Lab_08

November 3, 2019

Probability for Data Science

UC Berkeley, Fall 2019

Ani Adhikari and Jim Pitman

[37]:
```
from datascience import *
from prob140 import *
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

1 Lab 8: Densities and `SymPy`

When you work with densities there is often some calculus involved. Sometimes, recognizing integrals as probabilistic quantities (probabilities, expectations, and so on) can help reduce the calculus.

But sometimes you just have to crank out the calculus. This can occasionally be messy. It would be nice to have some help. Fortunately, Python can help.

The `SymPy` library is a set of tools that help you do symbolic math on the computer. To see what this means, let's start with a simple numerical calculation of a familiar sort.

[38]:
```
x = 3
(2 * x + 1) ** 2
```

[38]:
49

What you can do in SymPy, assuming that you have explained to SymPy that x is a symbol, is to have `(2*x + 1) ** 2` appear as $(2x + 1)^2$. You can then expand it to get $4x^2 + 4x + 1$. SymPy does the math that you did long ago in algebra class, and renders the math in symbols. It gives you an exact symbolic answer, instead of numerical answer (that is often an approximation) in a particular case.

SymPy contains a large number of mathematical functions and operations, including integration and differentiation. This can be very handy when working with densities.

What you will learn in this lab:

- How to use SymPy
- How SymPy can help you understand properties of random variables that have densities
- How to implement the theory of densities (Parts 1-3) including change of variable
- How to implement the theory of joint densities (Part 4) including conditioning on continuous variables

This lab is intended not only to guide you through symbolic math on SymPy but also to develop your skills for working with densities. Pay close attention to each of the questions in Parts 2-5 as they will help you grasp the theory of continuous random variables.

1.0.1 Getting Started

We must import the library and add a line of code that makes the math appear in the way that it appears in the Prob 140 textbook.

[39]:
```
from sympy import *
init_printing()
```

Before you begin, it is **strongly recommended** that you skim Section 15.5 (Calculus in SymPy) of the textbook. It will give you a good sense of what you will do in the lab, and might help you answer some of the questions.

1.1 Instructions

Your labs have two components: a written portion and a portion that also involves code. Written work should be completed on paper, and coding questions should be done in the notebook. You are welcome to LaTeX your answers to the written portions, but staff will not be able to assist you with LaTeX related issues. It is your responsibility to ensure that both components of the lab are submitted completely and properly to Gradescope. Refer to the bottom of the notebook for submission instructions.

2 newpage

2.1 Part 1: The Operations

We will start with simple examples of the basic operations and syntax. For more, take a look at the SymPy Tutorial.

2.1.1 1a) Creating a Symbolic Variable

You are used to assigning names to numbers, arrays, and so on. In SymPy, we create a symbol using Symbol. Its required argument is a string that contains the symbol we want to use. We can assign that to a name, just as we have done before.

[40]:
```
# Create a symbol

x = Symbol('x')
x
```

[40]:
$$x$$

2.1.2 1b) Constructing a Symbolic Expression

Consider the math function $f(x) = 2x + 1$. We can create a SymPy expression equal to the right hand side, as follows.

[41]:
```
# An expression that is the right hand side
# of the definition of f(x)

2*x + 1
```

[41]:
$$2x + 1$$

[42]:
```
# Name the expression and display it

f = 2*x + 1
f
```

[42]:
$$2x + 1$$

It is important to make a distinction between the math function f and the SymPy expression f. Though f doesn't have an x in the name, it is equal to $f(x)$, the math function f evaluated at the point x.

You can create new expressions by using math operations.

[43]:
```
f ** 2
```

[43]:
$$(2x + 1)^2$$

You can ask SymPy to expand the square, to get an equivalent expression.

[44]:
```
expand(f ** 2)
```

[44]:
$$4x^2 + 4x + 1$$

2.1.3 1c) Evaluating an Expression at a Point

To evaluate $f(5)$ in math, we have take the expression $2x + 1$ and substitute the generic x with the specific value 5. So also in SymPy, we evaluate the symbolic expression f at the point 5 by using substitution.

[45]:
```
# Evaluate the expression at the point x = 5

f.subs(x, 5)
```

[45]:
$$11$$

Points themselves can be symbols. For example, SymPy recognizes pi as π. So we can evaluate the function f at π:

[46]:
```
f.subs(x, pi)
```

[46]:
$$1 + 2\pi$$

You can see that SymPy makes choices about exactly how to display expressions. All symbolic math systems make such decisions, but not always in the same way.

It is important to notice the difference between symbolic and numeric calculations such as those you perform using NumPy. For example, recall that np.pi is the NumPy expression for π, and run the cell below.

[47]:
```
f.subs(x, pi), f.subs(x, np.pi)
```

[47]:
$$(1 + 2\pi,\ 7.28318530717959)$$

An important aspect of the distinction is that the first expression is exact while the numerical calculation results in an approximate value.

2.1.4 1d) Integration

Integration is one of the most important operations in probability. If a random variable X has a density f_X, then $P(a < X < b) = \int_a^b f_X(x)dx$, $E(X) = \int_{-\infty}^{\infty} x f_X(x)dx$ (assuming the integral exists), and so on.

Let's learn how to integrate functions using SymPy.

The indefinite integral $\int f(x)dx$ can be displayed using Integral(f).

[48]:
```
# Display the indefinite integral

Integral(f)
```

[48]:
$$\int (2x + 1)\, dx$$

That's nice, but it would be even nicer if SymPy could actually do the integral, not just show it to us.

To make this happen, we are going to be a bit rude. We are going to tell SymPy to just doit().

[49]:
```
# Evaluate the indefinite integral

Integral(f).doit()
```

[49]: $x^2 + x$

This creates a new SymPy expression, which you can assign to a name as usual.

[50]:
```
F = Integral(f).doit()
F
```

[50]: $x^2 + x$

Definite integrals can be calculated in two ways, corresponding to the left and right hand sides of the example below.

$$\int_3^7 f(x)dx \quad = \quad F(7) - F(3)$$

You already know how to compute the right hand side using substitution and the expression representing the indefinite integral.

[51]:
```
# right hand side

F.subs(x, 7) - F.subs(x, 3)
```

[51]: 44

To display a definite integral, Integral requires an additional argument that specifies the variable being integrated and the two limits of integration:

- (variable_name, lower_limit, upper_limit)

At the moment we are working with a function of just one variable, so it seems unnecessary to provide the name of the variable. But the name will become important as soon as we start double integration.

[52]:
```
# left hand side
# Display the definite integral

