5

$a = 11.5$

xlow = 0 xhigh = a

ylow = 0 yhigh = 0.18 Choosing yhigh to make sure box encloses the region we want

$$\text{BellCurve}(x) = 0.424889977333939\,\frac{e^{-0.0902657464194174}}{\sqrt{2\pi}}$$

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

STAT.01.T1

xRandoms(k) = Random(xlow , xhigh)

yRandoms(k) = Random(ylow , yhigh)

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

BoxArea = (xhigh − xlow)(yhigh − ylow)

BoxArea = 2.07 *Calculate*

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

Do a few computations

AreaEst(1)

AreaEst(1) = 0.818892 *Calculate*

AreaEst(2)

AreaEst(2) = 0.818892 *Calculate*

AreaEst(3)

AreaEst(3) = 0.827172 *Calculate*

Take 100 averages to get the best estimate:

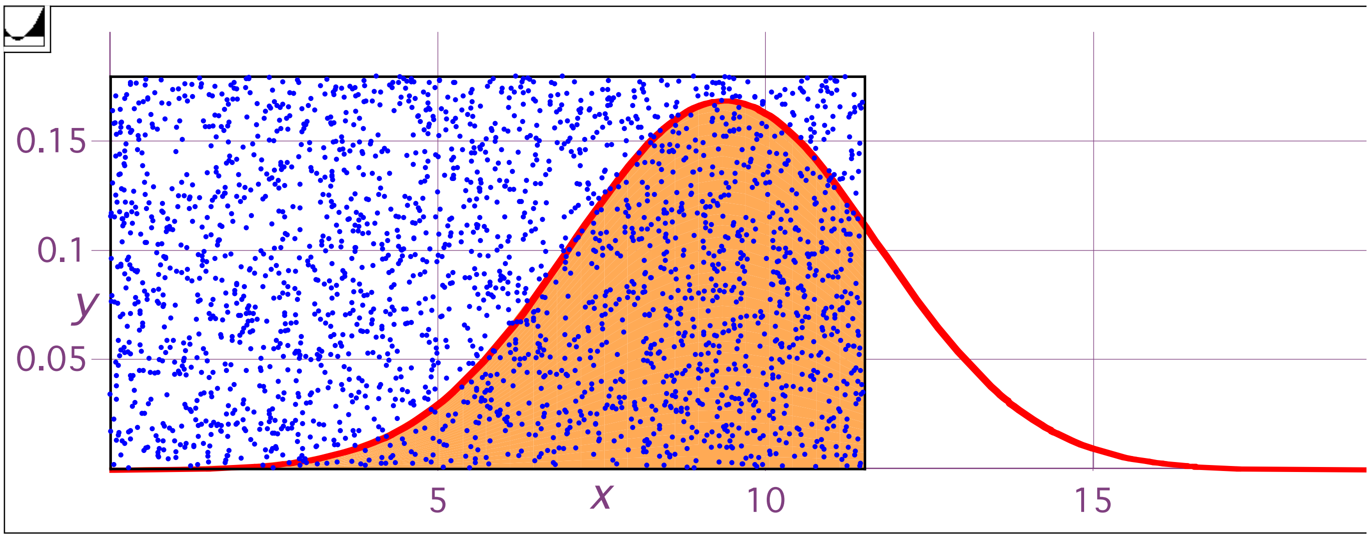
$\frac{1}{100}\sum_{j=1}^{100}$ AreaEst(j)

$\frac{1}{100}\sum_{j=1}^{100}$ AreaEst$(j) = 0.8224938$ *Calculate*

Now, remember that Prob(X≤11.5) = CumDist(11.5, X)

CumDist$(11.5, X) = 0.8$

Pretty close. Notice that the computations above did not include the actual data set X - we only used μ, σ, the BellCurve(x) formula, and the Monte Carlo method.



0 ... 20 = left...right Stretch to Fit

0 ... 0.2 = bottom...top cropping Moderately

Graph Building Blocks

Surface at (x, y) where x = xlow ... xhigh and $y = 0$ ... BellCurve July Lighting surface has no mesh and is shaded using Solid Camel is the solid color.

Curve at $(t$, BellCurve$[t])$ where $t = 0 \ldots 20$ with a extra heavy line, colored Red.

Box

Scatter plot of (xRandoms$[k]$, yRandoms$[k]$) where $k = 1 \ldots 250$ using point spots colored Blue.

using 2 point spots colored Blue.

Computation #3: Prob(9.2 < X ≤ 12.3)

Since we know the data set X here, and we have LiveMath, we can compute this probability using the CumDist(x,X) funciton:

$X = (8.9, 8, 12, 8.1, 6.2, 12, 9.6, 9.5, 9.1, 6.7, 9.4, 8, 8.9, 11, 9.4,$

Prob(9.2 < X ≤ 12.3)

= CumDist(12.3, X) - CumDist(9.2, X)

$\text{CumDist}(12.3, X) - \text{CumDist}(9.2, X)$

$\text{CumDist}(12.3, X) - \text{CumDist}(9.2, X) = 0.905000000000001$

$\text{CumDist}(12.3, X) - \text{CumDist}(9.2, X) = 0.44$ *Calculate*

So the probability of a pull from X being > 9.2 and ≤ 11.5 is 44%. Easy peasy.

But now let's look at something interesting with the associated Bell curve graph:

$\mu = 9.3495$

$\sigma = 2.35355045622566$

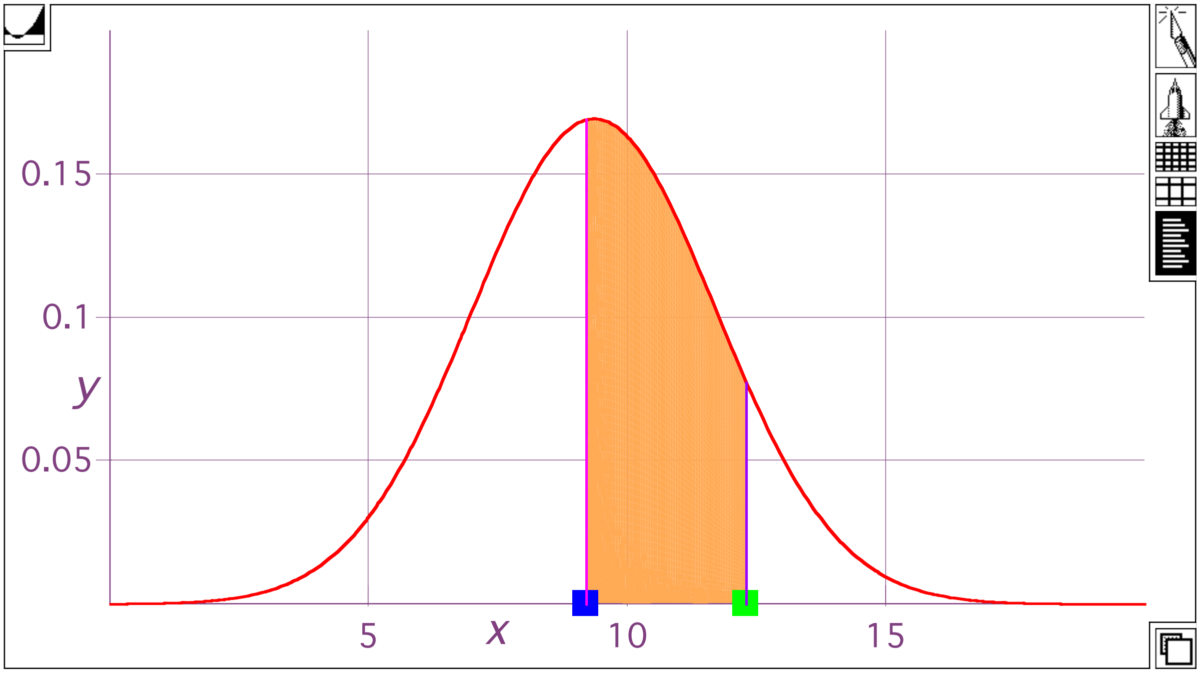
$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.424889977333939\,\frac{e^{-0.090265746419417}}{\sqrt{2\pi}}$$

Prob(9.2 < X≤ 12.3), so let a=9.2, b= 12.3

$a = 9.2$

$b = 12.3$



$0 \ldots 20$ = left...right   Stretch to Fit

$0 \ldots 0.2$ = bottom...top   cropping Moderately

Graph Building Blocks

Surface at (x, y) where $x = a \ldots b$ and $y = 0 \ldots \text{BellCurve}(x)$; April Lighting surface has no mesh and is shaded using Solid coloring; Camel is the solid color.

