4 Part B: Transforming Continuous Variables (Due March 31st at 5 PM)

4.1 Identify Your Lab Partner

This is a multiple choice question. Please select **ONE** of following options that best describes how you complete Lab 6B.

- I am doing Part B of this lab by myself and I don't have a partner.
- My partner for Part B of this lab is [PARTNER'S NAME] with email [berkeley.edu email address]. [SUBMITTER'S NAME] will submit to Gradescope and add the other partner to the group on Gradescope after submission.

Please copy and paste **ONE** of above statements and fill in blanks if needed. If you work with a partner, make sure only one of you submit on Gradescope and that the other member of the group is added to the submission on Gradescope. Refer to the bottom of the notebook for submission instructions.

I am doing part B of this lab by myself

4.2 Section 4. Extension to Continuous Distributions

Now suppose you want to simulate a random variable that has a specified continuous distribution. Let's start with the exponential (λ) distribution.

Let T have the exponential distribution with rate λ. Let F_T be the cdf of T. Refer to Section 15.4.1 for the formula for F_T, and don't forget that $F_T(t) = 0$ for $t < 0$.

4.2.1 4a) Plotting the Exponential CDF

For a numerical example, let T be a random variable that has the exponential distribution with rate $\lambda = 0.5$, or equivalently, expectation 2. Define a function `expon_mean2_cdf` that takes a numerical argument x and returns the numerical value of $F_T(x)$. Use `np.exp(y)` for e^y.

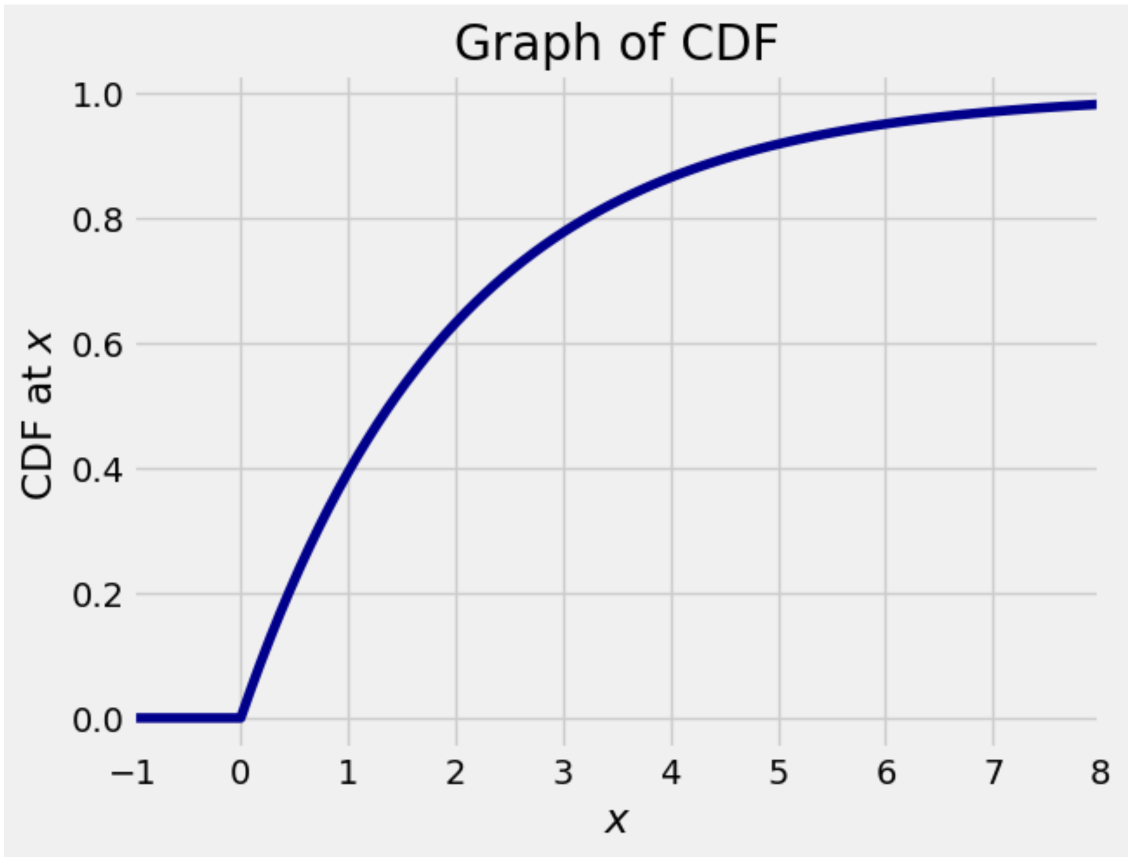
Make sure your function returns the correct value of $F_T(x)$ for **all** real numbers x.