Integral(f, (x, 3, 7))
```

[52]: $\int_3^7 (2x + 1)\, dx$

To evaluate the integral, use doit() as before. You will get the same answer as you did earlier.

[53]:
```
# Evaluate the definite integral

Integral(f, (x, 3, 7)).doit()
```

[53]:
$$44$$

The standard operations on integrals can be performed just as you would expect. At each cell below, pause and read the expression carefully. Then run the cell and examine the output just as carefully.

[54]:
```
# Display a sum of integrals

Integral(f, (x, 3, 4)) + Integral(f, (x, 4, 7))
```

[54]:
$$\int_{3}^{4} (2x+1)\, dx + \int_{4}^{7} (2x+1)\, dx$$

[55]:
```
# Evaluate it

( Integral(f, (x, 3, 4)) + Integral(f, (x, 4, 7)) ).doit()
```

[55]:
$$44$$

[56]:
```
# Display the integral of a new function of x

Integral((x-7)*f, (x, 2, 10))
```

[56]:
$$\int_{2}^{10} (x-7)\,(2x+1)\, dx$$

[57]:
```
# Evaluate the integral

Integral((x-7)*f, (x, 2, 10)).doit()
```

[57]:
$$-\frac{56}{3}$$

You could have evaluated each of these integrals without displaying it first. But it is strongly recommended that you do display the integral first, so that you are confident that the right quantity is being calculated.

2.1.5 1e) Differentiation

We started with $f(x) = 2x + 1$:

[58]:
```
f
```

[58]:
$$2x + 1$$

So

$$\frac{d}{dx}f(x) = 2$$

To get the derivative in SymPy, use diff.

[59]:
```
diff(f)
```

[59]:
2

Since we defined F as the integral of f, the Fundamental Theorem of Calculus says that

$$f(x) = \frac{d}{dx}F(x)$$

[60]:
```
# Differentiate F to get back f

diff(F)
```

[60]:
$2x + 1$

[61]:
```
# You can differentiate F twice, just for fun
# That's the same as differentiating f

diff(diff(F))
```

[61]:
2

2.1.6 1f) Solving an Equation

Equations can have numerous different forms, so symbolic math systems require consistent ways of specifying the equations to be solved. That way they don't have to have separate functions for solving all the different kinds of equations. Let's see how to solve equations in SymPy.

We will restrict ourselves to equations involving a single unknown. For a sense of the variety of the equations that SymPy can solve, see the Solver module reference.

The general form of an equation in x is

$$h(x) = g(x)$$

for two functions h and g. Solving this equation is the same as solving the equation

$$h(x) - g(x) = 0$$

which can be written as $d(x) = 0$ for $d = h - g$.

So every equation can be written with 0 as the right hand side. This helps simplify syntax. That is why Solver in SymPy requires 0 as the right hand side of the equation. That is, the equation should be in the form $d(x) = 0$.

The call is solve(d, x) where d is the expression on the left hand side and involves the symbol x.

Solving $d(x) = 0$ means finding the value of x for which the value of $d(x)$ is 0. The instruction solve(d, x) tells the computer to find the value of x for which the value of the expression d is 0.

For example, to solve $5x = 15$, SymPy finds the value of x that solves the equation $5x - 15 = 0$.

[62]:
```
solve(5*x - 15, x)
```

[62]:
$[3]$

Some equations have multiple solutions, so the output is a list of all the solutions. In the case of the equation $5x - 15 = 0$, the list contains just one element, 3.

When you want to work with the element, you can access it by specifying its index.

[63]:
```
solve(5*x - 15, x)[0]
```

[63]:
3

Here are the solutions to the equation $x^2 + 5x + 6 = 0$. In an algebra class you will have learned to factor the left hand side to get $(x+2)(x+3) = 0$ and hence the two solutions $x = -3$ and $x = -2$.

[64]:
```
solve(x**2 + 5*x + 6, x)
```

[64]:
$[-3, -2]$

You can use previously defined expressions as part of the expression in solve. For example, we previously defined f as the right hand side of $f(x) = 2x + 1$. We can solve the equation $f(x) = 9$ as follows.

[65]:
```
solve(f - 9, x)
```

[65]:
$[4]$

2.1.7 1g) Inverse

Finding the inverse of a function f at a point y means finding the value of x such that $f(x) = y$.

This just means writing x in terms of y. The formal math move is to solve for x in the equation $f(x) = y$.

Because we are going to solve the equation using SymPy, we will solve for x in the equation $f(x) - y = 0$.

To execute this plan in SymPy, we will first have to declare y as a symbol, and then solve the equation.

[66]:
```
y = Symbol('y')
solve(f - y, x)
```

[66]:
$$\left[\frac{y}{2} - \frac{1}{2}\right]$$

Does the answer make sense? Remember that $f(x) = 2x + 1$ and solve the equation the old-fashioned way.

$$y = 2x + 1 \iff y - 1 = 2x \iff x = \frac{y-1}{2} = \frac{y}{2} - \frac{1}{2}$$

It works. As before, you can access the inverse by specifying its index in the list.

[67]:
```
f_inverse = solve(f-y, x)[0]
f_inverse
```

[67]:
$$\frac{y}{2} - \frac{1}{2}$$

3 newpage

3.1 Part 2: Working with Densities

Now that you have an idea of what you can do with SymPy, it's time for some probability theory.

In Prob 140, calculus is used almost exclusively when working with densities. Later in the course it will be used for some optimization as well, but density calculations are its main use.

In this part of the lab, you will use SymPy to work with a very simple density. That way, you will be able to check by hand that SymPy is getting the right answers. In subsequent parts of the lab, the densities will get more complicated. Treat this part of the lab as a warm-up.

3.1.1 2a) A Density on the Unit Interval

Let X have density $f_X(x) = 6x^5$ for $0 < x < 1$.

To work with this density, you have to start by creating a symbol representing x. Notice that Symbol takes an optional argument that allows you to specify that a variable is positive. This can be useful, as you will see below.

Complete the cell by using x to construct an expression f_X that is equal to $f_X(x)$.

[68]:
```
x = Symbol('x', positive=True)
f_X = 6*x**5
f_X
```

[68]:
$6x^5$

Is the function f_X in fact a density?

What do you need to check to answer this? Explain in the cell below and use the cell below that for calculation.

Your answer here

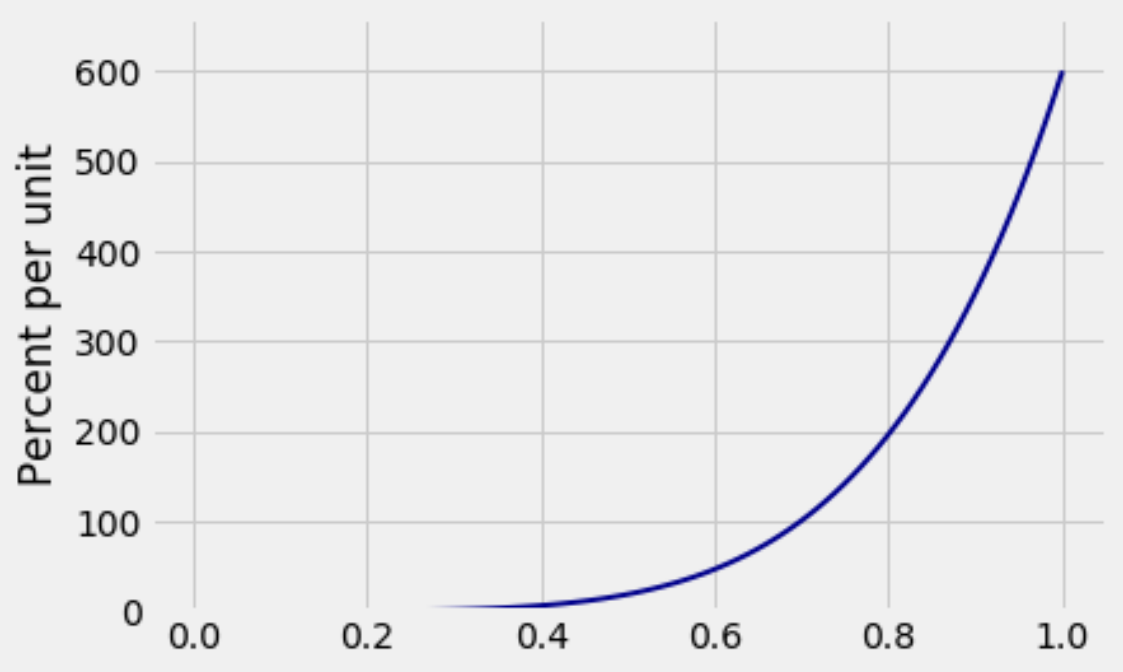
[69]:
```
print(Integral(f_X, (x, 0, 1)).doit())
print(Integral(f_X, (x, 0, 1)).doit()==1)
```