Curve at $(x, \text{BellCurve}[x])$ where x = left ... right with a heavy line, colored Red.

Scatter plot of $(a, 0)$ where ? using 10 point solid squares colored Blue.

Scatter plot of $(b, 0)$ where ? using 10 point solid squares colored Green.

Curve at $(a, 0)t + (1 - t)(a, \text{BellCurve}[a])$ where $t = 0 \ldots 1$ with a heavy line, colored Magenta.

Curve at $(b, 0)t + (1 - t)(b, \text{BellCurve}[b])$ where $t = 0 \ldots 1$ with a heavy line, colored Purple.

What is the area of that yello region under the Bell Curve? We can't use basic geometry to get it, but we can use the Monte Carlo method!

Remember the Monte Carlo idea:

Because the points are approximately uniformly distributed, yo

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

so that:

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

Try it out

Prob(9.2 < X≤ 12.3), so let a=9.2, b= 12.3

$a = 9.2$

$b = 12.3$

$\text{xlow} = a$ $\quad$ $\text{xhigh} = b$

$\text{ylow} = 0$ $\quad$ $\text{yhigh} = 0.18$ $\quad$ Choosing yhigh to make sure box encloses the region we want

$$\text{BellCurve}(x) = 0.424889977333939\,\frac{e^{-0.0902657464194174}}{\sqrt{2\pi}}$$

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

STAT.01.T1

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

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

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

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

$\text{BoxArea} = 0.558$ *Calculate*

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

Do a few computations

AreaEst(1)

AreaEst(1) = 0.4178304 *Calculate*

AreaEst(2)

AreaEst(2) = 0.4167144 *Calculate*

AreaEst(3)

AreaEst(3) = 0.4303296 *Calculate*

Take 100 averages to get the best estimate:

$\frac{1}{100}\sum_{j=1}^{100} \text{AreaEst}(j)$

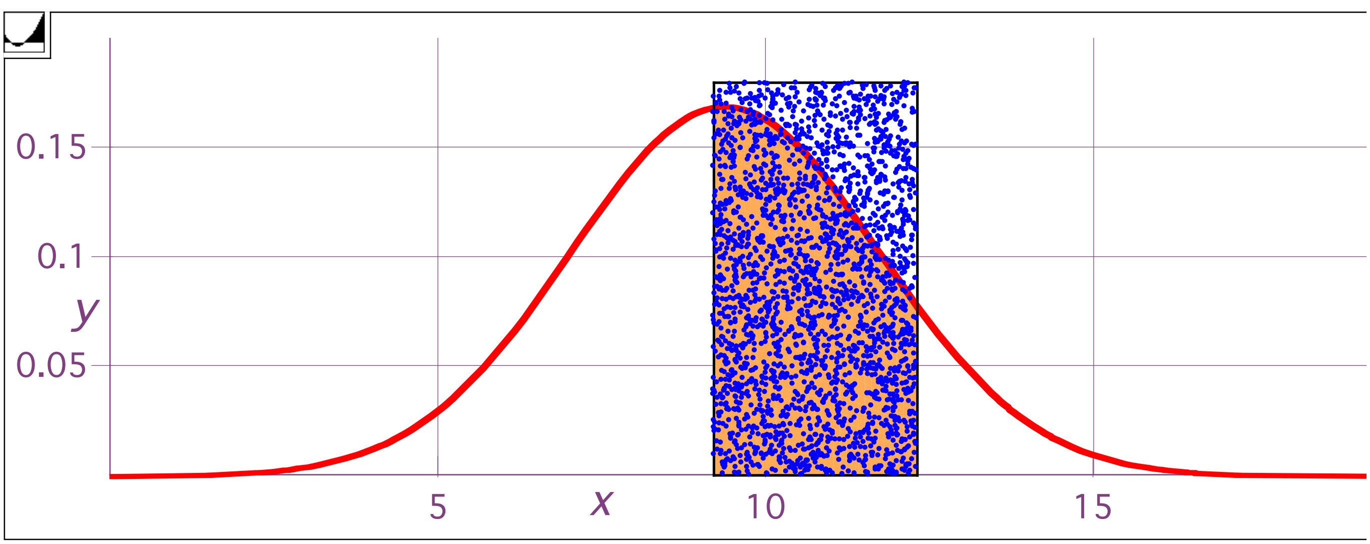
$\frac{1}{100}\sum_{j=1}^{100} \text{AreaEst}(j) = 0.421544448$ *Calculate*

Now, remember that
Prob(9.2 < X ≤ 12.3)

= CumDist(12.3, X) - CumDist(9.2, X)

$\text{CumDist}(12.3, X) - \text{CumDist}(9.2, X) = 0.44$

Pretty close. Notice that the computations above did not include the actual data set X - we only used μ, σ, the BellCurve(x) formula, and the Monte Carlo method.



0 ... 20 = left...right Stretch to Fit

0 ... 0.2 = bottom...top cropping Moderately

Graph Building Blocks

Surface at (x, y) where x = xlow ... xhigh and $y = 0$ ... BellCurve July Lighting surface has no mesh and is shaded using Solid Camel is the solid color.

Curve at $(t, \text{BellCurve}[t])$ where $t = 0 \ldots 20$ with a extra heavy line, colored Red.

Box

Scatter plot of $(\text{xRandoms}[k], \text{yRandoms}[k])$ where $k = 1 \ldots 250$ using 2 point spots colored Blue.

Computation #4: Area of BellCurve vs. CumDist(x,X)

We will now set up a Case Theory that graphs the CumDist(x,X) function, along with a number of Area computations from the Bell Curve.

Area under Bell Curve vs. CumDist(x,X)=Prob(x≤X)

$X = (8.9, 8, 12, 8.1, 6.2, 12, 9.6, 9.5, 9.1, 6.7, 9.4, 8, 8.9, 11, 9.4,$

$\mu = \text{ExpectVal}(X)$

$\mu = 9.3495$ Calculate

$\sigma = \sqrt{\text{Var}(X)}$

$\sigma = \sqrt{5.53919975}$ Calculate

$\sigma = 2.35355045622566$ Calculate

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.424889977333939\,\frac{e^{-0.0902657464194 17}}{\sqrt{2\pi}}$$

N = Number of Area values to compute.

Depending upon your patience level, you may increase the number chop-up points, as well as increase the number of random samples and number of averages taken.

