Lab 1: exponential growth

NRES 470/670

Spring 2021

Contents

In this lab we will have an opportunity to do some simple population modeling using three different software packages: MS Excel, R, and InsightMaker. I encourage you to work in groups!

Nomenclature for Population Ecology

First of all, we need a symbol to represent the population size. This is N!

ΔN represents the change in population size, $N_{t+1} - N_t$

The famous "BIDE" equation is a way to break down ΔN into components.

$\Delta N = B + I - D - E$ (Eq. 1)

where B represents the number of births, I represents the number of immigrants, D represents the number of deaths, and E represents the number of emigrants.

If we ignore immigration and emigration, then the BIDE equation simplifies to:

$\Delta N = B - D$ (Eq. 2)

Now let's focus on B and D. The important thing to recognize is that the number of births and deaths in a population is not constant.

What does the number of births depend on?

What **is** more likely to be constant is the per-capita rate of producing offspring, or dying. Does this make sense?

Examples of per-capita rates:

- "*for every individual in the population now*, we expect 0.1 (10%) to die in the coming year"
- "*for every individual in the population now*, we expect 0.8 new juveniles to enter the population in the coming year"
- "*for every female in the population now*, we expect 1.1 offspring to be born into the population in the coming year"

- "*for every individual in the population now*, we expect 0.03 (3%) to be harvested in the coming year"
- "*for every juvenile in the population now*, we expect 0.34 (34%) to disperse away in the coming year"

These per-capita rates are often expressed as lower case letters. So b represents per-capita birth rate, and d represents per-capita death rate (see the first bullet point above for a way to think about per-capita deaths!).

To compute per-capita rates, you can just divide the total number of births (B) and deaths (D) by the population size N:

$b = \frac{B_t}{N_t}$ (Eq. 3)

–or, re-factored in terms of B–

$B_t = b \cdot N_t$

The letter t of course represents time. So the above equation could be described as follows: "the number of births at a given time is equal to the per-capita birth rate times the total population size at that time"

Similarly,

$D_t = d \cdot N_t$ (Eq. 4)

Okay, we're almost there.

If $\Delta N = B - D$ (Eq. 5)

then

$\Delta N = b \cdot N_t - d \cdot N_t$ (Eq. 6)

which is equal to

$\Delta N = (b - d) \cdot N_t$ (Eq. 7)

which could also be written:

$\Delta N = r \cdot N_t$ (Eq. 8)

Where r represents the difference between the per-capita birth and death and death rates. If r is positive, then births are greater than deaths and the population grows. If r is negative then deaths exceed births and the population declines.

We can use *calculus notation* to consider the instantaneous change in population size:

$\frac{\partial N}{\partial t} = r \cdot N$ (Eq. 9)

This is probably the most fundamental equation of population ecology.

If you integrate this equation across time from the initial time (t=0) to time t, you get an equation that describes the population size at any time t:

$N_t = N_0 e^{rt}$ (Eq. 10)

That is, population size at time t is equal to the population size at time zero (initial abundance, N_0) multiplied by the base of the natural logarithm (e) to the rt power.

There you have it! Now you can compute population growth and population size over time!

But wait, what about Lambda?? You've probably seen this term before to represent population growth rate.

The greek symbol lambda (λ), is used to represent the *finite rate of growth*, or $\frac{N_{t+1}}{N_t}$.

Lambda is what you multiply the current population size by to compute the population size in the next time step.

$N_{t+1} = N_t + B - D$ (Eq. 11)

$N_{t+1} = N_t + b_d \cdot N_t - d_d \cdot N_t$ (Eq. 12)

I am using the d subscript to indicate that the per-capita birth and death rates now represent **discrete growth** (these rates only apply once per time step and are not compounded continuously)

Here are some expressions to illustrate the difference between discrete and continuous rates:

b: "Babies enter into the population at a rate of 0.9 per adult female per year, but babies enter enter the population continuously throughout the year"
b_d: "Babies enter into the population at a rate of 0.9 per adult female per year, but the pool of babies enters the population simultaneously on April 1 each year"
d: "Approximately 23% of the population dies each year, but the deaths occur evenly throughout the year"
d_d: "Approximately 23% of the population dies each year, but all the deaths are assumed to occur on Jan 1 of each year"
r: "The population is growing constantly (continuously) at a rate of 15% per year (NOTE: this can be confusing- you will not have exactly 15% more individuals in the population one year later, but in fact you will have 16.2% more individuals! This should be more clear by the end of the lab)".
r_d: "The population size one year from now will be 15% higher than it is today"

