Fathom™ Tutorial

Dynamic Statistical Software

Programming Simulations

using the power of

Fathom™

by Roland W. Meisel
Version 1.0 2004-09-28

Please Return This Guide to the Presenter at the End of the Workshop

if you would like an electronic copy of this guide, please send email to:

rollym@vaxxine.com

Thank you!

Foreword

This tutorial is designed for the teacher who is already familiar with the rudiments of Fathom™, and would like to use the power of the software to program simulations. Such simulations are useful in many courses, and particularly as part of the culminating project in the grade 12 Mathematics of Data Management course. The tutorial includes step-by-step keystroke instructions and liberal use of screen shots. However, the novice use may wish to first work through one of the other tutorials which offer an introduction to the basic features of Fathom™, such as the one keyed to the *McGraw-Hill Ryerson Mathematics of Data Management* text, © 2002, McGraw-Hill Ryerson Limited. The cited tutorial includes an example on how to use Fathom™ to simulate the effect of guessing on a multiple-choice test.

About Fathom™

Fathom™ is a powerful dynamic statistical software package published by

An excellent web site, with Fathom™ resources and links to other web sites, is at **www.keypress.com**.

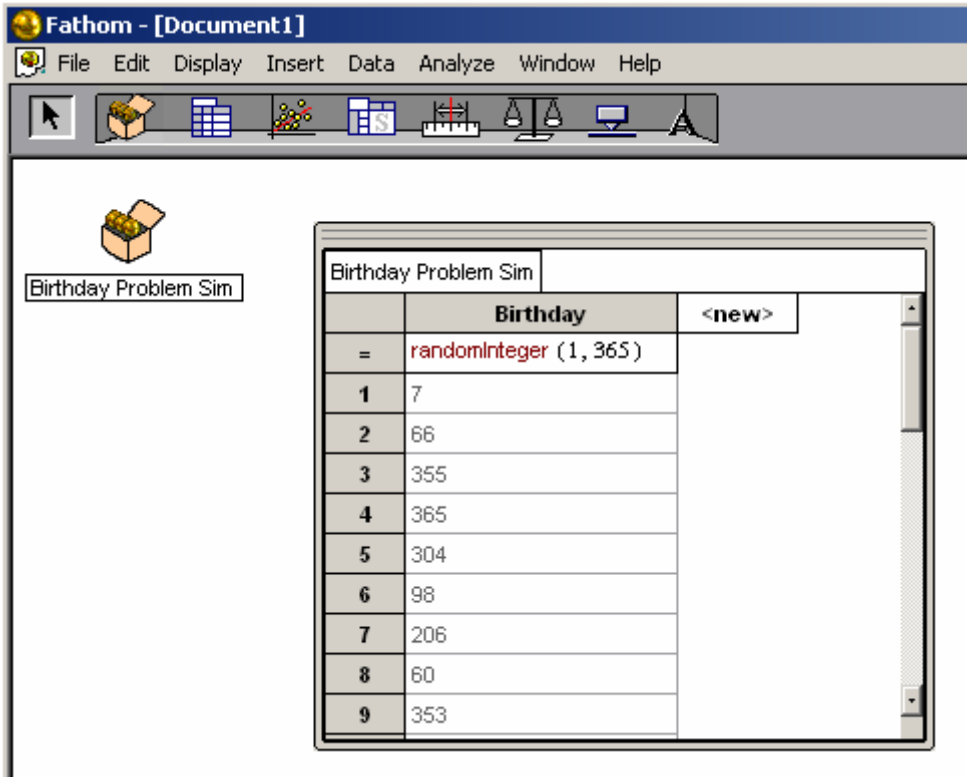
Introduction:

Fathom™ is a dynamic statistical software package that offers a variety of powerful data analysis tools in an easy-to-use format. A complete guide is available on the Fathom™ CD.

Example 1: The Birthday Problem

There are 30 people at a party. What is the probability that at least two of them have the same birthday?

Solution: At first, it would appear intuitive that the probability would be low, since there are 365 possible birthdays, and only 30 people at the party. This is a standard type of combinatoric problem, and can be solved using combinatorics and elementary probability theory. However, this does not illuminate the problem very much from the student's point of view. We'll use Fathom™ to simulate the problem, and find an empirical solution.



