Means and Moments and Exploring New Plotting Techniques

Introduction

OBJECTIVE: Extend the concept of the moments of a density function of a solid to applications in probability and engineering.

What do means and multiple integrals have to do with probabilities? How can the method of moments be used to help determine whether or not an object will float in an upright position? These questions will be answered as you explore this module. You will also get to practice new and interesting ways to plot functions.

Technology Guidelines

NOTE: If you have just finished a module, restart *Mathematica* or close the *Kernel* before executing a new module.
TO OPEN CELLS, put your cursor on the right cell bracket and double click.
TO STOP AN EXECUTION
Select the *Kernel* pull-down menu and click on *Abort Evaluation.*
ORDER OF EXECUTION
Execute cells in the order given. Do not skip any Input cells within a given notebook.
SAVING NOTEBOOKS
You can save anytime to any directory you choose, and it is wise to save often.
However, before you do your final save, it is a good idea to delete all your output by selecting the
Delete All Output selection under the *Kernel* pull-down menu.
EXPERIENCING MAJOR PROBLEMS
Save if appropriate, then shut down *Mathematica* and start it up again.

Part I: Masses and Moments Applied to Probability Functions

In multivariable probability, probability density functions are defined over specified regions (domains) in the same way that density functions are defined over regions of a solid. These probability functions are designed so that the total "mass" is always 1. Probabilities are then defined to be the integrals of the density functions over a particular subregion of the domain. Consider the following example.

Suppose that a national fast-food outlet is interested in the joint behavior of the random variables x, defined as the total time between a customer's arrival at the store and departing from the service window, and y, the time that a customer waits in line before reaching the service window. Since x contains the time a customer waits in line, x must be greater than y. The relative frequency distribution of observed values of x and y can be modeled by the probability density function:
$f(x,y) = 0.04\, e^{-0.1x-0.3y}$ for $0 \le y \le x < \infty$ and 0 elsewhere, where x and y are measured in minutes.

In[1]:=

```
Clear[f, x, y]
```

```
f[x_, y_] := .04 e^(-.1x-.3y)
```

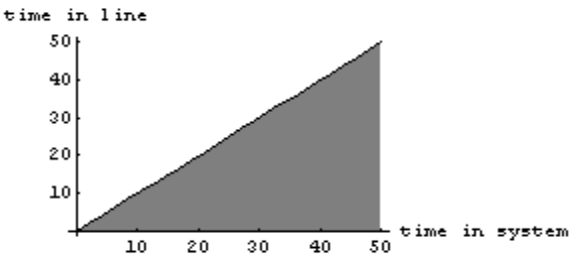
We examine the domain of this function. First we must load a package to assist us in graphing.

In[3]:=

```
<< Graphics`FilledPlot`
```

In[4]:=

```
FilledPlot[x, {x, 0, 50},
 AxesLabel -> {"time in system", "time in lin
```



We begin by verifying that the integral of the density function over the entire domain is 1.

In[5]:=

```
Integrate[f[x, y], {x, 0, ∞}, {y, 0, x}]
```

Out[5]=

```
1.
```

To compute the average or expected values of x and y over the domain, we find the first moments of the density function with respect to the x and y axes. The second integral is difficult to evaluate as x approaches ∞, so we substitute a large value of x for the upper bound.

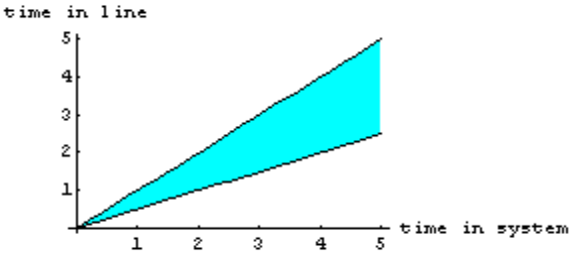
In[6]:=

```
Print["Expected time in system = ",
 Integrate[x f[x, y], {x, 0, ∞}, {y, 0, x}]," :

Print["Expected time waiting in line = ",
 Integrate[y f[x, y] // Simplify, {x, 0, 100000
 " minutes."]


Expected time in system = 12.5 minutes.


Expected time waiting in line =
 2.5 minutes.
```

Now we compute the probability that y is at least one-half as large as x. First we can draw the region over which we will integrate (the part for $x < 5$).