$N_{t+1} = N_t + (b_d - d_d) \cdot N_t$ (Eq. 13)

$N_{t+1} = N_t + r_d \cdot N_t$ (Eq. 14)

$N_{t+1} = N_t \cdot (1 + r_d)$ (Eq. 15)

$N_{t+1} = \lambda \cdot N_t$ (Eq. 16)

Just so you know, you can convert between Lambda and the continuous rate of growth (r) easily: all you need to do is use the natural logarithm:

$e^r = \lambda$
$r = ln(\lambda)$

Okay let's start the lab! The first software we will use is our old friend, MS **Excel**!

Exponential growth in Excel

1. Open the Excel spreadsheet ExpGrowthExcel.xlsx. To download this file, right click on the link and select "*Save link as..*". In the first column, we have a timestep of 1 year for 30 years. In the second column, we have an initial population size (N_0) of 100 individuals. We also have a per-capita rate of increase (r) that is currently set at 0.1 (10%) per year. Assume for now that r represents r_d, or the *discrete rate of increase.*

2. To generate N_t for the remaining time steps, we need to apply our knowledge of population ecology. Specifically we need to apply equation 15 or 16, above (assuming discrete population growth). Do this by clicking in the empty N_1 cell (position B3), typing '=' in the cell, clicking on the N_0 cell (position B2), completing the equation and hitting enter. As you do this, you should see the equation you are creating appear in the equation editor.

3. You can fill the remainder of the cells using the same equation for the other time steps by clicking and dragging (or double clicking) the small square at the bottom of the N(2) cell, which appears when the cell is selected.

4. What happened? We are not seeing a growing population here- actually it seems quite flat! this is surely not what we want! Click on the N(3) cell to see what equation is being used to calculate the cell value. The equation is B3*(1+D4). The B3 part is correct - we want to calculate the N(3) population size using the N from the previous timestep - but the D4 part is incorrect. We always want to use the same r or λ which is always in the same cell. You can see that when you drag down an equation as we have done, Excel adds 1 for each row so that the equation references the same relative positions in the spreadsheet for each new cell you want to calculate. We like that Excel did that for N, but not for r or λ, so we can tell Excel to stay in the same row (row 3) for r (or λ) by inserting a dollar sign in our equation.

5. In the N(2) cell, edit the equation in the "functions bar" so that there is a dollar sign in D3 (i.e., 'D$3' instead of 'D3') (or just use the F4 shortcut).

6. Now drag the equation down again, and you should have a population size in row 32 of 1745 (representing the population size at year 30!).

NOTE: you can format the cells in column B to be whole numbers using the context menu (select column B » Format Cells » Number » Decimal places = 0)

7. Now we will plot our population against time. Select both columns of data, and select the *scatter plot* (or "line plot") option under the 'Insert' menu. A plot of *N* by Time will automatically appear. You can change the *r* value, the data and chart will automatically adjust.

Exercise 1

Please provide short answers to the following questions, and **provide your Excel spreadsheet to back up your answers**.

- ***Short answer (1a.)*** Apply equation 10 (above) to compute expected population size in year 30 (now you are assuming that the per-capita rate of growth in cell D3 represents *continuous* and not *discrete* growth). Do you get the same answer as you did for the discrete-time model (what we did together as a demo)? Why or why not? What about year 100 (what is the abundance after 100 years for both the discrete-time and continuous-time models? NOTE: computing abundance at time t in the continuous-time model is a *single calculation*- don't overthink this one. You could use a calculator instead of Excel if you really want! HINT: use the EXP function in Excel to raise *e* (base of the natural logarithm) to any power: to compute $e^{1.7}$ you type "=EXP(1.7)" in Excel. In general, if you don't know the syntax for a function in Excel, click on the button labeled "fx" and you can search for functions easily!