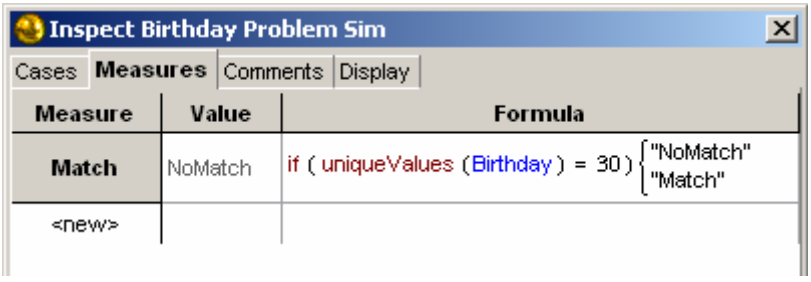
Run Fathom™, and create a collection with a case table of 30 cases and one attribute called **Birthday**. Edit the formula to:

randomInteger(1,365)

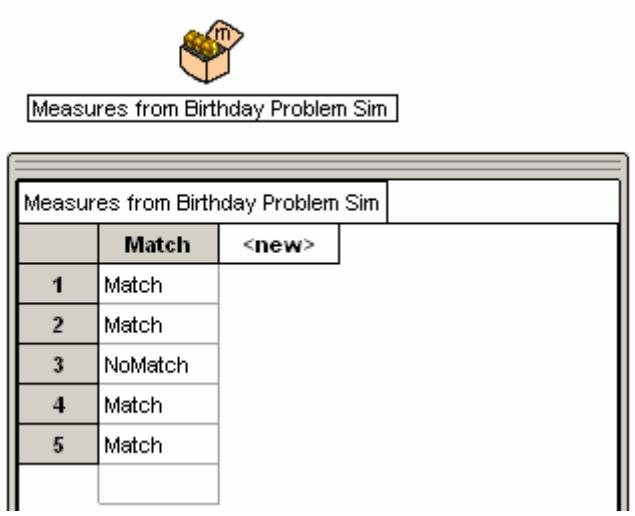
We could try to eyeball a match, but it is easier to use the **uniqueValues** function. Open the **Inspector** for the collection, and create a new measure called **Match**. Edit the formula to:

$$\text{if(uniqueValues(Birthday)} = 30 \begin{cases} \text{"NoMatch"} \\ \text{"Match"} \end{cases}$$

Note that the **uniqueValues** function looks through the list of 30 random birthdays, and counts how many different numbers are in the list. If there are 30 different birthdays, then no match was found.



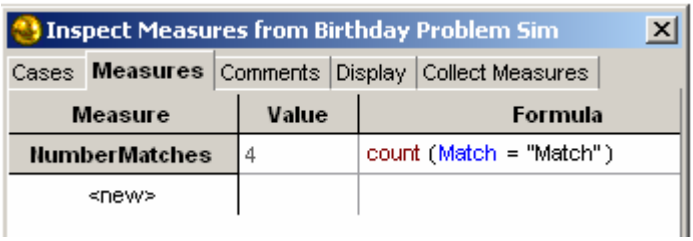
Select the Birthday Problem collection, and then select **Collect Measures** from the **Analyse** menu. Note that a new collection is created, called **Measures from Birthday Problem Sim**. Select this collection, and drag a new case table to the workspace. Note the number of matches.



Fathom™ has now run the simulation for five different parties, and noted whether a match occurred, or not, in each party.

Open the **Inspector** for the collection **Measures from Birthday Problem Sim**, and create a measure called **NumberMatches**. Edit the formula to:

count(Match = "Match").