[51]:
```python
# Don't use "lambda" as that means something else in Python
lamb = 0.5

def expon_mean2_cdf(x):
    if x < 0:
        return 0
    else:
        return 1 - np.exp(-lamb * x)
```

The function `plot_continuous_cdf` plots the cdf of a continuous variable. The first two arguments: - an interval (a, b) over which to draw the cdf - the name of a function that takes a numerical input and returns the value of the cdf at that input

Run the cell below to check that your function `expon_mean2_cdf` looks good.

[52]:
```python
plot_continuous_cdf((-1, 8), expon_mean2_cdf)
```



4.2.2 4b) Idea for Simulating an Exponential Random Variable

Suppose you are given one uniform $(0, 1)$ random number and are asked to simulate T. Based on Part 3 of the lab, propose a method for doing this by using the graph above.

You don't have to prove that the method works. There's a formal proof in the textbook. Just propose the method.

A method to simulate T given one uniform $(0, 1)$ random number U would be to find the corresponding T-value where the CDF equals U. We take U as a probability and determine the T-value that has that probability of occuring under the exponential distribution.

4.2.3 4c) Visualizing the Idea

The animation in the cell below is analogous to the one in Part 3. Its arguments are: - a plotting interval - the name of a continuous cdf function

The output demonstrates a method for generating a number on the positive real line by starting with a value on the unit interval that forms the vertical axis.

Run the cell and move the slider around to see how the returned value changes depending on the starting value on the vertical axis.

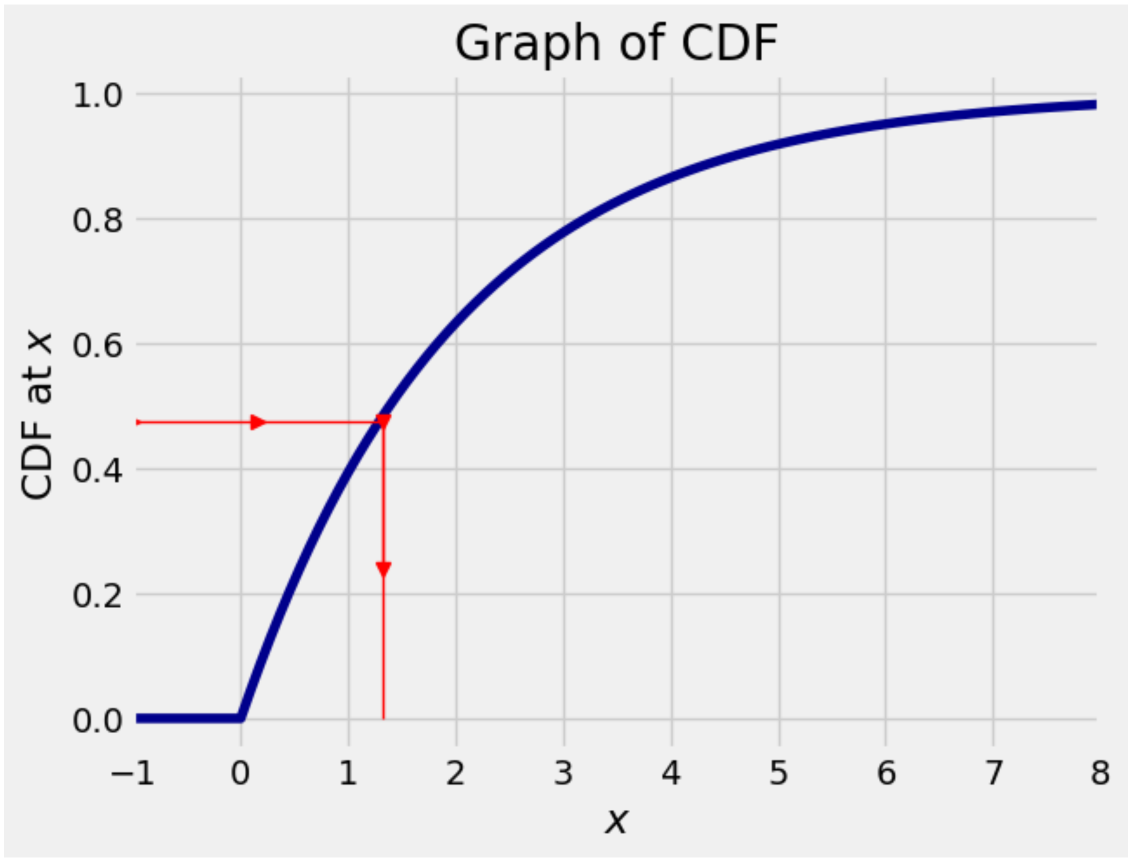
```
[53]: unit_interval_to_continuous((-1, 8), expon_mean2_cdf)
```

```
interactive(children=(FloatSlider(value=0.5, description='u', max=1.0, step=0.
↪02), Output()), _dom_classes=('w…
```