- ***Short answer (1b.)*** What are the *units* of the per-capita rate of population growth, *r*? HINT: The answer is in the Gotelli book

- ***Short answer (1c.)*** What if the time step for your simulation were in months instead of years? How Would *r* change (i.e., assuming r represents 0.1 growth per year, what is the growth per month)? Try it in Excel! Use both methods (eq. 15 vs equation 10) to compute population size in year 30 (that is, use both the discrete-time and continuous-time methods to compute the population size in year 30). (HINT: when you are applying equation 10, you should now use the monthly rate instead of the annual growth rate, and you should use months and not years to represent *t*) Do you see any difference in final population size between the two methods (discrete vs continuous growth)? Is the discrepancy between your discrete and continuous estimates for $N_{year=30}$ greater or less with a 1-month discrete time step than it was with a 1-year discrete time step? Why or why not? [in other words, is the discrete-time estimate closer to the continuous-time abundance estimate when you use months instead of years as the time step?]

- ***Short answer (1d.)*** What is the difference between continuous population growth and discrete population growth? Can you think of at least one case where continuous growth would be a more biologically realistic model than discrete growth? Can you think of at least one case where discrete growth would be a more biologically realistic model than continuous growth? Explain your answer.

Exponential growth in R

R is probably the most common software used by ecologists and conservation biologists for data analysis and simulation. There is a little bit of a learning curve with R, and I appreciate InsightMaker in many ways for making it easy to get started with programming and modeling, but R is much more powerful, much faster,

and more widely used than InsightMaker. For that reason, I will try to integrate R into this class as much as I can. We will do more with R when we get into data analysis! And you will do a LOT more with R in NRES 488!

SET UP

Open the R software from the program menu or desktop.

PROCEDURE

STEP I: Set up R and RStudio!

Go to website http://cran.r-project.org/. This is the source for the free, public-domain R software and where you can access R packages, find help, access the user community, etc.

Install Rstudio. This is a wrapper around R that makes R easier to use!

STEP II. Take some time to get familiar with R

Take a quick look at the R manual, Introduction to R. To jump into the deep end of the pool, try to implement the steps in Appendix A, located here. You might not understand everything right away, but you have the link, so you can return to this!

If you already have some R expertise, this is your opportunity to help your peers develop the level of comfort and familiarity with R that they will need to perform data analysis and programming tasks in this course.

Depending on whether you are already familiar with R, you may also find the remainder of this document useful as you work your way through the course (and there are many other good introductory R resources available online... let me know if there is one you particularly like and I will add it to the course website (Links page). As you work your way through this tutorial (on your own pace), please ask the instructor or your peers if you are uncertain about anything.

For a more detailed tutorial, see my "R Bootcamp" website: https://kevintshoemaker.github.io/R-Bootcamp/!

Set up the workspace

The first thing we usually do when we start an R session is we set up the workspace. That is, we load **packages** (extensions), assign key parameters and initialize variables. When you write code, always type in the "script" window in Rstudio. You can execute commands using command-enter or control-enter in Rstudio.

In this case, setting up the workspace is easy. We just need to define our parameter of interest - r -, and set up a **vector** to represent the years of interest.

We can store data in memory by assigning it to an "object" using the assignment operator **<-**. For example, this would assign the object "x" the value of 5.

```
x <- 5      #Assign the value of 5 to the object "x"
x           #Print the value of the object "x"
```

Note that any text after a pound sign (#) is not evaluated by R. These are *comments* and are intended to help you follow the code. You should always include comments in any code that you write- we humans tend to read and understand written language better than computer code!

Let's assign our per-capita population growth rate, r (but this could be called anything), and our initial population size to an object called **N0** (that is, population size at time 0), and the number of years to simulate.

```
r <- 0.1      #Assign the value of 0.1 to the object "r", or per-capita growth rate (discrete)
lambda <- 1+r
N0 <- 100     #Assign the value of 100 to the object "N0", or initial population size
nyears <- 30 #Assign the value of 30 to the object "nyears", or the number of time steps to simulate
```

If we want to know what the population size is at the next time step, we can simply multiply N0 by $(1+r)$.

```
N0 * lambda    #Multiplies the value stored in the object "N0" by 1 plus the value stored in the object
```

```
## [1] 110
```

How can we find the population size for the next 30 years? Let's first make an object that is a vector of years using the **seq()** or "sequence" function.

```
years <- seq(from=0, to=nyears, by=1)    #Creates a sequence of numbers from 0 to the value stored in th
years                                     #Print the value of the object "years" that you just created.
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
## [26] 25 26 27 28 29 30
```

Now, let's build a storage structure to store simulated population size over this time period

```
N <- numeric(nyears+1)    #Make an empty storage vector. The numeric() function takes the contents with
N                          #Prints the contents of the object "N".
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Run the simulation!

Then we can use a **for loop** (a very powerful computer programming trick) to automatically generate the population size for each of those years (note the similarity in the equation inside the for loop to Expression 1.15 in Gotelli).