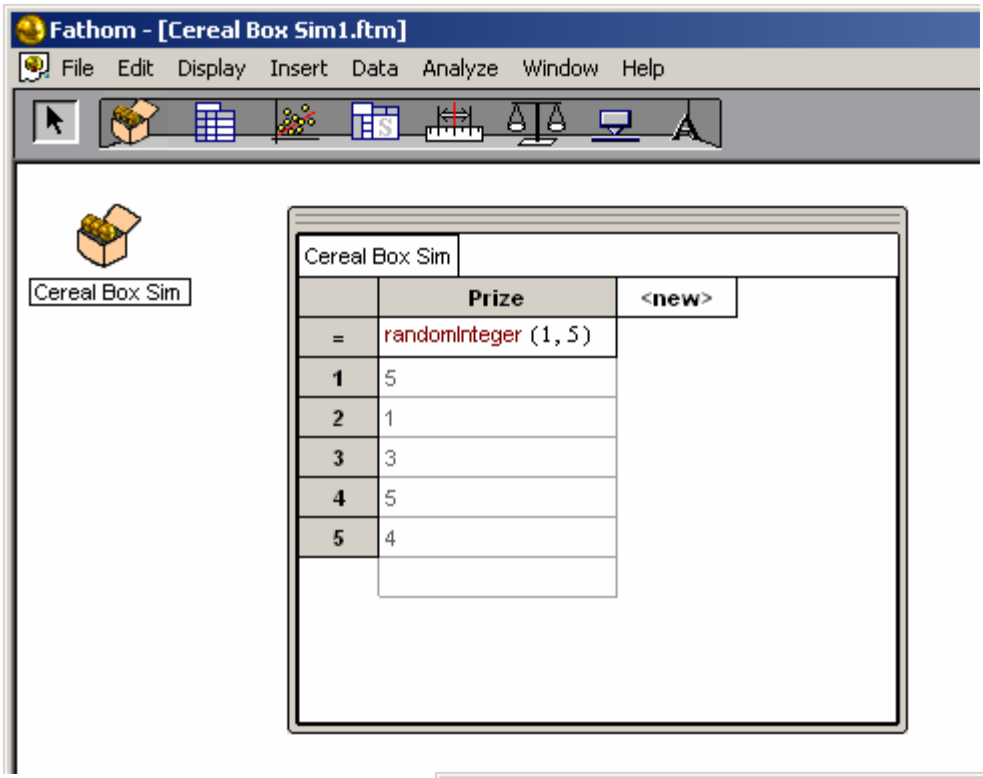
Select the **Collect Measures** tab. Turn off animation. Collect another five measures. Note how many matches and non-matches occurred. Increase the number of measures to 1000, and run the simulation. Repeat several times. Estimate the empirical probability of a match.

It appears that the probability of a match is very high when there are 30 people at the party. Select the collection **Birthday Problem Sim**, and reduce the number of cases to 25. Run 1000 measures several times. Has the probability of a match dropped below 0.5? Adjust the number of cases, and find the number of people at the party that results in a probability of 0.5 for a match.

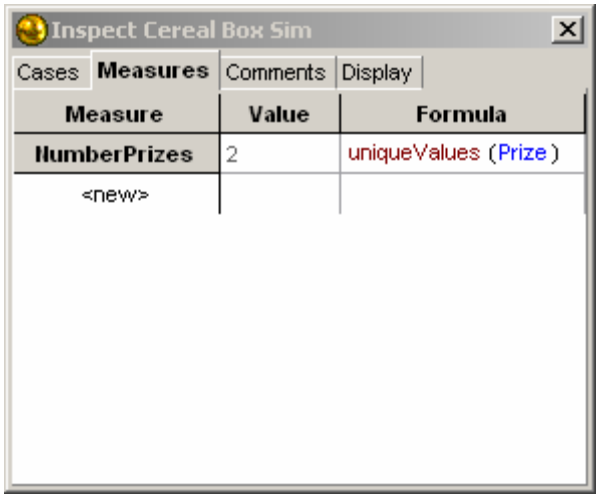
Example 2: The Cereal Box Problem

Five different prizes are available, one in each box of cereal. If you buy x boxes, $x > 4$, what is the probability that you will get all of the prizes?

Solution: Start with $x = 5$. Open Fathom™, and create a collection with one attribute called **Prize**, and five cases. Insert the formula randomInteger(1,5) for this attribute. Use **CTRL-y** to run several simulations.



Note whether any of your trials resulted in obtaining all five prizes.