```
1
True
```

The prob140 library includes a plotting function Plot_continuous that plots a line graph of a probability density. The arguments are a list containing the endpoints of the interval over which to draw the graph, and a probability density which can be specified by a SymPy expression.

Run the cell below to plot the graph of f_X on the unit interval.

[70]:
```
Plot_continuous([0, 1], f_X)
```



Before you go further, please review Part **1d** on Integration very carefully.

3.1.2 2b) Finding Probabilities

Find $P(X > 0.75)$ by integrating the density appropriately. The code in the first cell below should display the integral and in the next cell you will evaluate that integral.

[71]:
```
# Display the integral for P(X > 0.75)

Integral(f_X)
```

[71]:
$$\int 6x^5 \, dx$$

[72]:
```
# Evaluate P(X > 0.75)
#1-Integral(f_X, (x, 0, 0.75)).doit()
Integral(f_X, (x, 0.75, 1)).doit()
```

[72]:
$$0.822021484375$$

In the cell below, fill in the blanks with numbers: The event $|X - 0.7| < 0.25$ is the same as the event $X \in (\_\_\_\_\_, \_\_\_\_\_)$.

Your answer

First blank: 0.45

Second blank:0.95

Find $P(|X - 0.7| < 0.25)$.

[73]:
```
# P(|X - 0.7| < 0.25)
Integral(f_X, (x, 0.45, 0.95)).doit()
#Integral(f_X, (x, 0.45, 0.95))
```

[73]: 0.726788125

3.1.3 2c) Finding Expectations

Read Section 15.3 before you get started.

Find $E(X)$.

[74]:
```
# E(X)
#Integral(x*f_X, (x, 0,1))
expectation_X = Integral(x*f_X, (x, 0,1)).doit()
expectation_X
```

[74]: $\frac{6}{7}$

Find $SD(X)$.

[75]:
```
# SD(X)
#Integral((x**2)*f_X, (x, 0,1))
variation_X=Integral((x**2)*f_X, (x, 0,1)).doit()-expectation_X**2
variation_X
standardiv_X = variation_X**(1/2)
standardiv_X
#np.sqrt(variation_X)
```

[75]: 0.123717914826348

Just because you can, find $E(\log(X))$. SymPy recognizes log. Use the first cell below to display the appropriate integral and use the next cell to evaluate it.

[76]:
```
# Display the integral for E(log(X))
Integral(log(x)*f_X, (x, 0,1))
#Integral(...)
```

[76]: $\int_0^1 6x^5 \log(x)\, dx$

Notice that SymPy writes its integrand in a different order than we do in class. That's OK; it doesn't affect the integral.

[77]:
```
# Numerical value of E(log(X))
Integral(log(x)*f_X, (x, 0,1)).doit()
```

[77]:
$$-\frac{1}{6}$$

Why is the answer negative?

The answer is negative because we are find the log of X and since x is smaller than one, the log(x) value is going to be negative, every f_X value is positive. Thus the value of the whole intergal is going to be negative.

Find $E(\sin^2(X))$, first displaying the integral and then evaluating it. Use sin for sine.

[78]:
```
# Display the integral for E(sin_squared(X))
Integral((sin(x)**2)*f_X, (x, 0,1))
```

[78]:
$$\int_0^1 6x^5 \sin^2(x)\, dx$$

[79]:
```
value_of_integral = Integral((sin(x)**2)*f_X, (x, 0,1)).doit()
value_of_integral
```

[79]:
$$-7\sin^2(1) - \frac{21\sin(1)\cos(1)}{2} - \frac{13\cos^2(1)}{4} + \frac{45}{4}$$

Sometimes SymPy can collect terms and simplify its answers. If you are good at half-angle formulae in trigonometry, you can check by hand that the simplification below is correct. But you don't need to do that for the lab.

[80]:
```
simplify(value_of_integral)
```

[80]:
$$-\frac{21\sin(2)}{4} + \frac{15\cos(2)}{8} + \frac{49}{8}$$

3.1.4 2d) Change of Variable

Let g be a monotone differentiable function and let $V = g(X)$. The change of variable formula for densities says that the density of V is given by

$$f_V(v) = \frac{f_X(x)}{\left|\frac{d}{dx}g(x)\right|} \qquad \text{evaluated at } x = g^{-1}(v)$$

See Section 16.2 of the textbook for the derivation and examples. Each time you use the formula, the main steps to find the density are:

- Find the possible values of $V = g(X)$.
- Find the inverse of g.
- Find the derivative of g.

- Divide the density of X by the derivative of g (if the derivative is negative, use its absolute value instead).
- Evaluate this quotient at the inverse of g.

Use the formula to find the density of the area of a disc that has radius X. That is, find the density of $V = \pi X^2$.

Start by constructing a SymPy expression g defined by $g(x) = \pi x^2$. SymPy recognizes pi as π. Remember that you have already declared x.

[81]:
```
g = Symbol('x', positive=True)
g=pi*(x**2)
g
```

[81]: πx^2

It is worth keeping in mind that you have two representations of π:

- pi produces the symbol π, using SymPy
- np.pi produces a numerical approximation to π, using NumPy

[82]:
```
pi, np.pi
```

[82]: $(\pi,\ 3.141592653589793)$

As always, when you are specifying a distribution, start with the possible values. What are the possible values of the random area V?

The possible values of the random area V is between (0,pi)

Now find all the elements of the right hand side of the change of variable formula, one by one.

First find the function g^{-1} by using the equation $g(x) = v$ to write x in terms of v. That is, solve for x in the equation $g(x) = v$, or, equivalently, in the equation $g(x) - v = 0$.

Review Parts **1f** and **1g** carefully before you proceed.

[83]:
```
v = Symbol('v', positive=True)
g_inverse = solve(g-v,x)
g_inverse
```

[83]: $\left[\frac{\sqrt{v}}{\sqrt{\pi}}\right]$

Notice that SymPy lists just one inverse -- the positive one -- because you specified that x is positive. The other inverse is $-\frac{\sqrt{v}}{\sqrt{\pi}}$ but it is not a permitted value of x because it is negative.

Run the cell below to extract the inverse from the list.

[84]:
```
g_inverse = g_inverse[0]
g_inverse
```

[84]:
$$\frac{\sqrt{v}}{\sqrt{\pi}}$$

The change of variable formula has another factor: the derivative in the denominator. Because g is an increasing function, its derivative is positive and you won't need the absolute value in the formula.

Use SymPy to find the derivative of g, so that the expression deriv_g is equal to $\frac{d}{dx}g(x)$.

Do not find the derviative by hand and simply assign that expression to deriv_g. See **1e** for how to differentiate in SymPy.

[85]:
```
deriv_g = diff(g)
deriv_g
```

[85]:
$$2\pi x$$

Now apply the change of variable formula to find f_V, the density of V. **In the comment line, enter the possible values of V.**

[86]:
```
"""For v in the interval (0,pi), the density of V at the point v is:"""