The method `plot_continuous_cdf` takes an optional third argument that is a number between 0 and 1.

Complete the cell below so that the third argument is a random number picked uniformly from (0, 1). Refer to **3f** for relevant code.

```
[54]: plot_continuous_cdf((-1, 8), expon_mean2_cdf, np.random.uniform(0,1))
```



Run the cell a few times. For generating a value of T, how is the output related to the method you proposed in **4b**?

The output is related to the method proposed in 4b because it visualizes the simulation process. The random value U corresponds to a cumulative probability, which is then matched to the CDF to find the corresponding value of T. This shows how the CDF function maps the uniform random variable U to a value from the exponential distribution.

4.2.4 4d) [ON PAPER] The General Method

Let F be any continuous increasing cdf. That is, suppose F has no jumps and no flat bits.

Suppose you are trying to create a random variable X that has cdf F, and suppose that all you have is F and a number picked uniformly on $(0, 1)$.

(i) **Fill in the blank:** Let U be a uniform $(0, 1)$ random variable. To construct a random variable $X = g(U)$ so that X has the cdf F, take $g =$ _________.

(ii) **Fill in the blank:** Let U be a uniform $(0, 1)$ random variable. For the function g defined by

$$g(u) \;=\; \underline{\hspace{6cm}}, \qquad 0 < u < 1$$

the random variable $X = g(U)$ has the exponential (λ) distribution.

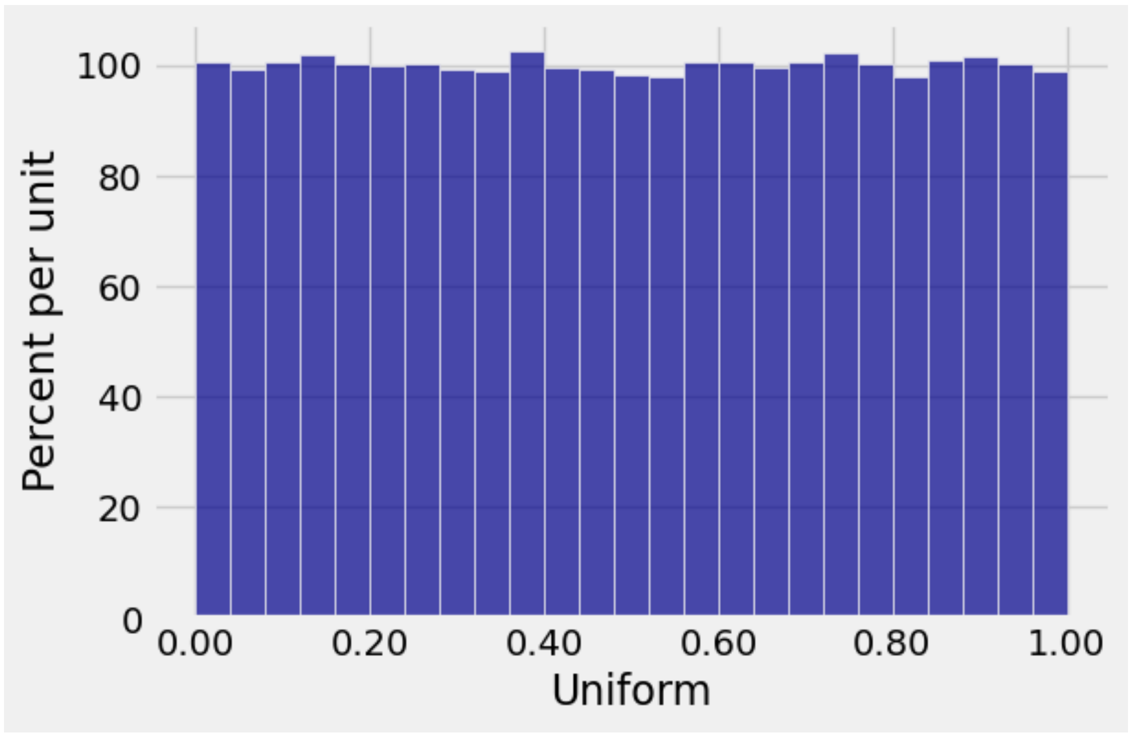
[Note: If F is a discrete cdf then the function g is complicated to write out formally, so we're not asking you to do that. The practical description of the method of simulation is in Part 3 of the lab.]