$N = 8$

K = Which Area Region to show

$K = 3$

The Monte Carlo Computations

$\text{xlow} = 0$

$\text{ylow} = 0$ $\quad \text{yhigh} = \text{BellCurve}(\mu)$

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

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

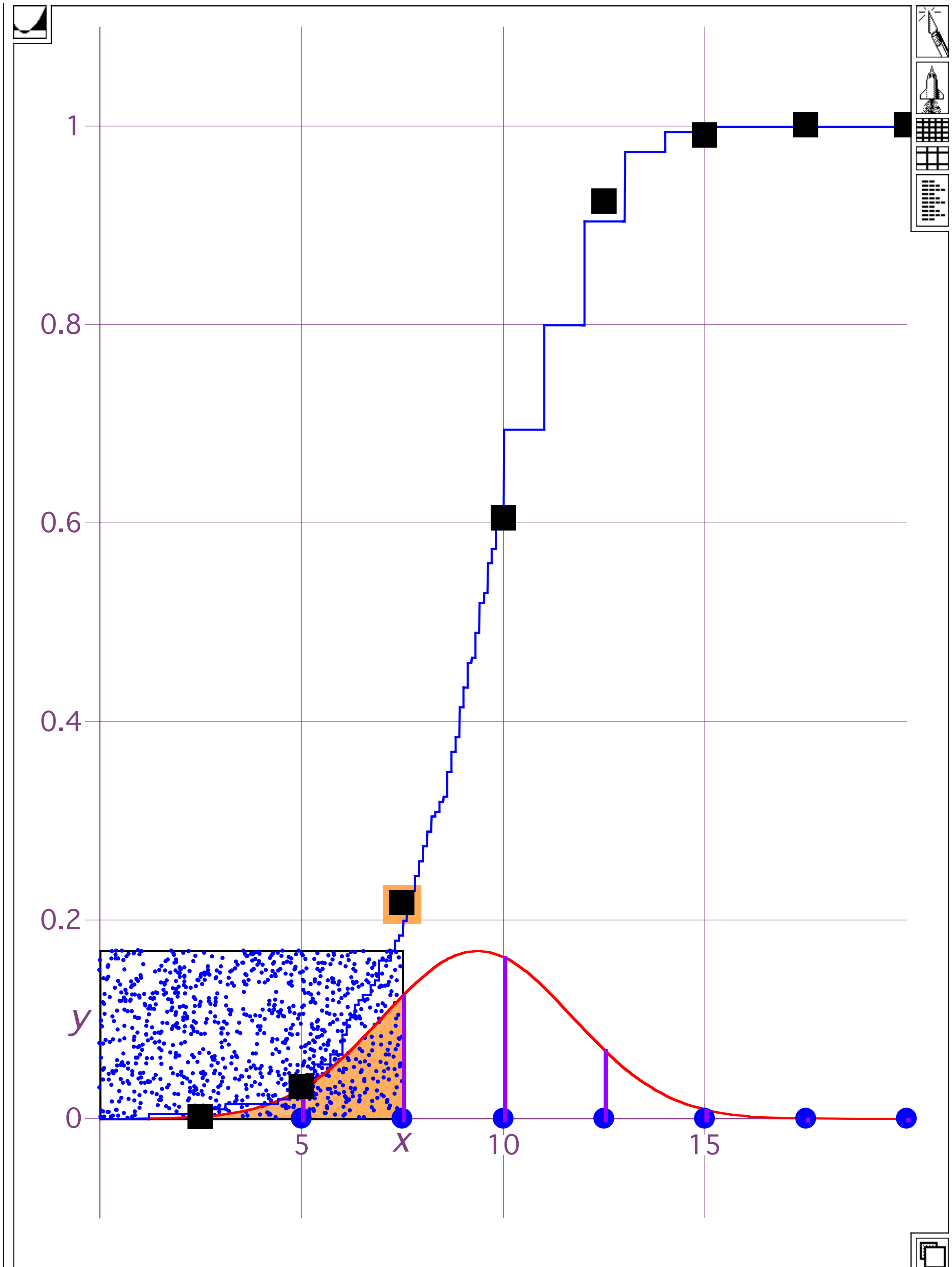
$$\text{fCounts}(n, x) = \sum_{k=1}^{1000} (\text{yRandoms}[k] \le \text{BellCurve}[\text{xRandoms}\{k$$

$$\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$$

$$\text{AreaEst}(n, x) = \frac{\text{fCounts}(n, x)}{1000} \text{fBoxArea}(x)$$

$$\text{AreaAvg}(n, x) = \frac{1}{20} \sum_{j=1}^{20} \text{AreaEst}(n, x)$$

Black squares height = Area under Bell Curve at Blue Spots



The accumulating areas under the Bell Curve match up perfectly with the CumDist(x,X) function.

This means: You can compute probabilities of the data set X by examining the AREA under the associated Bell curve.

B.2.b.i) Does this Bell curve Area -> CumDist(x,X) work for all data sets X?

Does this Area under the Bell curve will compute the Prob(x≤X) = CumDist(x,X) trickery work for all data sets X?

Answer:

Nope.

The data sets that this **does** work for are called *approximately normally distributed*.

B.2.b.ii) Checking to see if a data set X is appoximately normally di

How do you tell whether a given data set X is approximately norma

Answer:

Just look at your data set in the Monte Carlo computation Case Theory above, and compare if the areas under the Bell curve match up with the CumDist(x,X) function.

Area under Bell Curve vs. CumDist(x,X)=Prob(x≤X)

$X = (12.8193, 6.24358, 0.392916, 5.74745, 18.8538, 4.97144,$

$\mu = \text{ExpectVal}(X)$

$\mu = 1.47577232$ *Calculate*

$\sigma = \sqrt{\text{Var}(X)}$

$\sigma = 8.23018899672781$ *Calculate*

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.121503892607762\,\frac{e^{-0.00738159795941927}}{\sqrt{2\pi}}$$

N = Number of Area values to compute.

Depending upon your patience level, you may increase the number chop-up points, as well as increase the number of random samples and number of averages taken.