In[8]:=

```
FilledPlot[{x, 1/2 x}, {x, 0, 5},
 AxesLabel -> {"time in system", "time in lin
```



The probability of this event is the integral of the density function over this region.

In[9]:=

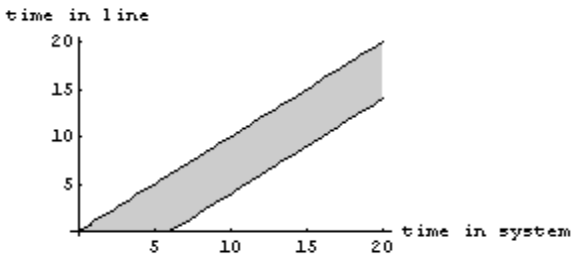
```
Print[
 "The probability that the time in line is a
   half as long as the time in the system is
 Integrate[f[x, y], {x, 0, ∞}, {y, 1/2 x, x}]]
```

```
The probability that the time
   in line is at least half as long
   as the time in the system is 0.2
```

If we want to know the probability that a customer spends no more than six minutes at the service window itself, we represent that quantity x - y, and we integrate over the portion of the domain where $x - y \leq 6$.

In[10]:=

```
p1 = FilledPlot[
        {x, x - 6}, {x, 6, 20}, Fills -> GrayLe
  DisplayFunction -> Identity];

p2 = FilledPlot[x, {x, 0, 6}, Fills -> GrayLev
  DisplayFunction -> Identity];

Show[p1, p2, DisplayFunction -> $DisplayFunct
 AxesLabel -> {"time in system", "time in lin
```



Looking at the domain, you can see that it is easier here to integrate with respect to x first.

In[13]:=

```
Print[
 "The probability that the time spent at the
   desk is no more than 6 minutes is ",
 Integrate[f[x, y], {y, 0, ∞}, {x, y, y + 6}]]
```

```
Integrate::gener :
 Unable to check convergence. More…
```

```
The probability that the time
   spent at the service desk is no
   more than 6 minutes is 0.451188
```

You Try It: Part I

Try your hand at computing the probability that the time at the service desk will be at least 8 minutes. Change the entries in red to the appropriate terms.

In[14]:=

```
Print["The probability is ",
 Integrate[f[x, y], {y, 0, ∞}, {x, y, y + 6}]]
```

```
Integrate::gener :
 Unable to check convergence. More…
```

```
The probability is 0.451188
```

Compute the probability that the time in line is no more than 30% of the time in the system. Change the entries in red to the appropriate terms.

In[15]:=

```
Print["The probability is ",
 Integrate[f[x, y], {x, 0, ∞}, {y, 1/2 x, x}]]
```

```
The probability is 0.2
```

Part II: Using Moments to Determine Acceptable Heights

Section 15.2, Extension of Exercise 43

Suppose that you have a buoy in the form of a solid hemisphere of radius 40 centimeters and uniform density that is topped with a solid cylinder of the same material and radius 40 centimeters. This buoy floats with the center of the hemisphere above the water surface. Determine the maximum height that the cylinder may attain that permits the buoy to continue floating upright. To picture this, we load two packages and plot the shape of the body.