f_V = (f_X / deriv_g).subs(x,g_inverse)
f_V
```

[86]:
$$\frac{3v^2}{\pi^3}$$

Check that the function f_V is a density.

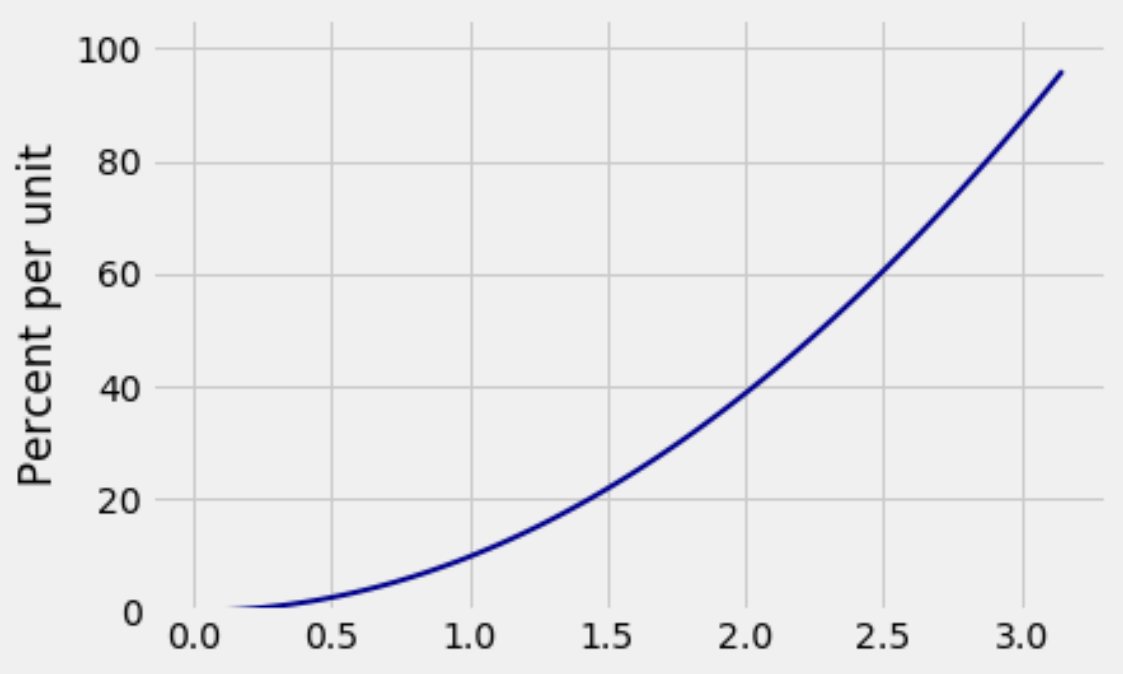
[87]:
```
print(Integral(f_V, (v, 0,pi)).doit()==1)
Integral(f_V, (v, 0,pi)).doit()
```