4.2.5 4e) Empirical Verification

Let's visualize what your method is doing and confirm that it works.

In the cell below, we have used `SciPy` to create a table that is called `sim` for "simulation". It consists of one column called `Uniform` that contains the values of 100,000 i.i.d. uniform $(0, 1)$ random variables. In the last line, the argument `bins=25` is used to create a histogram with 25 bins of equal width. Run the cell.

```
[55]: N = 100000
      u = stats.uniform.rvs(0, 1, size=N)
      sim = Table().with_columns('Uniform', u)
      sim.hist('Uniform', bins=25)
```



Use **4d** and the values in the column `Uniform` to create an array of values that have the exponential distribution with rate 0.5. Use `np.log(y)` for $\log(y)$.

Then augment `sim` with a column containing the new array.

Do not simulate new random numbers, as you will lose the connection with the values in `Uniform`.

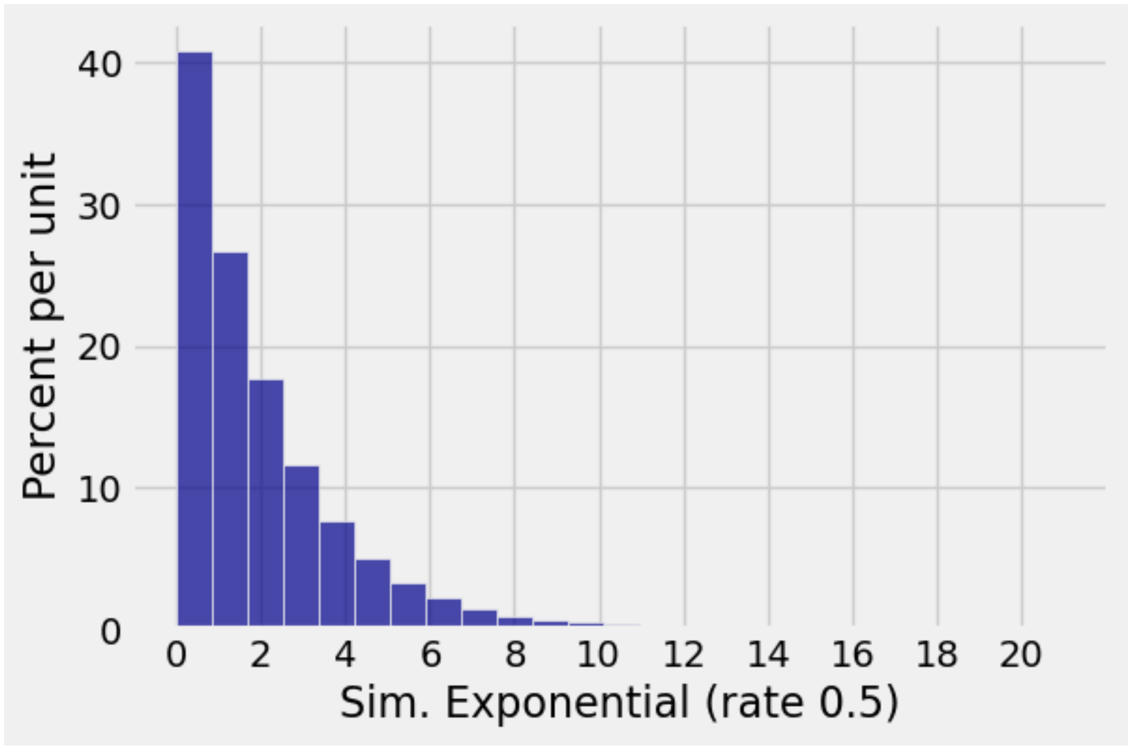
```
[56]: def uniform_to_exponential_mean2(u):
          return -np.log(u)/0.5

      exponential_mean2 = sim.apply(uniform_to_exponential_mean2, 'Uniform')
      sim = sim.with_columns('Sim. Exponential (rate 0.5)', exponential_mean2)
      sim
```

```
[56]: Uniform  | Sim. Exponential (rate 0.5)
      0.30364  | 2.38382
      0.612643 | 0.979945
      0.28921  | 2.48121
      0.816231 | 0.406117
      0.87234  | 0.273152
      0.362081 | 2.03178
      0.642735 | 0.884046
      0.196652 | 3.25264
      0.947984 | 0.106835
      0.90544  | 0.198669
```

```
… (99990 rows omitted)
```