In[16]:=

```
<<Graphics`ParametricPlot3D`
```

```
<<Graphics`Graphics3D`
```

In[18]:=

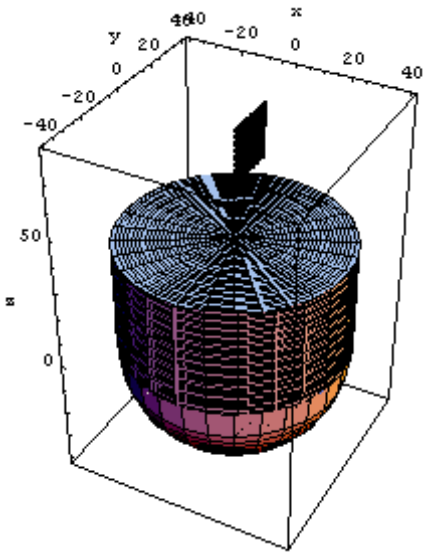
```
Clear[x, y, z, u, v, t, r, θ]
```

```
hemisplot = CylindricalPlot3D[{-√(1600 - r^2), {r
  {θ, 0, 2 π}, AxesLabel -> {x, y, z},
  DisplayFunction -> Identity];
```

```
cylpts = Table[{40 Cos[t], 40 Sin[t],
 u}, {t, 0, 2 Pi, Pi / 12},
 {u, 0, 40}];
```

```
cylplot = ListSurfacePlot3D[cylpts,
  DisplayFunction -> Identity];
```

```
topplot = CylindricalPlot3D[40, {r, 0, 40}, {θ
  AxesLabel -> {x, y, z}, DisplayFunction -> Id

flagpolepts = Table[{0, 0, t}, {t, 35, 80}];

flagpts = Table[{0, u, v}, {u, 0, 20}, {v, 60, 8

fp1 = ScatterPlot3D[flagpolepts, DisplayFunct

fp2 = ListSurfacePlot3D[flagpts, DisplayFunct

Show[hemisplot, cylplot, topplot, fp1, fp2,
 DisplayFunction :> $DisplayFunction];
```



You may assume that the flag on top is weightless.

The method of moments can help us. It can be determined that the buoyancy force exerted by the body of water due to the amount of water displaced will be in a direction through the point that would be the midpoint of the sphere, as long as the object is upright or tilted so that no part of the cylinder is immersed. Hence, if the center of gravity of the solid is at that point or closer to to the bottom of the hemisphere, the object will stay upright. If the center of gravity is in the region of the cylinder, the object will tilt.

We first determine the z-coordinate of the center of gravity of the hemisphere. Recall that the volume of a hemisphere with radius of r is $2/3\ \pi\ r^3$.

In[28]:=

```
volhemis = 2 / 3 π 40^3;

hemis = -√(1600 - x^2 - y^2);

zbar =
 Integrate[z , {x, -40, 40}, {y, -√(1600 - x^2) ,
   {z, hemis, 0}] / volhemis
```

Out[30]=

```
-15
```

The z-coordinate of the center of mass of the cylinder is $h/2$, so if we equate the first moment of the hemisphere to the first moment of the cylinder, we can solve for h. Since the radius is 40 here, the volume of the cylinder is $1600\,\pi\,h$.

In[31]:=

```
solh = Solve[-zbar * volhemis == h / 2 * (1600 π h)

Print["The height can be at most ", solh[[2,
 " or ", N[solh[[2, 1, 2]]], " centimeters."]
```

Out[31]=

```
{{h → -20 √2}, {h → 20 √2}}

The height can be at most
 20 √2  or 28.2843 centimeters.
```

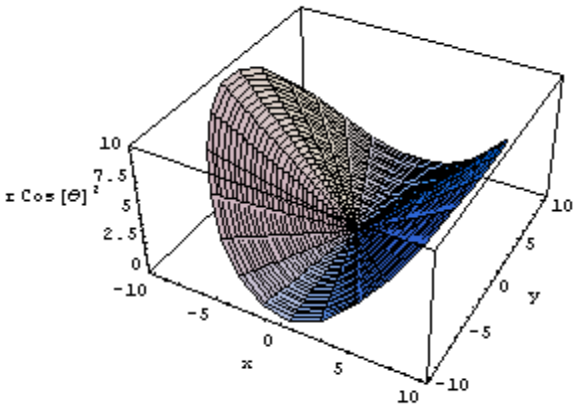
In general, if the radius were R, the maximum height of the cylinder could be at most $\frac{R\sqrt{2}}{2}$.