$N = 8$

K = Which Area Region to show

$K = 3$

The Monte Carlo Computations

$\text{xlow} = -20$

$\text{ylow} = 0$ $\text{yhigh} = \text{BellCurve}(\mu)$

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

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

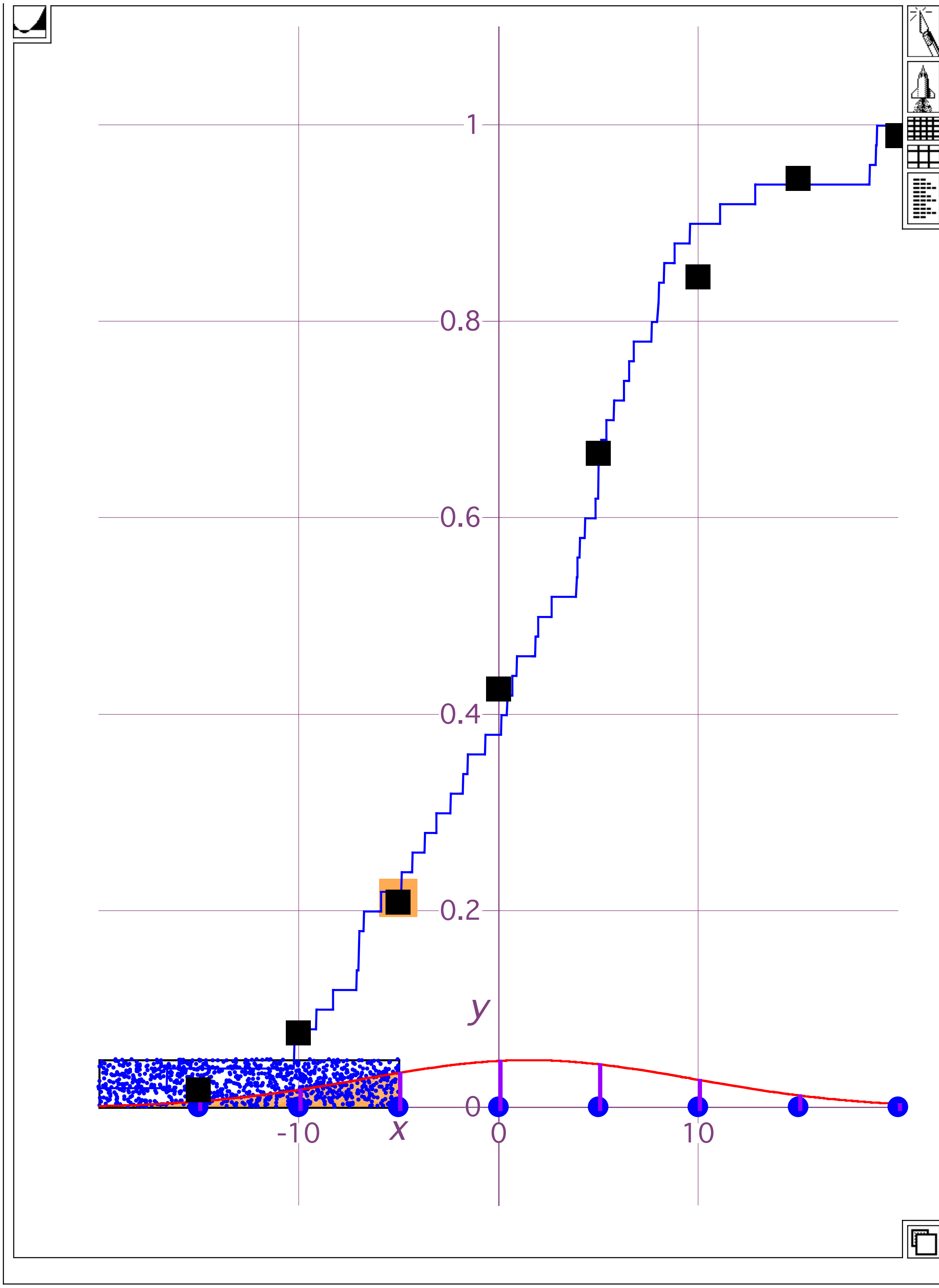
$$\text{fCounts}(n, x) = \sum_{k=1}^{1000} (\text{yRandoms}[k] \leq \text{BellCurve}[\text{xRandoms}\{k,$$

$$\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$$

$$\text{AreaEst}(n, x) = \frac{\text{fCounts}(n, x)}{1000} \text{fBoxArea}(x)$$

$$\text{AreaAvg}(n, x) = \frac{1}{20} \sum_{j=1}^{20} \text{AreaEst}(n, x)$$

Black squares height = Area under Bell Curve at Blue Spots



Not much doubt about it.

The Area squares match up near perfectly to the CumDist(x,X) cumulative distribution function of X.

The Call:

The given data set X is *approximately normally distributed*.

Try it again with this new data set:

Area under Bell Curve vs. CumDist(x,X)=Prob(x≤X)

$X = (1.5, 0.38, 4.2, 0.71, 6.1, 1.4, 2.2, 0.046, 0.31, 0.68, 0.91, 5.3,$

$\mu = \text{ExpectVal}(X)$

$\mu = 1.42471607142857$ *Calculate*

$\sigma = \sqrt{\text{Var}(X)}$

$\sigma = 1.49910960302593$ *Calculate*

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.667062633700374\,\frac{e^{-0.22248627863964(x-}}{\sqrt{2\pi}}$$

N = Number of Area values to compute.

Depending upon your patience level, you may increase the number chop-up points, as well as increase the number of random samples and number of averages taken.

$N = 8$

K = Which Area Region to show

$K = 3$

The Monte Carlo Computations

$\text{xlow} = \min(X)$

$\text{xlow} = 0.0012$ *Calculate*

$\text{xhigh} = \max(X)$

$\text{xhigh} = 9.9$ *Calculate*

$\text{ylow} = 0$ $\text{yhigh} = \text{BellCurve}(\mu)$

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

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

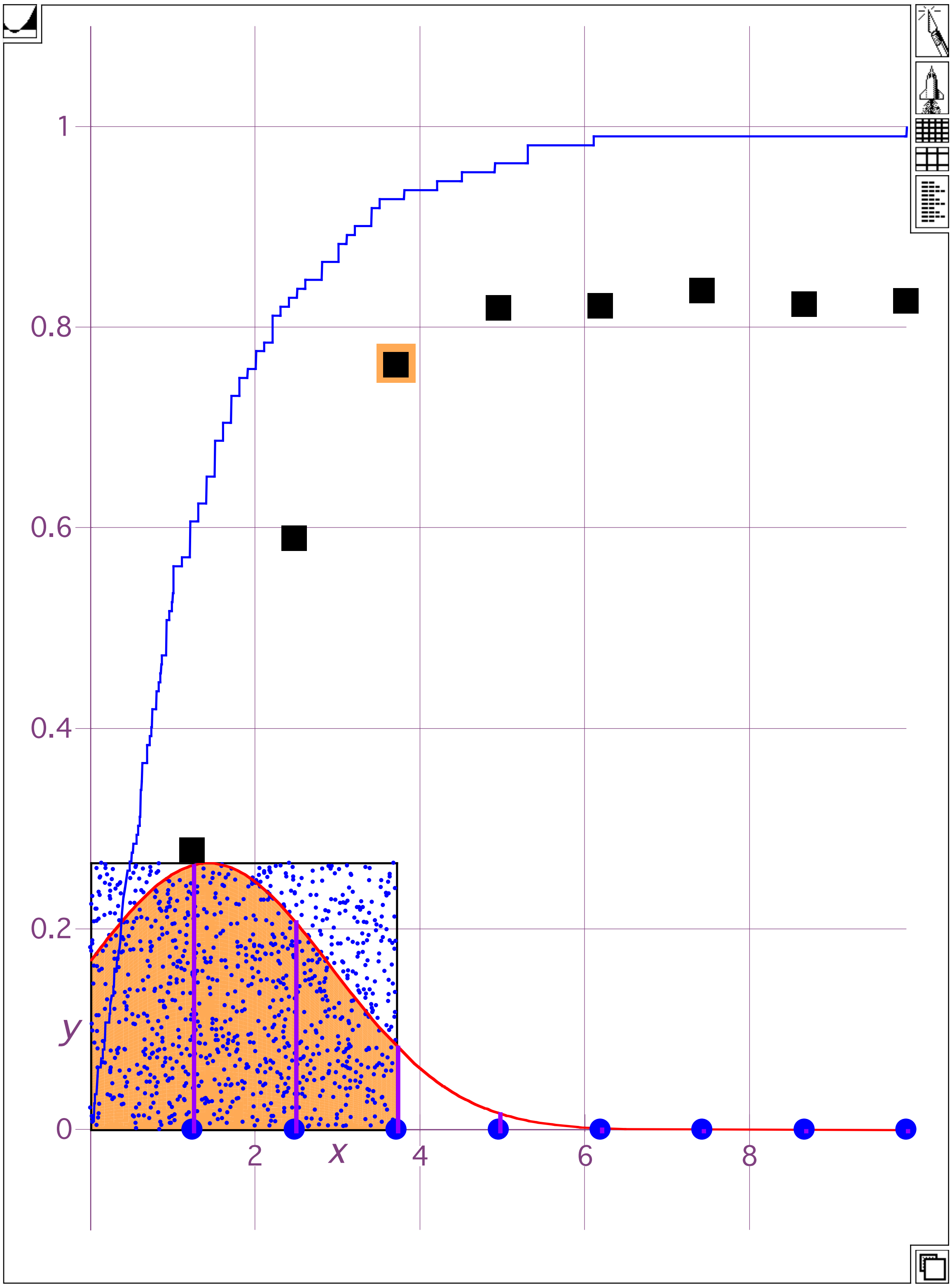
$$\text{fCounts}(n, x) = \sum_{k=1}^{1000} (\text{yRandoms}[k] \le \text{BellCurve}[\text{xRandoms}\{k,$$

$\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$

$$\text{AreaEst}(n, x) = \frac{\text{fCounts}(n, x)}{1000}\,\text{fBoxArea}(x)$$

$$\text{AreaAvg}(n, x) = \frac{1}{20}\sum_{j=1}^{20}\text{AreaEst}(n, x)$$

Black squares height = Area under Bell Curve at Blue Spots



Not much doubt about it.
The Area squares **do not match up** to the CumDist(x,X) cumulative distribution function of X.