```
N[1] <- N0                # The brackets [] are used to indicate the position of an element within a ve
lambda <- 1 + r           # (1 + r) is equal to lambda, the finite rate of growth.  This stores the res
for (i in 2:(nyears+1)){  # This for-loop will run through the line of code between the curly brackets
  N[i] <- lambda*N[i-1]   # This takes the [i - 1] element of "N", multiplies that element by the value
}                         # This ends the for-loop.
N                         # Now print the contents of the object "N".
```

```
##  [1]  100.0000  110.0000  121.0000  133.1000  146.4100  161.0510  177.1561
##  [8]  194.8717  214.3589  235.7948  259.3742  285.3117  313.8428  345.2271
## [15]  379.7498  417.7248  459.4973  505.4470  555.9917  611.5909  672.7500
## [22]  740.0250  814.0275  895.4302  984.9733 1083.4706 1191.8177 1310.9994
## [29] 1442.0994 1586.3093 1744.9402
```

Plotting

Let's plot our population size against time.

```
plot(N~years)    #This plot() function tells R to plot the y variable by the x variable. "N" is the y va
```
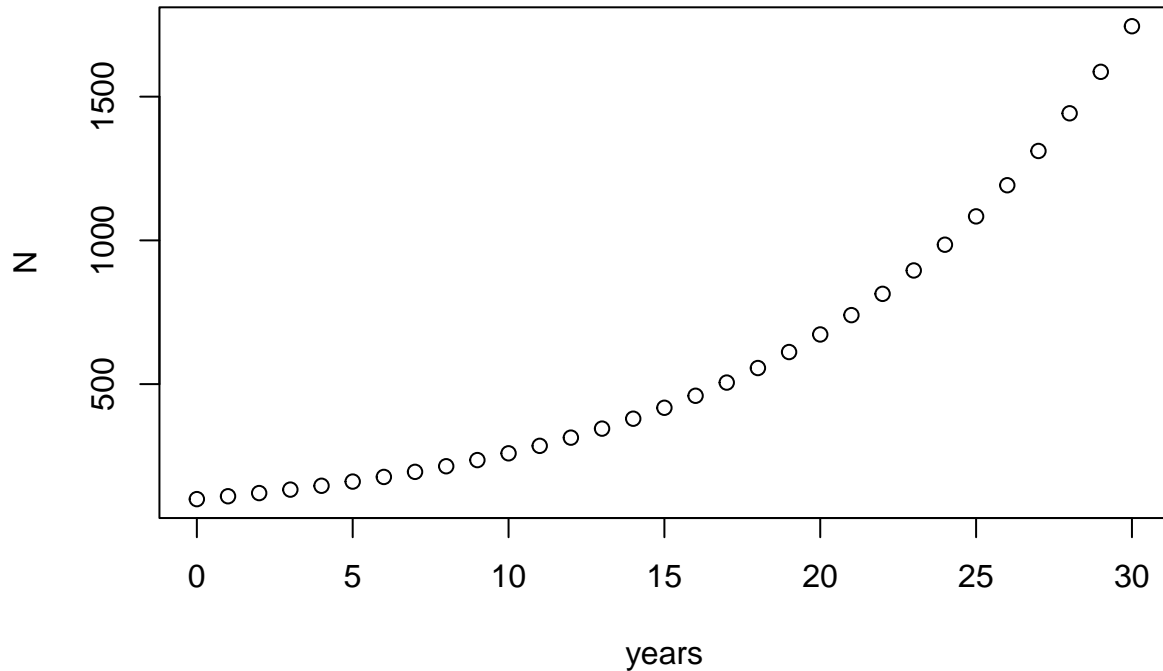

Exercise 2 (R-related problems)

Please provide short answers to the following questions, and **provide your R code to back up your answers**.

- ***Short answer (2a.)*** Tweak the above code to run for 100 years. Plot your results. What is the final population size?

- ***Short answer (2b.)*** Change r to 0.5 and run again for 100 years (and plot the results). What is the final population size now? Include your plot as part of your answer.

- ***Short answer (2c.)*** Try to tweak the value of r such that the final population size after 100 years is 1000. What is the value of r? After you solve it by trial and error, can you solve this problem analytically using Eq. 10 above? Show your calculations!

- ***Short answer (2d.)*** Change the value of r to -0.1. How long until the population goes extinct? Explain your answer. Include a plot of your results.

Exponential growth in InsightMaker

You should already have created a (free) account in insightmaker, and you should already know the basics about how to set up and run a model.

1. Click "Create New Insight" to start a new model (click "Clear this Demo" to clear the canvas and have an open workspace). Save the blank model by clicking the "Save" button.

2. Create a new [Stock] named *Population* using the "**Add Primitive**" button at top left ("Primitive" is just a computer-sciencey term referring to basic building blocks of a computer programming language). You can name the [Stock] and configure it in the properties tab at the right.
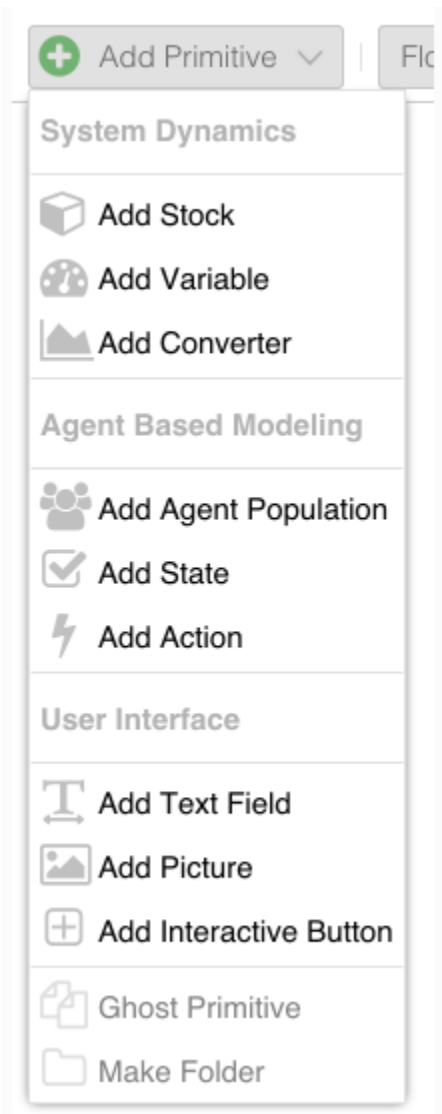
3. Change the Initial Value of *Population* to 100.

4. Create a new [Flow] going from empty space to the primitive *Population* (make sure the **Flow/Transitions** button is activated instead of **Links** at the top, hover over *Population* until an arrow appears, click and drag to create the [Flow], use the **Reverse Connection Direction** button to change the flow direction). Name the flow *Births*.