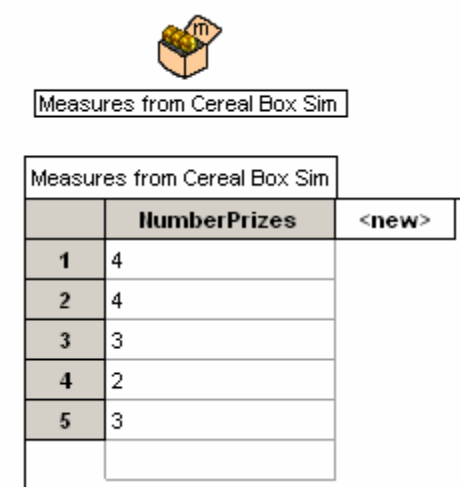


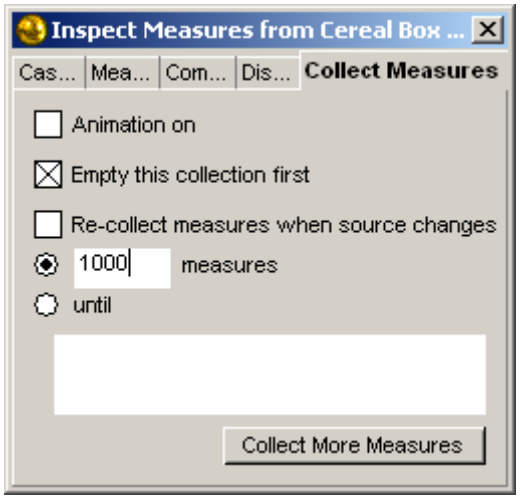
Open the **Inspector** for the collection, and create a measure called **NumberPrizes**. Edit the formula to:

uniqueValues(Prize)

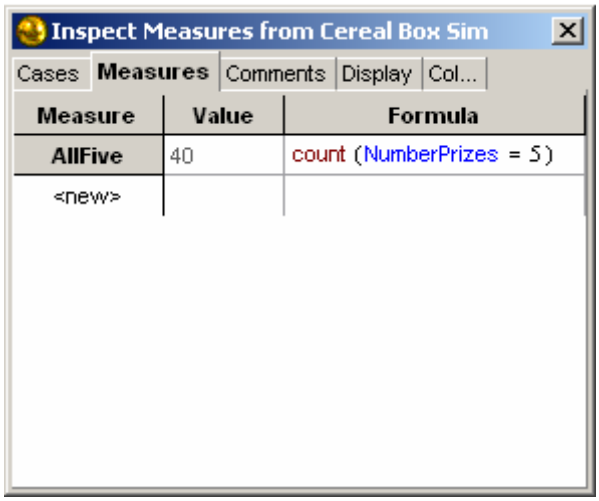
This function is found under **Functions/Statistical/One Attribute**, or, you can type it in. It will return how many unique values the attribute **Prize** contains. Run the simulation a few times, and note how the value of **NumberPrizes** changes.

Select **Collect Measures** from the **Analyse** menu, and collect the default five measures. Drag a case table to the workspace to see the result.





Open the **Inspector** for the **Measures from Cereal Box Sim** collection. Turn off animation, and reset the number of measures 1000. Collect 1000 measures.



to

Select the **Inspector** for the **Measures from Cereal Box Sim** collection, and create a measure called **AllFive**. Edit the formula to:

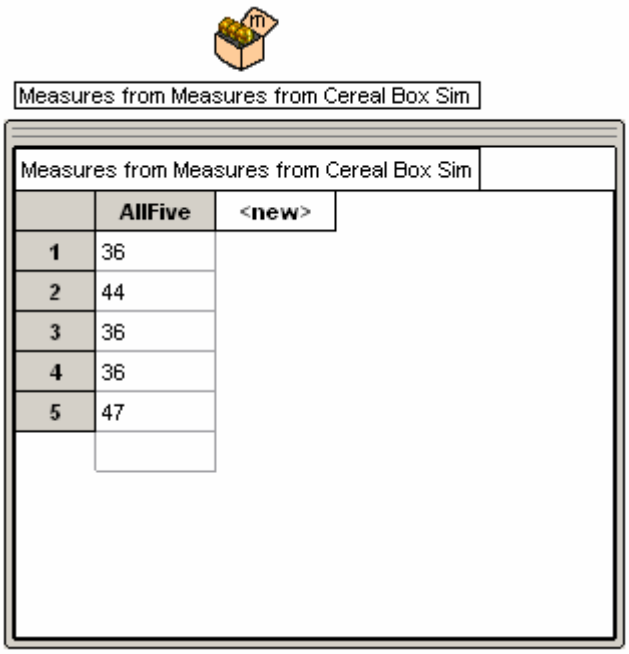
count(NumberPrizes = 5)

Note the value. Collect another 1000 measures, and note the value again. Do this a