The Call:

The given data set X is *NOT approximately normally distributed*.

B.2.c) The main advantage you get when you have an approximatel normally distributed data set

When you know that data set X is approximately normally distribute then you know that the CumDist(x,X) cumulative distribution functi can be computed by looking at the areas under the Bell Curve.

What is the main advantage you and others dealing with you get fr

Answer:

When you know that data set X is approximately normally distribu then there is little need to send anyone the whole data set. Inste can communicate most of its probability properties merely by sending the two numbers

μ = Expect(X) and $\sigma = \sqrt{\text{Var(X)}}$

and telling others that the data set is approximated normally dis Others can fire up the their Area calculators and do any probability estimates that they want.

For instance, when you say that your data set X is approximately distributed with

Expect(X) = 9.4 and Var(X) = 2.0,

and you are asked to compute the Prob(x≤10.7, X) = % of memb of the data set that are below or equal to 10.7.

Keep in mind: We will do this computation without knowing the data set!

Prob(x≤10.7, X)

$\mu = 9.4$

$\sigma = \sqrt{2}$

$\sigma = 1.4142135623731$ *Calculate*

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

$$\text{BellCurve}(x) = 0.707106781186547\,\frac{e^{-\frac{1}{4}(x-9.4)^2}}{\sqrt{2\pi}}$$ *Substitute*

$a = 10.7$

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

STAT.01.T1

The Monte Carlo Computations

$\text{xlow} = \mu - 5\sigma$ $\text{xhigh} = a$

$\text{xlow} = 2.32893218813452$ *Calculate*

$\text{ylow} = 0$ $\text{yhigh} = \text{BellCurve}(\mu)$

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

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

$$\text{fCounts}(n, x) = \sum_{k=1}^{1000} (\text{yRandoms}[k] \le \text{BellCurve}[\text{xRandoms}\{k,$$

$\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$

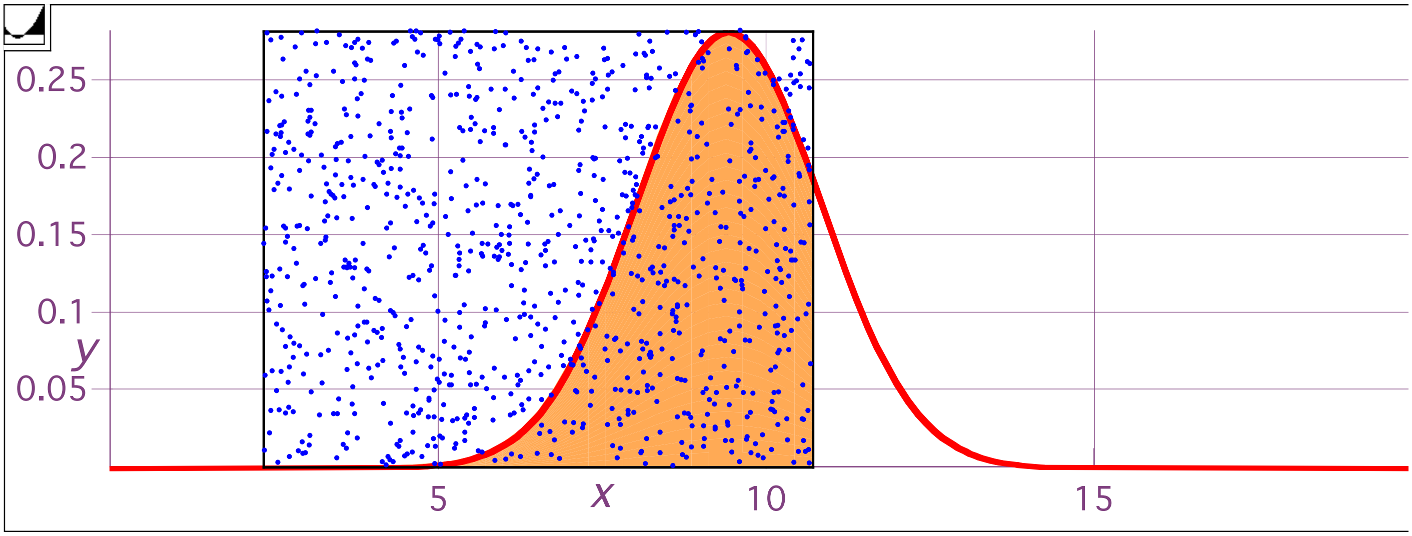
$$\text{AreaEst}(n, x) = \frac{\text{fCounts}(n, x)}{1000}\,\text{fBoxArea}(x)$$

$$\text{AreaAvg}(n, x) = \frac{1}{20}\sum_{j=1}^{20} \text{AreaEst}(n, x)$$

$\text{AreaAvg}(1, a)$

$\text{AreaAvg}(1, a) = \text{AreaAvg}(1, 10.7)$ *Substitute*

$\text{AreaAvg}(1, a) = 0.825793690570228$ *Calculate*



About 82% of the members of the data set X are below or equal

This is true for ANY DATA SET X that is:

a) Normally distributed

b) Expected Value = 9.4

c) Variance = 2.0

Scientists like normally distributed data sets because they can "r

from the probability computations

B.2.d) Random Numbers are easy to use with a computer to detern but how did they do this before computers?

Back in the old days, before computers and graphing and algebra software like LiveMath, the practical need for computing the Probabilities of a data set X were still very real. How did they do it?

Answer:

Using trapezoids to approximate the area under the Bell Curve. Take a look at this example:

Compute Prob($x \leq 5.6$) using only trapezoids

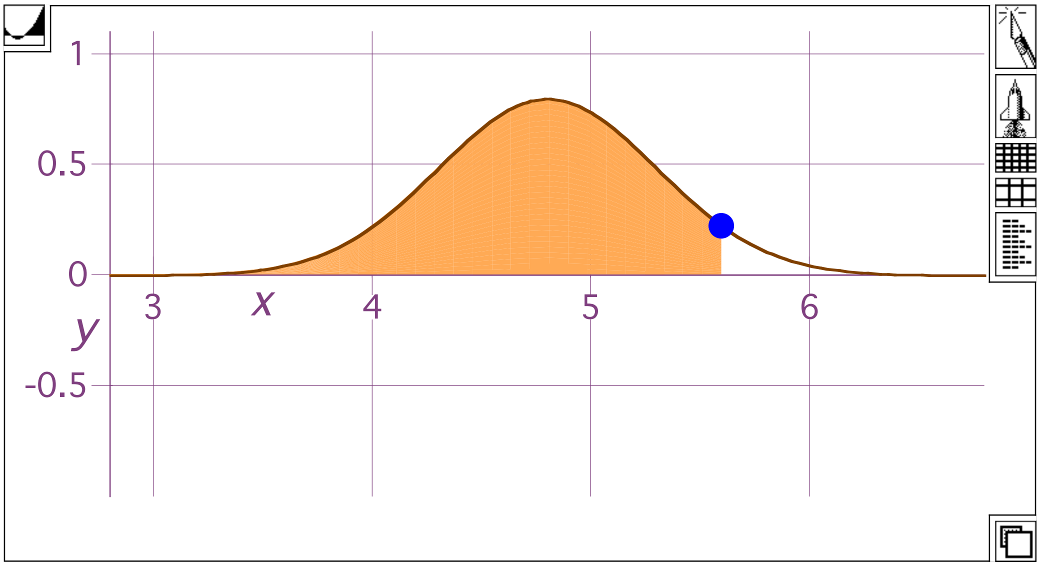
$\mu = 4.8$

$\sigma = 0.5$

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

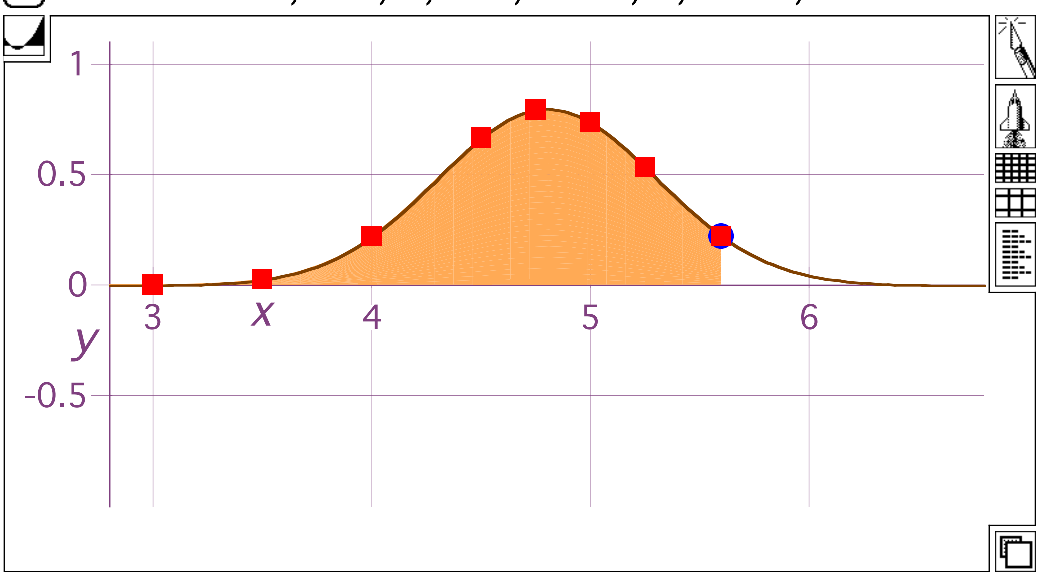
$a = 5.6$

Prob(x ≤ 5.6) = area under the Bell Curve.

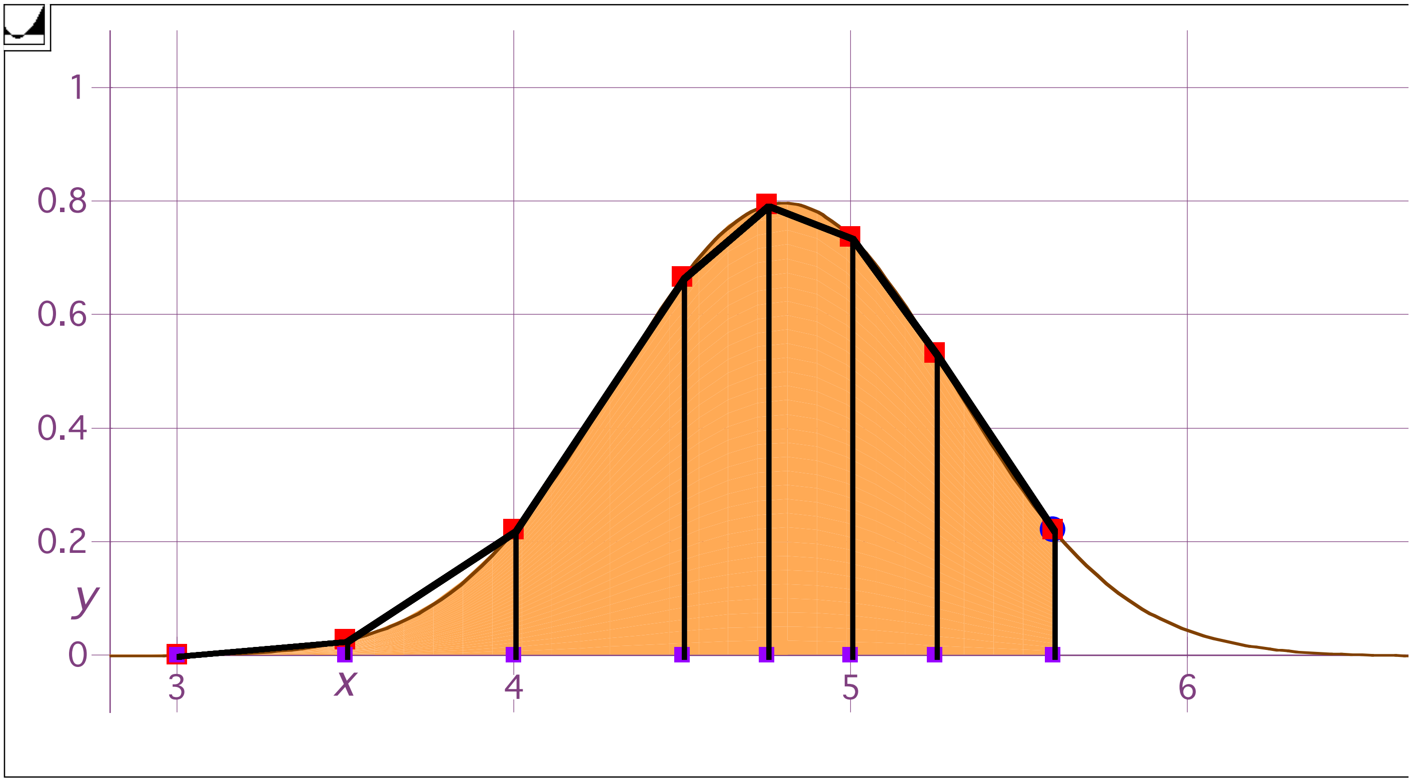


Choose some well-placed points to lay down some trapezoids:

$\text{Points} = (3, 3.5, 4, 4.5, 4.75, 5, 5.25, 5.6)$



Now draw in the trapezoids determined by these points.



Add up the areas of these trapezoids: some trapezoids will have a bit too much area, some will have a bit too little.