```
True
```

[87]:
$$1$$

Plot the graph of the density of V. Recall from Part **a** that the first argument of Plot_continuous is a list of two elements: the left and right ends of the interval over which to plot the graph. They must be numerical expressions, not symbolic.

[88]:
```
Plot_continuous([0, 3.15], f_V)
```



4 newpage

4.1 Part 3: Transforming an F Distribution

The density you worked with in Part 2 was deliberately chosen to be simple. The real value of SymPy is that without much difficulty you can use essentially the same code to handle more complicated density functions.

In this part of the lab you will start with one of the famous distributions of statistical inference, confusingly named the F distribution. It's not named for our old friend the cdf. It is named for our other old friend Sir Ronald Fisher, the extraordinary scientist whose contributions include what is now the standard method of testing statistical hypotheses. You might recall from Data 8 that it was Fisher who found 5% to be a "convenient" cutoff for the P-value, which then became set in stone as the cutoff for statistical significance.

The F distribution arises as the distribution of the ratio of two independent gamma random variables (apart from a constant multiplier). The ratio arises in tests of whether three or more random samples come from the same underlying distribution.

For the purposes of this lab, the F distribution is just an ordinary distribution on the positive numbers. It has two parameters, which we will call n for "numerator" and d for "denominator". And its density has a rather intimidating formula.

4.1.1 The $F_{n,d}$ Density

The gamma function of mathematics is denoted Γ (that's the upper case Greek letter gamma) and is defined as an integral. Both its domain and range are the positive numbers. In your current Homework assignment you are examining the definition and properties of the gamma function. In this lab, you will only need one of those results:

For positive integer n, $\Gamma(n) = (n-1)!$.

Now for the definition of the F density.

Let n and d be positive integers. The random variable X has the $F_{n,d}$ distribution if the density of X is

$$f_X(x) = \frac{\Gamma\left(\frac{n}{2} + \frac{d}{2}\right)}{\Gamma\left(\frac{n}{2}\right)\Gamma\left(\frac{d}{2}\right)} \left(\frac{n}{d}\right)^{\frac{n}{2}} x^{\frac{n}{2}-1} \left(1 + \frac{n}{d}x\right)^{-\frac{n+d}{2}}, \qquad x > 0$$

That looks awful. But it isn't, really. Let's see why.

4.1.2 3a) The Constant

As with many densities, the part of the formula that looks most impressive is actually the least interesting as far as the shape of the density is concerned. It's just the constant of integration: the numerical factor that makes the density integrate to 1.

To evaluate the constant you will need to evaluate the gamma function numerically. The SymPy function gamma can be used for this. The function also has another use but you will not need that for this lab.

[89]:
```
gamma(4)
```

[89]:
6

[90]:
```
gamma(4.5)
```

[90]:
11.6317283965674

[91]:
```
gamma(5)
```

[91]:
24

[92]:
```
r = Symbol('r', positive=True)
gamma(r)
```

[92]:
$\Gamma\left(r\right)$

Start out by evaluating the constant in the $F_{n,d}$ density. Define a Python function constant_F that takes n and d as its arguments and returns the normalizing constant in the density above.

[93]:
```
def constant_F(n, d):
    return gamma(n/2 + d/2) / (gamma(n/2) * gamma(d/2)) * (n/d)**(n/2) * x**((n/
↪/2)-1) * (1 + (n/d)*x)**(-(n+d)//2)
```

Work out (by hand or by mental math) the constant when $n = d = 4$, and check that your function returns the right value.

[94]:
```
constant_F(4, 4)
```

[94]:
$$\frac{6.0x}{(1.0x+1)^4}$$

As another check, work out the constant when $n = 2$ and $d = 4$, and check that your function returns the right value.

[95]:
```
constant_F(2, 4)
```

[95]:
$$\frac{1.0}{(0.5x+1)^3}$$

4.1.3 3b) The $F_{6,4}$ Density

As a numerical example, let X have the $F_{6,4}$ density, which we will call f_X. Construct a SymPy expression f_X that is equal to $f_X(x)$.

[96]:
```
x = Symbol('x', positive=True)
f_X = constant_F(6,4)
f_X
```

[96]:
$$\frac{40.5x^2}{(1.5x+1)^5}$$

That's not very scary at all. It's just a ratio of two polynomials, because $n = 6$ and $d = 4$ are both even.

Check that f_X is a density. Here you get to use something cute:

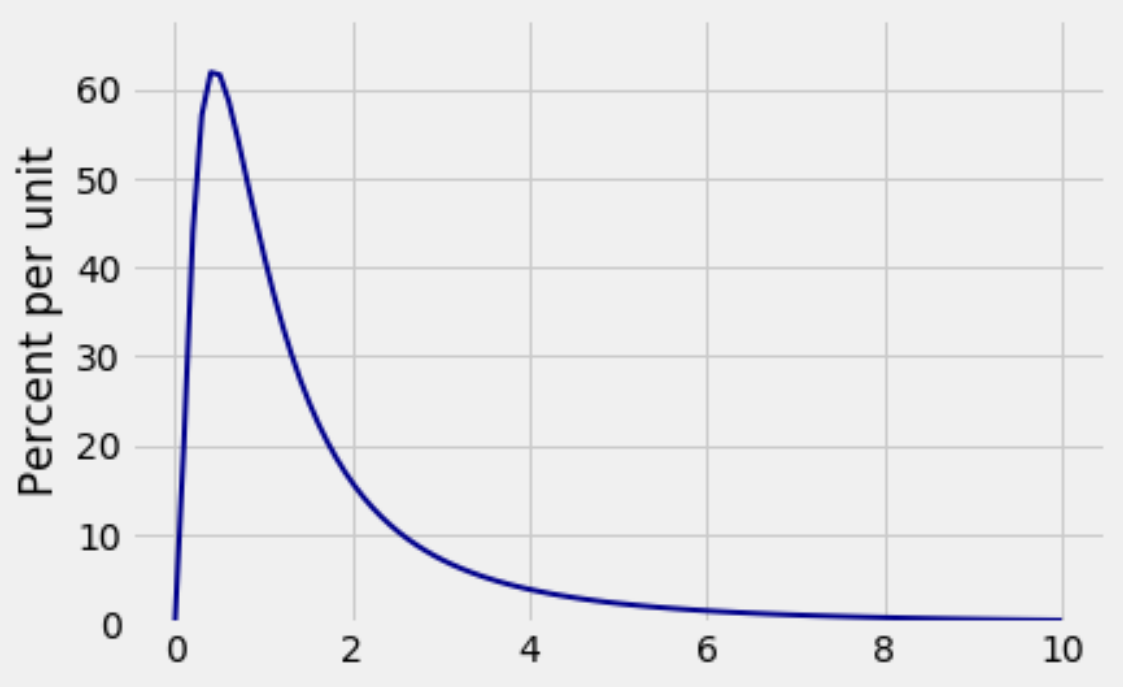
- As the symbol for infinity, SymPy uses two lower case letter o's side by side, because oo looks like ∞.

[97]:
```
Integral(f_X, (x, 0, oo)).doit()
#Integral(f_X, (x, 0, oo))
```

[97]:
$$1.0$$

Run the cell below to plot the density.

[98]:
```
Plot_continuous([0, 10], f_X)
```



4.1.4 3c) A Transformation and its Inverse

In the rest of this part of the lab, you will find the density of a function of X. Review Part 2d very carefully first.

Define a new random variable V by applying the following function g to the random variable X:

$$g(x) = \frac{\frac{n}{d}x}{1+\frac{n}{d}x}$$

Then

$$V = g(X) = \frac{\frac{n}{d}X}{1+\frac{n}{d}X}$$

With the help of SymPy, you are going to find the density of V.

As always, start with the possible values. What are the possible values of V?

The possible value of V are between (0,1)

For $n = 6$ and $d = 4$, construct an expression g that is equal to $g(x)$ as defined above.

[99]:
```
g = ((6/4) * x) / (1 + (6/4)*x)
g
```

[99]:

$$\frac{1.5x}{1.5x+1}$$

Is the function g increasing or decreasing? Explain your answer.

The function g is increasing, because as x's value increase, the value of g also increase.

To find the density of $V = g(X)$, you will need the inverse of g.

The function g has a unique inverse. Let $v = g(x)$. Find the function g^{-1} by completing the cell below.

[100]:
```
v = Symbol('v', positive=True)
g_inverse = solve(g-v,x)[0]
g_inverse
```

[100]:
$$-\frac{2.0v}{3.0v-3.0}$$

Remember that for each v, $g^{-1}(v)$ is a value of x. Since x is positive, the inverse you just found better not be negative. Show by algebra (without using SymPy) that the inverse calculated above is positive for each possible value v of V.

We know that the range of V is within (0,1). Since g_inverse is the inverse function of V, (0,1) is the domain of this funciton. We devide both the numerator and the denumerator by v, the the equation evaulats to $\frac{-2.0}{3-\frac{3}{v}}$ since v is between 0 and 1, the demoninator is always going to be smaller than 1. And thus, g_inverse is positve for each possible value of V.

4.1.5 3d) A Derivative

Complete the cell below so that deriv_g equals $\frac{d}{dx}g(x)$.

[101]:
```
deriv_g = diff(g)
deriv_g
```

[101]:
$$-\frac{2.25x}{(1.5x+1)^2}+\frac{1.5}{1.5x+1}$$

Explain why this derivative is positive for all positive x.

Since x is always going to be positive, the right part of the equation is always going to be positive and the left side is going to be negative. However, because the denumenator is squared, the value of the negative side(left) is always going to be smaller than the positive one(right). So the derivative is always going to be positive.

4.1.6 3e) The Density of the Transformation

Use the change of variable formula in **2d** to find f_V, the density of V. Start by filling in the possible values of V in the comment line.

[102]:
```
# Density of V:

"""For v in the interval (0,positive infinity), the density of V at the point v␣
↪is:"""

f_V = (f_X / deriv_g).subs(x,g_inverse)
f_V
```

[102]:
$$\frac{162.0v^2}{(3.0v-3.0)^2\left(-\frac{3.0v}{3.0v-3.0}+1\right)^5\left(\frac{4.5v}{(3.0v-3.0)\left(-\frac{3.0v}{3.0v-3.0}+1\right)^2}+\frac{1.5}{-\frac{3.0v}{3.0v-3.0}+1}\right)}$$

Yikes! That looks like a horrible mess.

Does it simplify at all? Run the next cell and see.

[103]:
```
f_V = simplify(f_V)
f_V
```

[103]:
$$12.0v^2\left(1-v\right)$$

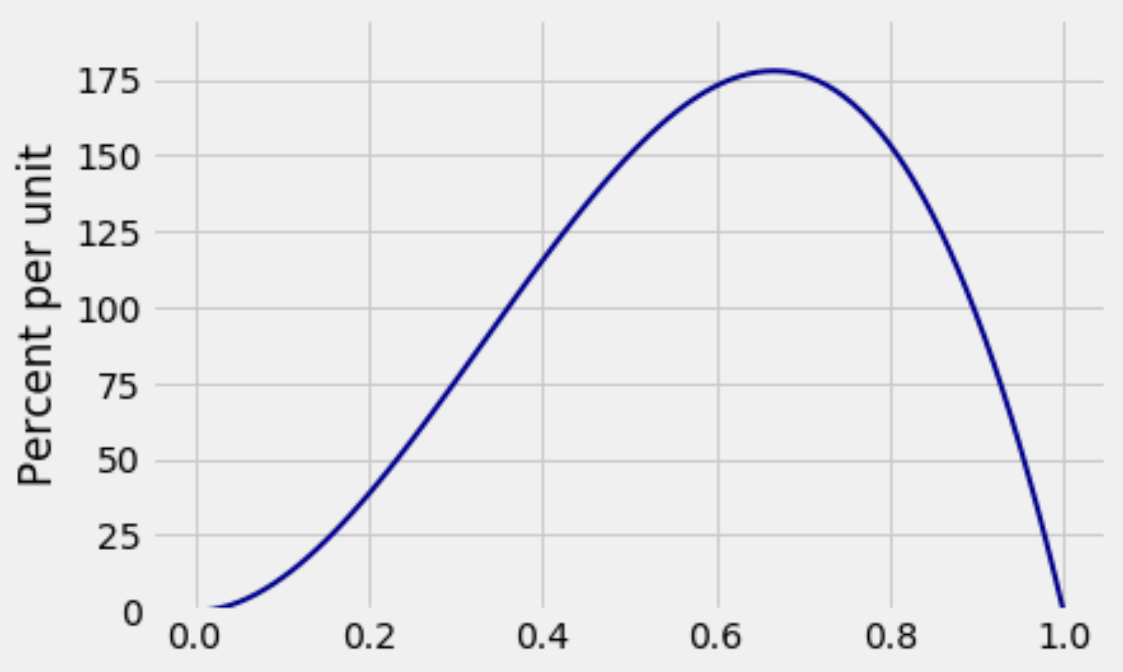
Algebra is truly wonderful. Especially when SymPy does it for you.

Recognize the density of V as one of the famous ones and state its name and parameters. If you are stuck, you may find Chapter 17.4 helpful.

s-1=1 so s=2 r-1=2 so r=3 Thus, it's the beta(r,s) distribution with the parameter r=3 and s=1

Plot the density of V over all the possible values of V.

[104]:
```
Plot_continuous([0, 1], f_V)
```



Note: There is nothing special about the choices $n = 6$ and $d = 4$ as far as this result is concerned. If you had started out with a different (n, d) pair, you would have ended up with a density in the same family but with different parameters. Based on your answer in the case $(6, 4)$, you might be able to guess what the parameters would be in general.

5 newpage

5.1 Part 4: Working with Joint Densities

A joint density is a function of two variables. This part of the lab shows you how to work with such functions in SymPy.

Before you get started, please read Section 17.1 of the textbook. The rest of the lab will go faster that way.

Let X and Y have joint density given by

$$f(x, y) = x + y, \qquad 0 < x, y < 1$$

In order to find probabilities, densities, or expectations of functions of X and Y, you have to create two symbols, which we will assign to the names x and y. Since both have values in $(0, 1)$, we will use the positive=True option.

[105]:
```
x = Symbol('x', positive=True)
y = Symbol('y', positive=True)
```

[106]:
```
f = x+y
f
```

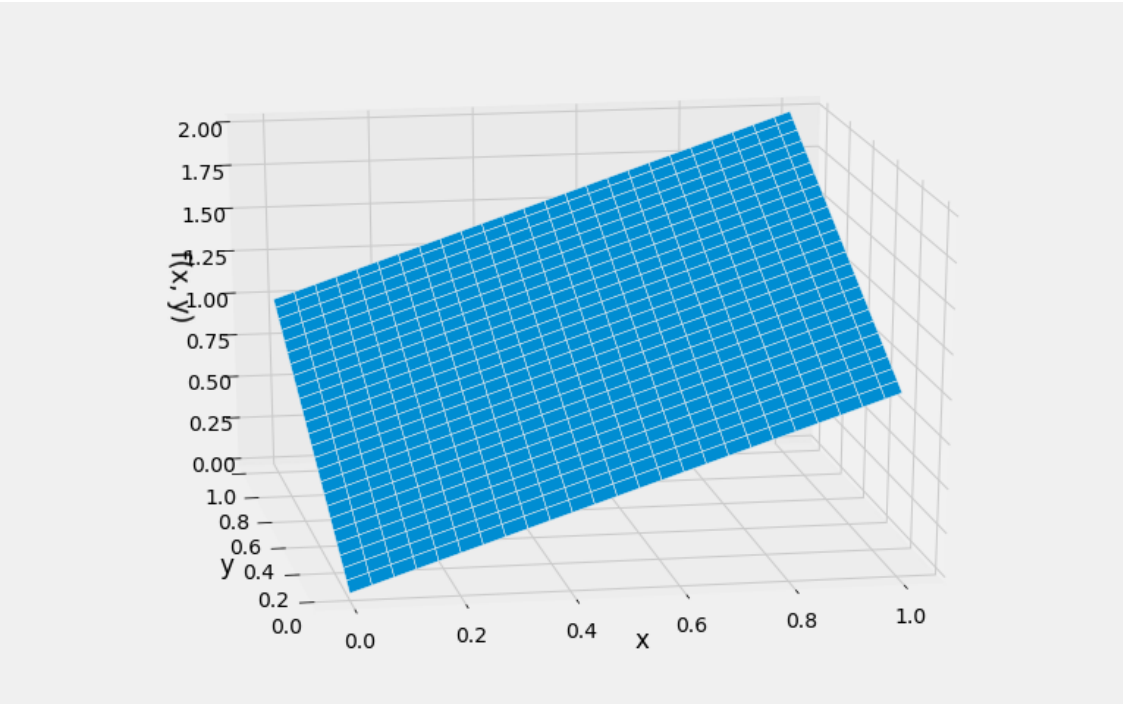
[106]:
$x + y$

You know that the integral of f over the entire unit square has to be 1. To calculate this using SymPy, you have to specify a double integral. The general expression used to display a definite double integral is Integral with three arguments:

- the name of the function to integrate
- (inner_integral_variable, lower_limit, upper_limit)
- (outer_integral_variable, lower_limit, upper_limit)

Read each of the three cells below carefully before you run them. You will need the syntax later.

[107]:
```
def joint(x,y):
    return x+y
Plot_3d(x_limits=(0,1), y_limits=(0,1), f=joint, cstride=4, rstride=4)
```



[108]:
```
# Integral of f over the unit square,
# integrating x first and then y