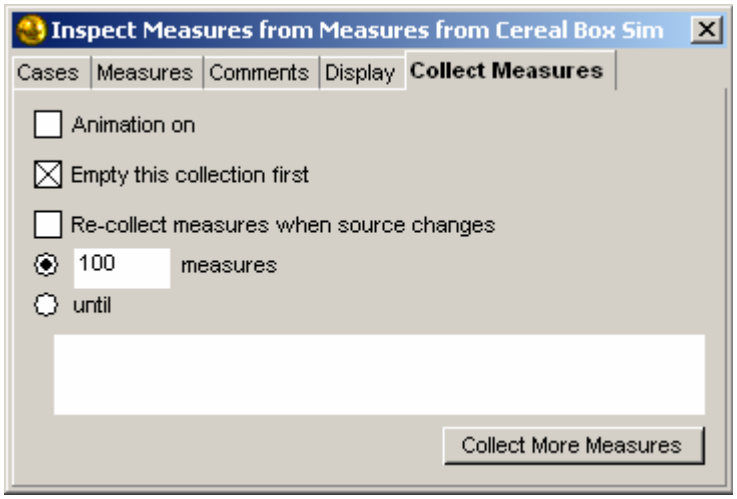
few

times.

Select the **Inspector** for the **Measures from Cereal Box Sim** collection, and select **Collect Measures** from the **Analyse** menu. Drag a case table to the workspace. Look at the default five measures. You have now instructed Fathom™ to "purchase" 1000 sets of five cereal boxes, 5 times, and note how many of each 1000 purchases resulted in obtaining all five prizes.



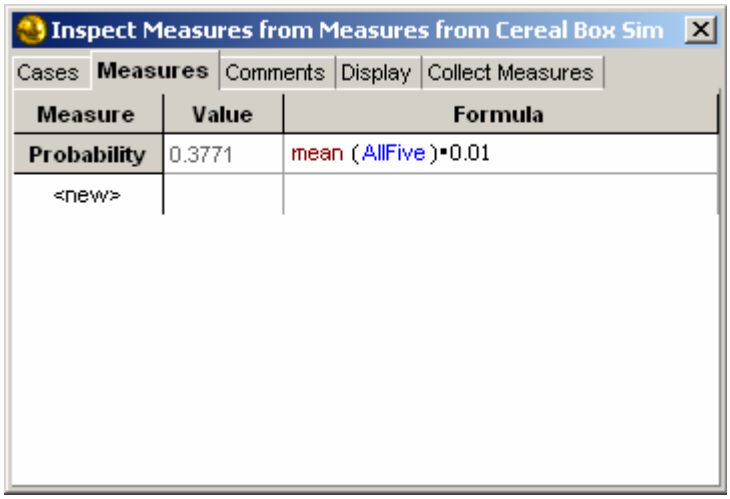
Open the **Inspector** for the collection **Measures from Measures from Cereal Box Sim**. Turn off animation, and increase the measures to 100.



Create a new measure called **Probability**. Edit the formula to:

mean(AllFive)*.01

Note that the probability of obtaining all five prizes with a purchase of five boxes of cereal is empirically about 0.4. Collect another 100 measures. Note any changes in the probability.



Select the collection **Cereal Box Sim**, and increase the number of cases to 6. Collect another 100 measures, and note the new probability.

Continue varying the number of boxes purchased at a time, and note the effect on the probability. Use Fathom™ to help you construct a table and a graph of **Probability** versus the number of boxes purchased.

Example 3: Waiting Times

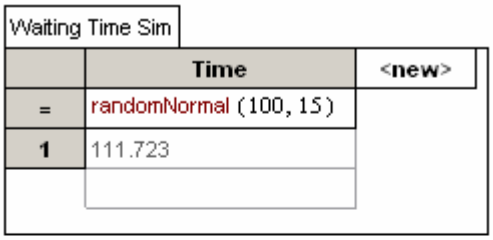
An aircraft waiting to takeoff on a runway with a high density of arriving traffic needs a "window" of 120 s to taxi and depart. The time between arriving aircraft is distributed normally, with a mean of 100 s and a standard deviation of 15 s. Use a simulation to empirically determine the expected waiting time before the aircraft can depart.

Solution: Although this problem can be done using the normal distribution in conjunction with tables or a calculator, there is value in seeing a simulation of the process.