5. Create a new [Flow] going from *Population* to empty space. Name the flow *Deaths*.

6. The model diagram should now look something like this:
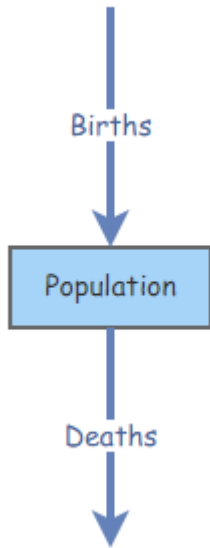
7. Change the **Flow Rate** property of *Births* to 0.16 ∗ [Population]. This represents the total number of individuals entering [Population] in each time step.

8. Change the **Flow Rate** property of *Deaths* to 0.10 ∗ [Population]. This represents the total number of individuals leaving [Population] in each time step.

Can you already tell whether this is a growing or declining population? (just a quick thought question, not part of the written lab!)

9. Run the model by clicking the **Simulate** button. We can change how the simulation is run by clicking the **Settings** button (left of Save). We can also change the settings of how the plot is created by clicking the **Configure** button within the simulation results window.

Exercise 3 (InsightMaker problems)

Please provide short answers to the following questions, and (when prompted) **provide your "Insights" to back up your answers**.

First, tweak the above model so that per-capita (discrete) birth rate (b_d) and death rate (d_d) are separate elements of the model (using the "Variable" primitive). Your Insight should look something like this:

To enable easy manipulation of these variables, change the **Show Value** Slider option of *Birth Rate* (in the properties window) to Yes. Change the **Slider Max value** to 1, the **Slider Min value** to 0, and the **Slider Step value** to 0.01. Do the same for *Death Rate* and *Population* (initial abundance N_0). For abundance, set the maximum value to 1000 and set the slider step size to 1 so we don't have fractional individuals! Now click on the white space of your model; you should now see the Birth Rate, Population and Death Rate sliders on the info tab. Change the slider values of the rates a few times, re-running the simulation each time. When you are confident that your model is working right, share it with your instructor and TA (save as a "public insight" and insert URL in the appropriate place in Top Hat).

Clone your previous Insight before you move on to the next problem (otherwise any changes you make will carry over to your answer to the previous problem!). The "Clone Insight" link is located in the upper right corner. In general, always clone your Insights after you have copied a link to an insight into your lab write-up. That way, you won't inadvertently change a model before your instructors have a chance to verify you did everything right!

- ***Short answer (3a.)*** Starting with a growing population, can you come up with two different scenarios in which *Population* is neither growing nor declining, by only changing one of the sliders from the starting conditions? Explain your answer.