Integral(f, (x, 0, 1), (y, 0, 1))
```

[108]:

$$\int_0^1 \int_0^1 (x+y)\, dx\, dy$$

[109]:
```
# Integral of f over the unit square,
# integrating y first and then x

Integral(f, (y, 0, 1), (x, 0, 1))
```

[109]:
$$\int_0^1 \int_0^1 (x+y)\, dy\, dx$$

[110]:
```
# Evaluating both integrals

Integral(f, (x, 0, 1), (y, 0, 1)).doit(), Integral(f, (y, 0, 1), (x, 0, 1)).
↪doit()
```

[110]:
$(1,\ 1)$

5.1.1 4a) [ON PAPER]

On a sketch of the unit square, shade the region corresponding to the event $Y > 2X$. Then write $P(Y > 2X)$ as a double integral, in two ways:

- by integrating y first and then x
- by integrating x first and then y

The region isn't a rectangle, so pay attention to the limits of integration.

5.1.2 4b)

In the two cells below, display your answers to Part a. Do not evaluate the integrals.

[111]:
```
# P(Y > 2X) by integrating y first and then x

Integral(f, (y, 2*x, 1), (x,0,0.5))
```

[111]:
$$\int_0^{0.5} \int_{2x}^1 (x+y)\, dy\, dx$$

[112]:
```
# P(Y > 2X) by integrating x first and then y

Integral(f, (x, 0, 0.5), (y, 2*x, 1))
```

[112]:

$$\int_{2x}^{1}\int_{0}^{0.5}(x+y)\,dx\,dy$$

Check that the numerical values of the two double integrals are the same. Notice that SymPy makes choices about how to represent numbers; you might need another cell to do some arithmetic.

[113]:
```
# numerical value of P(Y > 2X)

Integral(f, (y, 2*x, 1), (x, 0, 0.5)).doit(), Integral(f, (x, 0, y/2 ), (y, 0,␣
↪1)).doit()
```

[113]:
$$\left(0.208333333333333,\ \frac{5}{24}\right)$$

[114]:
```
x_first=5/24
y_first=Integral(f, (y, 2*x, 1), (x, 0, 0.5)).doit()
print(y_first,x_first)
print(x_first==y_first)
```