You Try It: Part II

Experiment with some of the 3-D plot functions presented in this section. First, just practice with a cylindrical plot. It is understood in these plots that z is a function of r and θ. Try some of your own functions. The package **ParametricPlot3D** is required for executing the cylindrical or spherical plots. Change the terms in red.

In[33]:=

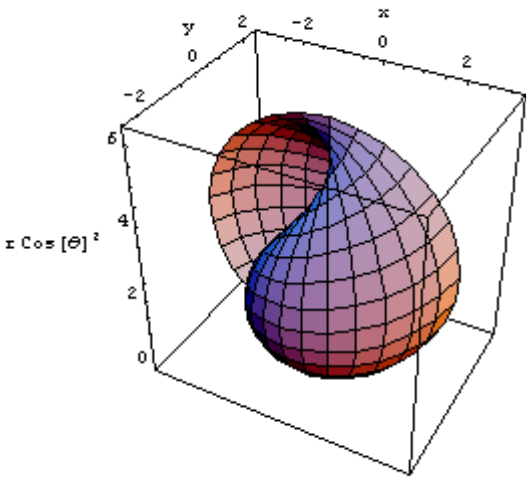
```
z = r Cos[θ]^2;
newcyl = CylindricalPlot3D[z, {r, 0, 10}, {θ, 0
  AxesLabel -> {x, y, z}];
```



You can create spherical plots to with the same package you used to create cylindrical plots. In these plots, it is assumed that ρ is a function of φ and θ. Try some of your own by changing the terms in red. Not everything will give satisfying results.

In[34]:=

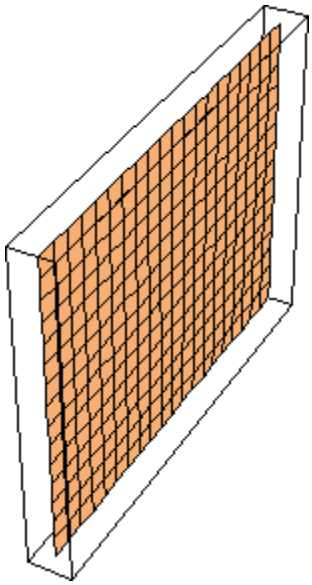
```
ρ = θ Cos[ϕ];
SphericalPlot3D[ρ, {ϕ, 0, π}, {θ, 0, 2 π},
 AxesLabel -> {x, y, z}];
```

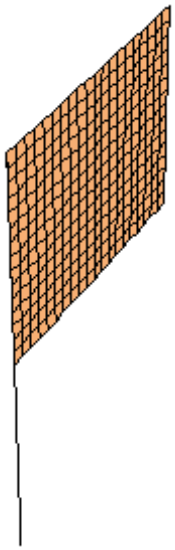


You may have noticed that to draw the flag we had to first generate points and then plot the points. The difference between the two was that the pole could be thought of as a line, whereas, the flag was more like a surface. Experiment with the following commands and draw something else by changing the terms in red. Remember to leave one parameter in the first expression and two parameters in the second experession.

In[35]:=

```
newflagpolepts = Table[{0, 0, t}, {t, 0, 30, .1
newflagpts = Table[{0, u, v}, {u, 0, 20}, {v, 1
fp1 = ScatterPlot3D[newflagpolepts];
fp2 = ListSurfacePlot3D[newflagpts];
Show[fp1, fp2, Boxed -> False, Axes -> False];
```





Created by *Mathematica* (April 3, 2005)