The simulations that you ran above produced population growth curves that were very smooth, but we all know that populations don't grow in this manner because of **stochasticity** (randomness). Let's add some randomness to our vital rates! Change the **Show Value Slider** option of the Birth Rate primitive to No. Change the Equation to 0.1 + RandNormal(0, 0.1). The RandNormal() function generates a normally distributed random number with *mean (average) equal to the first argument* and *standard deviation equal to the second argument*. This means that in each time step we have a slightly different birth rate. Since we specified a *mean* of zero, we are not actually changing the average birth rate! Do the same for *Death Rate*. Run multiple simulations (at least 10) and look for patterns. Use the **Compare Results** tool to compare and provide output. When you are confident the model is right, add the URL for your InsightMaker model to Top Hat in the appropriate place.

- ***Short answer (3b.)*** Is it possible to have a declining population even when *Birth Rate* and *Death Rate* are the same? Is it possible to have a declining population when *Birth Rate* is greater than *Death Rate*? Explain your reasoning!

##Checklist for Lab 1 completion *Please submit all files (Excel file, and R code as text file) and responses via Top Hat. The InsightMaker models should be shared by saving your Insights as "public" and sharing the URL link with your instructors in Top Hat.

Due Feb. 1

- Top Hat short answers
  - **Exercise 1**
    * *Short answer (1a.)*
    * *Short answer (1b.)*
    * *Short answer (1c.)*
    * *Short answer (1d.)*
    * *Submit Excel file (1e.)*
  - **Exercise 2**
    * *Short answer (2a.)*
    * *Short answer (2b.)*
    * *Short answer (2c.)*
    * *Short answer (2d.)*
  - **Exercise 3**
    * *Short answer (3a.)*
    * *Short answer (3b.)*
- Excel file (submit in Top Hat)
  - **Exercise 1**
    * Your Excel file should show that you were able to successfully use formulas to calculate N_t for each time step (year and month) and show a plot of N by Time.
- R file (submit in Top Hat)
  - **Exercise 2**
    * Your R code should show that you were able to (a) adapt the given code to run for 100 years, and can display a plot of the results;
    * (b) change r to 0.5 and run for 100 years and plot the results;
    * (c) identify a value of r that gives a population size of 1000 after 100 years; and
    * (d) change r to -0.1 and run until the population goes extinct- and plot the results.
- InsightMaker models
  - **Exercise 3**
    * Your first model should show that you were able to alter the given model as instructed so that the simulation runs correctly. This should be shared via Top Hat (copy InsightMaker link in the appropriate place in Top Hat).
    * Your second model should include stochasticity in *Birth Rate* and *Death Rate*. This should be shared via Top Hat.