Create a collection with one attribute called **Time**, and one case. Edit the formula to:

randomNormal(100,15)

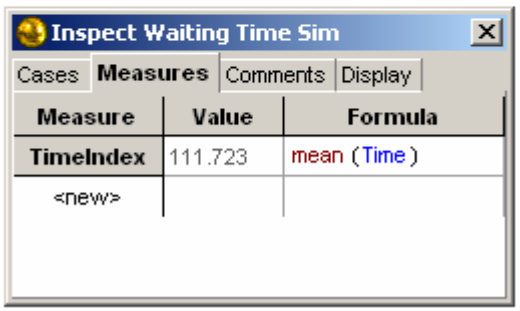
This random number function will generate random numbers selected from a normal distribution with a mean of 100 and a standard deviation of 15.



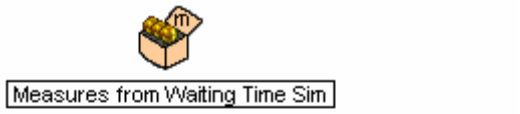
Use **CTRL-y** to generate a few random times. Does a window of 120 s or more seem to occur frequently or infrequently?

Open the **Inspector** for the collection. Create a measure called **TimeIndex**. Edit the formula to:

mean(Time)



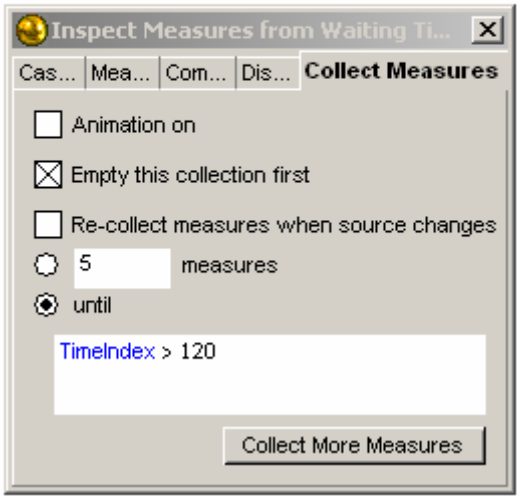
Note: Fathom™ does not permit stand-alone attributes as variables. Since the **Time** attribute contains only one case, it is always true that Time = mean(Time) for each run of the simulation.



Select the collection box, and then select **Collect Measures** from the **Analyse** menu. Select the collection **Measures from Waiting Time Sim**, and open the **Inspector**. Turn off Animation. Select the "until" radio button. Edit the formula to:

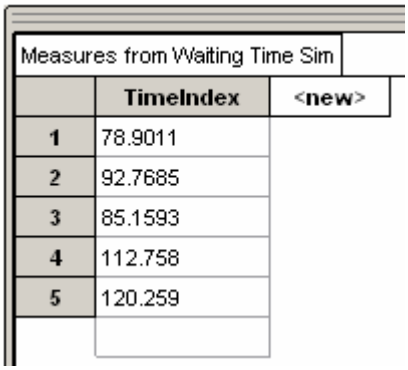
TimeIndex > 120

This will run the simulation until the first desired length of window occurs.



Ensure that the collection **Measures from Waiting Time Sim** is selected, and drag a new case table to the workspace. **Collect More Measures**. Note that the simulation will run until the **TimeIndex** exceeds the desired 120 s.

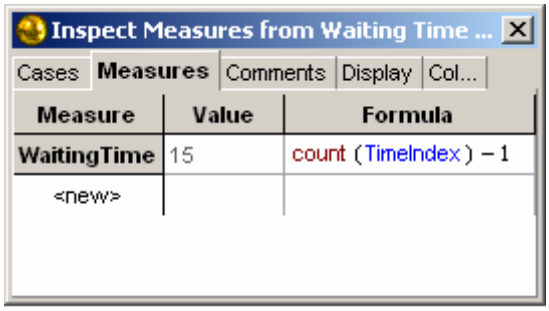
Collect More Measures. Note that the waiting time is now different. Run the simulation several times to get a feel for the distribution of the waiting times.