```
0.208333333333333 0.20833333333333334
True
```

5.1.3 4c) Marginal

Find the marginal density of X. See Section 17.3.

6 For x in the interval 0 to 1, the marginal density of X at the point x is:

f_X = Integral(f, (y, 2*x, 1)) f_X

[115]:
```
#Integral(f, (y, 2*x, 1))
f_X = Integral(f, (y, 0, 1)).doit()
f_X
#Integral(f, (x, 0, 1)).doit()
```

[115]:
$$x+\frac{1}{2}$$

Without calculation, say what the marginal density of Y is. Explain your answer.

The marginal density of Y is $f_Y = y + 1/2$, this is becasue X and Y are independent variable with same distribution. Thus the varibles x have the same distributio as variable y, so when we are doing maringal density of Y, we can thus simply subsitute the nema of the variable. And since $f_X = x + \frac{1}{2}$, f_Y will also be $f_Y = y + \frac{1}{2}$

6.0.1 4d) Expectation

What is the relation between $E(X+Y)$ and $E(X)$?

E(X+Y) is a linear transformation of E(X). And since Y is independent of X E(X+Y)=E(X)+E(Y)=2*E(X).

Use f_X (not f) to find the numerical value of $E(X+Y)$.

[116]:
```
Integral(x*f_X,(x,0,1)).doit()*2
#Integral((x+y)*f,(x,0,1),(y,0,1)).doit()
```

[116]: $\frac{7}{6}$

Use the first cell below to display $E(XY)$ as an integral (see Section 17.1), and use the second cell to find its numerical value.

[117]:
```
# E(XY) displayed as an integral
Integral((x*y)*f, (x, 0, 1 ), (y, 0, 1))
```

[117]: $$\int_0^1 \int_0^1 xy\,(x+y)\,dx\,dy$$

[118]:
```
# Numerical value of E(XY)
#Integral((y/x)*120*x*(y-x)*(1-y), (x, 0, y ), (y, 0, 1)).doit()
Integral((x*y)*f, (x, 0, 1 ), (y, 0, 1)).doit()
```

[118]: $\frac{1}{3}$

6.0.2 4e) Conditioning

For $x \in (0,1)$, find the conditional density of Y given $X = x$. This should be an expression involving both x and y, assigned to the name f_Y_given_X_is_x. Fill in the possible values in the comment.

[119]:
```
"""Given X=x, the possible values of Y are between (0,1) For y in this range:"""

f_Y_given_X_is_x = f/f_X
f_Y_given_X_is_x
```

[119]: $\frac{x+y}{x+\frac{1}{2}}$

For each x, this should be a density. Is it? First, display the integral that you have to calculate to check this.

[120]:
```
Integral(f_Y_given_X_is_x,(y,0,1))
```

[120]:

$$\int_0^1 \frac{x+y}{x+\frac{1}{2}}\, dy$$

Now evaluate the integral and check that it has the right value. Use simplify if necessary.

[121]:
```
simplify(Integral(f_Y_given_X_is_x,(y,0,1)).doit())
```

[121]:
$$1$$

Use f_Y_given_X_is_x to find the conditional density of Y given $X = 0.25$ and assign it to the name f_Y_given_X_is_025.

[122]:
```
f_Y_given_X_is_025 = f_Y_given_X_is_x.subs(x,0.25)
f_Y_given_X_is_025
#(y+0.25)/(0.25+0.5)
```

[122]:
$$1.33333333333333y + 0.333333333333333$$

The cell below assigns $P(Y^2 > 0.9 \mid X = 0.25)$ to the name prob and $E(Y \mid X = 0.25)$ to cond_exp. Find the two numerical values by completing the cell.

[123]:
```
prob = Integral(f_Y_given_X_is_025,(y,(0.9)**(1/2),1)).doit()
#prob
cond_exp = Integral(y*f_Y_given_X_is_025,(y,0,1)).doit()
#cond_exp

prob, cond_exp
```

[123]:
$$(0.0837722339831622,\ 0.611111111111111)$$

6.1 Conclusion

Congratulations! The work that you have done in this lab will help reinforce your understanding of some of the most fundamental concepts of probability theory.

What you have learned:

- How to do symbolic math in Python
- How to work with densities and joint densities
- How to find the density of a transformed random variable
- How to use conditioning when random variables are continuous

6.2 Submission Instructions

Many assignments throughout the course will have a written portion and a code portion. Please follow the directions below to properly submit both portions.

6.2.1 Written Portion

- Scan all the pages into a PDF. You can use any scanner or a phone using an application. Please **DO NOT** simply take pictures using your phone.
- Please start a new page for each question. If you have already written multiple questions on the same page, you can crop the image or fold your page over (the old-fashioned way). This helps expedite grading.
- It is your responsibility to check that all the work on all the scanned pages is legible.

6.2.2 Code Portion

- Save your notebook using File > Save and Checkpoint.
- Generate a PDF file using File > Download as > PDF via LaTeX. This might take a few seconds and will automatically download a PDF version of this notebook.
  - If you have issues, please make a follow-up post on the general Lab 8 Piazza thread.

6.2.3 Submitting

- Combine the PDFs from the written and code portions into one PDF. Here is a useful tool for doing so.
- Submit the assignment to Lab 8 on Gradescope.
- **Make sure to assign each page of your pdf to the correct question.**
- **It is your responsibility to verify that all of your work shows up in your final PDF submission.**

6.2.4 We will not grade assignments which do not have pages selected for each question.

[ ]:

lab8.

3.

d.

$$\text{func} = \frac{-2.25x}{(1.5x+1)^2} + \frac{1.5}{1.5x+1}$$

$$= \frac{-2.25}{1.5x+\frac{1}{x}+3} + \frac{2.25}{1.5^2x+1.5}$$

$\Rightarrow$ comparing $1.5x+\frac{1}{x}+3$ & $1.5^2x+1.5$ & $x \in (0,\infty)$.

$$1.5x+\frac{1}{x}+3 > 1.5^2x+1.5$$

$$1.5x+\frac{1}{x}+1.5 > 1.5^2x$$

$$1.5+\frac{1}{x^2}+1.5 > 1.$$

$\Rightarrow$ always true in $(0,\infty)$

$\Rightarrow$ thus, the derivative deriv-g is always positive for all possible x.

4.

a. $\Rightarrow$ unit square $\rightarrow$ event $Y > 2X$

$\Rightarrow$ write $P(Y>2X)$ as double integral $\rightarrow$ y first then x / x first then y.

$\Rightarrow \int_x \int_y f(x,y)\,dy\,dx = 1$

$\Rightarrow f(x,y) = x+y$ & $0 < x,y < 1$

$\Rightarrow$ ① $P(Y>2X) \rightarrow$ y first then x

$$\Rightarrow P(Y>2X) = \int_0^{0.5}\int_{2x}^{1}(x+y)\,dy\,dx$$

$$\Rightarrow P(Y>2X) = \int_{2x}^{1}\int_0^{0.5}(x+y)\,dx\,dy.$$