$$T_1 = \frac{\text{BellCurve}(3.5) + \text{BellCurve}(3)}{2} \cdot 0.5$$

$T_1 = 0.0141948711637994 \cdot 0.5$ *Calculate*

$T_1 = 0.0070974355818997$ *Calculate*

$$T_2 = \frac{\text{BellCurve}(4) + \text{BellCurve}(3.5)}{2} \cdot 0.5$$

$T_2 = 0.124503803913141 \cdot 0.5$ *Calculate*

$T_2 = 0.0622519019565706$ *Calculate*

$$T_3 = \frac{\text{BellCurve}(4.5) + \text{BellCurve}(4)}{2} \cdot 0.5$$

$T_3 = 0.444145437571255 \cdot 0.5$ *Calculate*

$T_3 = 0.222072718785628$ *Calculate*

$$T_4 = \frac{\text{BellCurve}(4.75) + \text{BellCurve}(4.5)}{2} \cdot 0.25$$

$T_4 = 0.730177150368811 \cdot 0.25$ *Calculate*

$T_4 = 0.182544287592203$ *Calculate*

$T_5 = \frac{\text{BellCurve}(5) + \text{BellCurve}(4.75)}{2} \cdot 0.25$

$T_5 = 0.765222687780335 \cdot 0.25$ *Calculate*

$T_5 = 0.191305671945084$ *Calculate*

$T_6 = \frac{\text{BellCurve}(5.25) + \text{BellCurve}(5)}{2} \cdot 0.25$

$T_6 = 0.634355390202078 \cdot 0.25$ *Calculate*

$T_6 = 0.158588847550519$ *Calculate*

$T_7 = \frac{\text{BellCurve}(5.6) + \text{BellCurve}(5.25)}{2}(5.6 - 5.25)$

$T_7 = 0.377006084578821\,(5.6 - 5.25)$ *Calculate*

$T_7 = 0.131952129602374$ *Calculate*

$T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7$

$T_1 + T_2 + T_3 + T_4 + T_5 + T_6 + T_7 = 0.955812993014278$ *Calc*

Area of these trapezoids is 95.5% = Prob (x ≤ 5.6, X) = CumDist(5.6, X)

All of these calculations above could be done by hand (a little calculator, or the calculator on your cell phone would make it less painful):

Let's check this answer against the Monte Carlo method for finding the area under the Bell Curve to compute Prob(x ≤ 5.6, X) :

Prob(x ≤ 5.6, X)

$\mu = 4.8$

$\sigma = 0.5$

$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$

$\text{BellCurve}(x) = 2\frac{e^{-2(x-4.8)^2}}{\sqrt{2\pi}}$ *Substitute*

$a = 5.6$

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

STAT.01.T1

The Monte Carlo Computations

$\text{xlow} = \mu - 5\sigma$ $\text{xhigh} = a$

$\text{xlow} = 2.32893218813452$

$\text{ylow} = 0$ $\text{yhigh} = \text{BellCurve}(\mu)$

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

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

$\text{fCounts}(n, x) = \sum_{k=1}^{1000} (\text{yRandoms}[k] \leq \text{BellCurve}[\text{xRandoms}\{k,$

$\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$

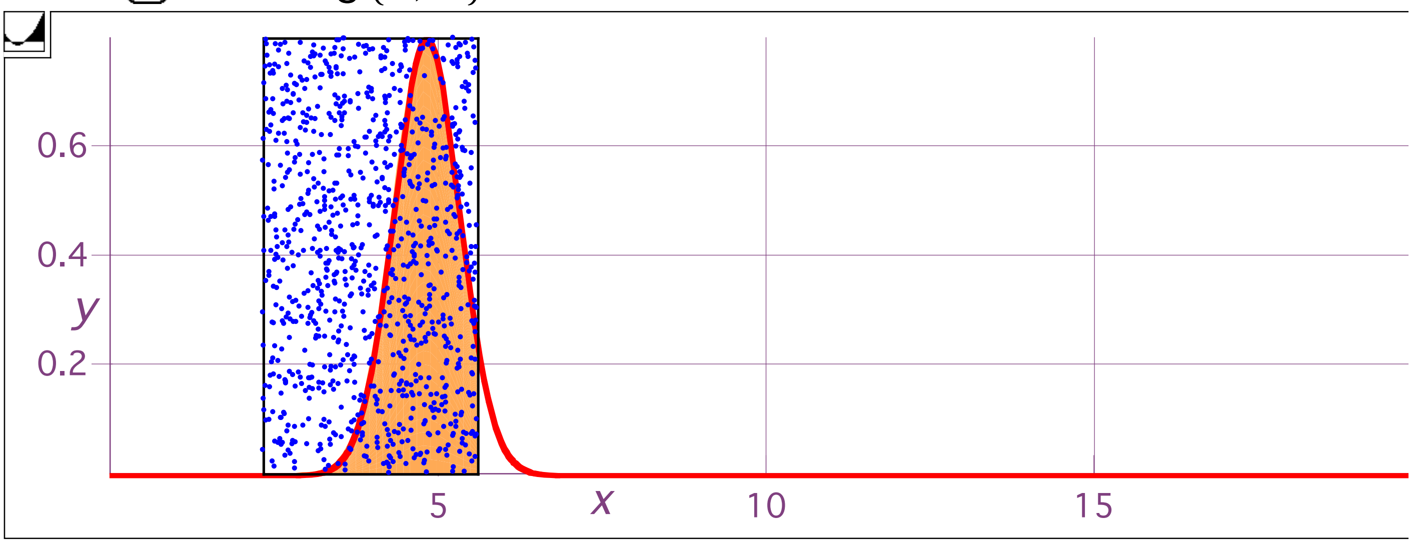
$\text{AreaEst}(n, x) = \dfrac{\text{fCounts}(n, x)}{1000} \text{fBoxArea}(x)$