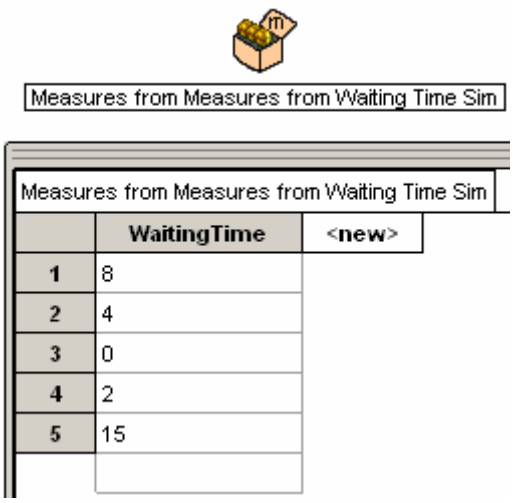
Select the **Inspector** for the collection **Measures from Waiting Time Sim**, and create a new measure called **WaitingTime**. Edit the formula t:

count(TimeIndex) – 1

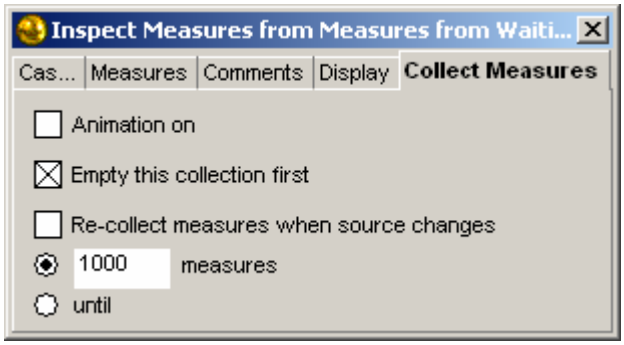
Select the collection **Measures from Waiting Time Sim**, and then select **Collect Measures** from the **Analyse** menu.



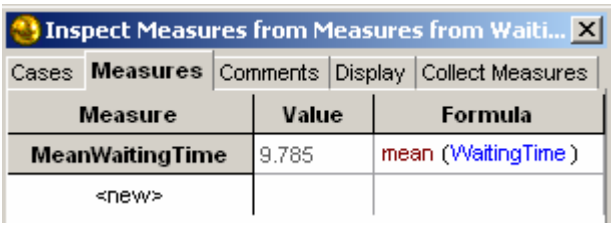
Select the collection **Measures from Measures from Waiting Time Sim**. Drag a new case table to the workspace. Note that Fathom™ has run the simulation five times, and made a list of the waiting times.



Open the **Inspector** for **Measures from Measures from Waiting Time Sim**. Turn off animation. **Collect More Measures**. Do this several times. Note how the waiting times change. Increase the number of measures to 1000. **Collect More Measures**.



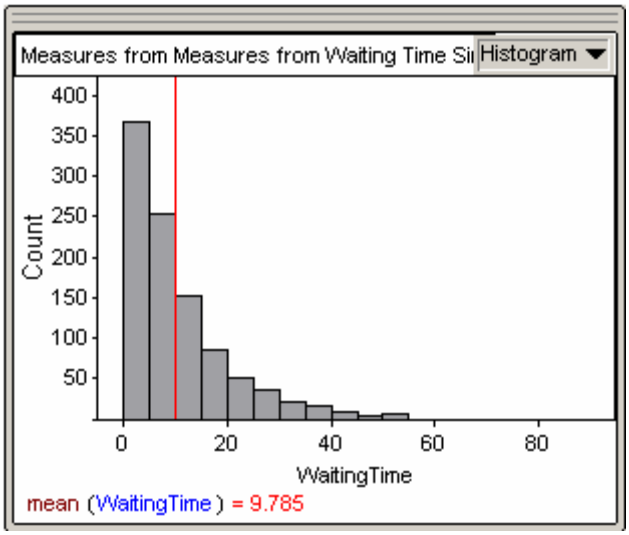
Select the **Measures** tab, and create a new measure called **MeanWaitingTime**.



Drag a new graph to the workspace, and drag the attribute **WaitingTime** to the horizontal axis. Change the type of graph to **Histogram**. Right-click on the graph, and select **Plot Value**. Edit the formula to mean(WaitingTime).

Run the simulation a few more times.

Edit the random number formula for a different mean and standard devation, say mean 90 and standard deviation 10. Run the simulation. How does this affect the waiting time?



Example 4: Hypothesis Testing

One of the more difficult concepts for students to grasp is hypothesis testing. We'll use Fathom™ to simulate an example. The Hippo Party enjoys 5% of the popular vote, according to the last few elections. A new party leader initiates a plan to make more people aware of the party platform. Just before the next election, he orders a poll of 100 voters, 10 of whom say they will support the Hippo party. Can he claim that the party has doubled its support?