$\text{AreaAvg}(n, x) = \frac{1}{20} \sum_{j=1}^{20} \text{AreaEst}(n, x)$

$\text{AreaAvg}(1, a)$

$\text{AreaAvg}(1, a) = \text{AreaAvg}(1, 5.6)$ *Substitute*

$\text{AreaAvg}(1, a) = 0.954453048268835$ *Calculate*



Here is a more compact Trapezoidal Probability Calculator

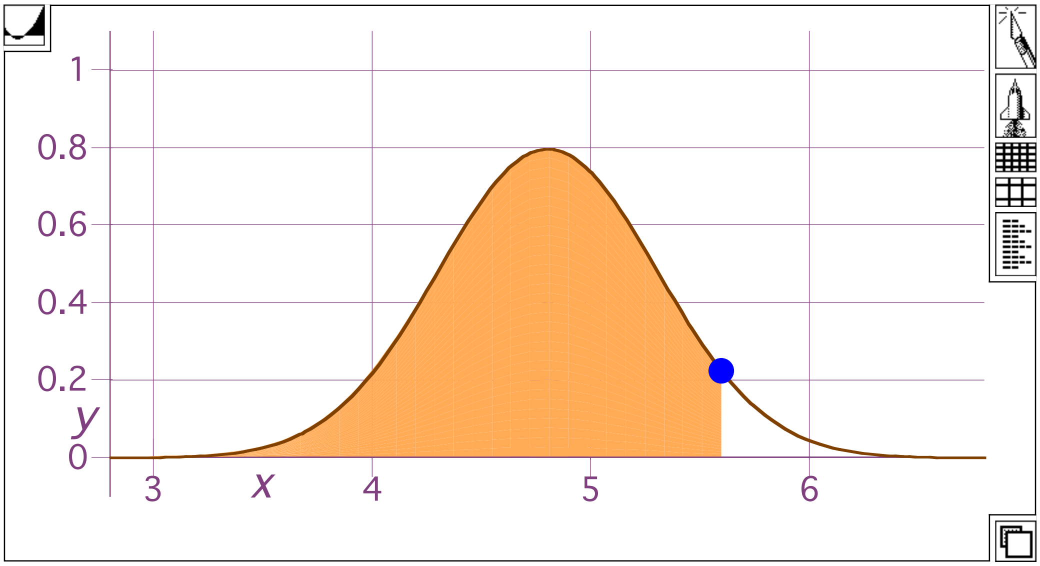
$\mu = 4.8$

$\sigma = 0.5$

$$\text{BellCurve}(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

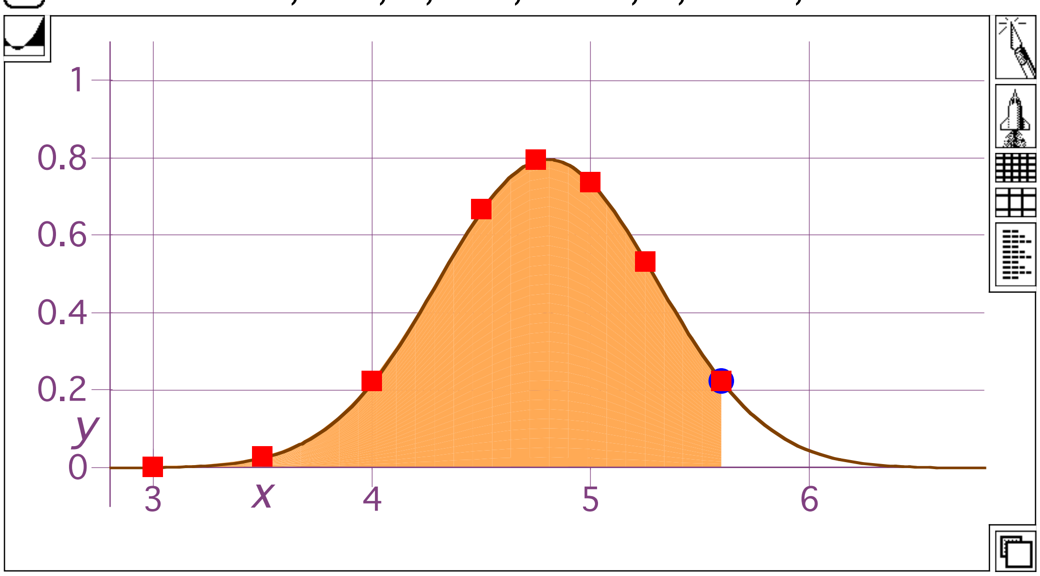
$a = 5.6$

Prob(x ≤ a.6) = area under the Bell Curve.

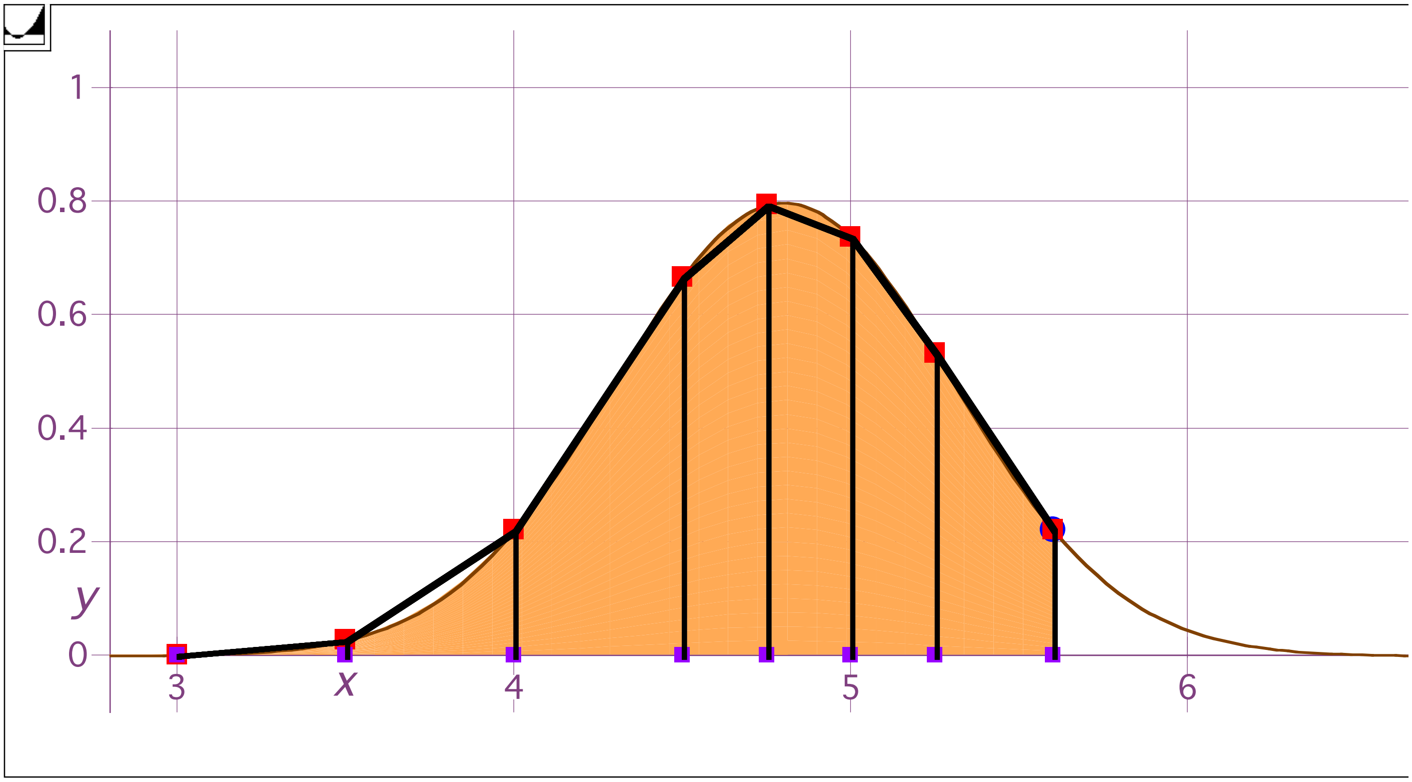


Choose some well-placed points to lay down some trapezoids:

$\text{Points} = (3, 3.5, 4, 4.5, 4.75, 5, 5.25, 5.6)$



Now draw in the trapezoids determined by these points.



Add up the areas of these trapezoids: some trapezoids will have a bit too much area, some will have a bit too little.
General formula:

$$TrapArea = \sum_{k=1}^{\#\,Points\,-\,1} \left(\frac{h1 + h2}{2}\right) * w$$

$$\text{TrapArea} = \sum_{k=1}^{\text{ColsOf}(\text{Points})-1} \frac{\text{BellCurve}\left(\text{Points}_{k+1}\right) + \text{BellCurve}\left(\text{Poi}\right.}{2}$$

$\text{TrapArea} = 0.955812993014278$ *Calculate*

Check using Monte Carlo method (that you could never do by hand) for accuracy:

The Monte Carlo Computations

$\text{xlow} = \mu - 5\sigma$ $\text{xhigh} = a$

$\text{xlow} = 2.32893218813452$

$\text{ylow} = 0$ $\text{yhigh} = \text{BellCurve}(\mu)$

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

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

$\text{fCounts}(n, x) = \sum_{k=1}^{1000} \left(\text{yRandoms}[k] \le \text{BellCurve}[\text{xRandoms}\{k,\right.$

- $\text{fBoxArea}(x) = (x - \text{xlow})(\text{yhigh} - \text{ylow})$
- $\text{AreaEst}(n, x) = \dfrac{\text{fCounts}(n, x)}{1000}\,\text{fBoxArea}(x)$
- $\text{AreaAvg}(n, x) = \frac{1}{20}\sum_{j=1}^{20} \text{AreaEst}(n, x)$
- $\text{AreaAvg}(1, a)$
  - $\text{AreaAvg}(1, a) = \text{AreaAvg}(1, 5.6)$ *Substitute*
    - $\text{AreaAvg}(1, a) = 0.948580695633875$ *Calculate*

Our computations on calculating probability, for a normally distributed data set, have gone from:

Brute Force: Using the full data set, calculate CumDist(a,X) = Prob(x ≤ a) via brute force

(hundreds or thousands of calculations (or more!).

Computer required.

Clever: Judiciously choose a few good points on the BellCurve graph, and compute the areas of their trapezoids:

Area of a few good trapezoids = Prob(x ≤ a).

No computer required!

That's how they computed probabilities before computers - with trapezoidal area under the Bell Curve.