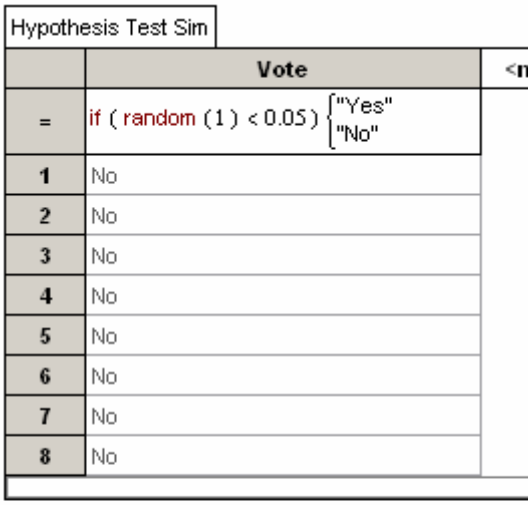
Solution: Create a collection called **Hypothesis Testing Sim**, with one attribute called **Vote**, and 100 cases for this attribute. Edit the formula to:

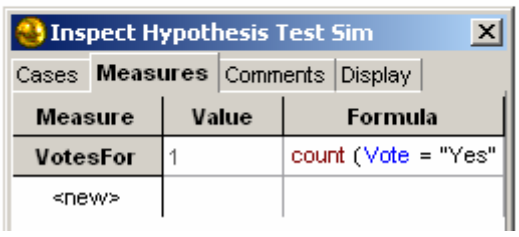
$$\text{if}(\text{random}(1)<0.05\begin{cases}\text{"Yes"}\\ \text{"No"}\end{cases}$$

Open the **Inspector** for this collection, and create a measure called **VotesFor**. Edit the formula to:

count(Vote = "Yes".

Use **CTRL-y** to run the simulation several times. Does a value of 10 or more seem to come up frequently, or rarely?





Select the collection box, and then select **Collect Measures** from the **Analyse** menu. Open the **Inspector** for the collection **Measures from Hypothesis Test Sim**. Turn off animation, increase the number of measures to 100, and **Collect More Measures**.

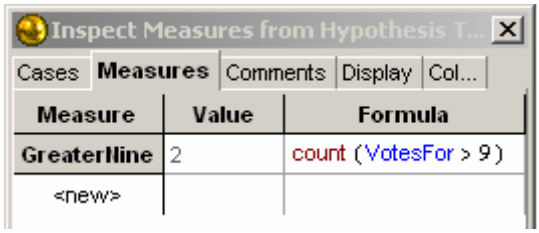
Measures from Hypothesis Test Sim

	VotesFor	<new>
1	5	
2	7	
3	4	
4	8	
5	3	
6	7	
7	7	
8	2	
9	4	
10	3	

Create a new measure called **GreaterNine**. Edit the formula to:

count(VotesFor>9)

Collect More Measures several times, and note how many times the number of votes for the Hippo party reaches 10 or more.



Select the collection **Measures from Hypothesis Testing Sim**, and then select **Collect Measures** from the **Analyse** menu. Open the **Inspector** for **Measures from Measures from Hypothesis Testing Sim**. Turn off animation, and collect more measures. Do this several times.

Can the party leader claim that there is no chance that the result of the original survey occurred by luck? Can she be 1% certain? 5% certain? 10% certain?

Exercise:

1) The owner of a casino suspects that the last shipment of dice that he received from his supplier were defective. He picked a pair at random, and rolled 10 sevens in 36 trials. Use Fathom™ to determine how sure he can be that the dice are defective. Employ the randomBinomial function to help you.

2) Each day Ms. Depoynt selects 10 students by lot, from a class of 25, to put homework solutions on the blackboard. Augusta believes that her name was omitted from the box, but does not wish to accuse Ms. Depoynt of this error without some evidence. She was not selected on any of the first four days in the week. Is this sufficient evidence? Use Fathom™ to simulate this problem.

3) While on a road trip, Tom suggested that he and his sister Amelia note down the last two digits on the licence plate of each approaching car. After 20 cars, they would look for a match. Amelia thought that the probability of a match would be low, but Tom thought it would be high. Use Fathom™ to simulate this situation. Then, adjust your simulation to determine what number of cars would result in a probability of 0.5 for a match.

This concludes the Fathom™ Tutorial.

For additional Fathom™ resources, visit the Key Curriculum Press web site at www.keypress.com.