Run the cell below to confirm that your calculation is correct.

[57]:
```
sim.hist('Sim. Exponential (rate 0.5)', bins=25)
plt.xticks(np.arange(0, 21, 2));
```



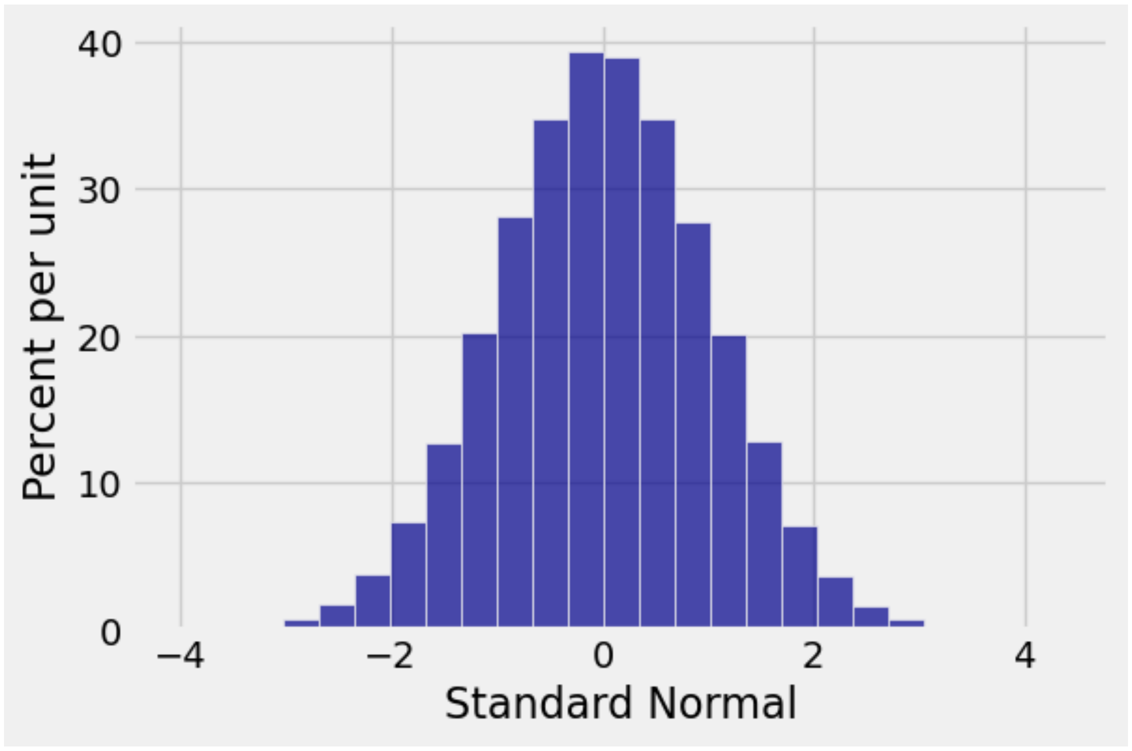
You have just discovered what is going on "under the hood" in functions that generate exponential random numbers. There are more complicated methods that work faster in particular cases. But this one is pretty good!

4.2.6 4f) Simulating the Standard Normal

Now use the general method in **4d** and the `Uniform` column of the table `sim` to generate 100,000 i.i.d. standard normal variables. Augment `sim` with a column labeled `Standard Normal` containing standard normal variables created by transforming the values in `Uniform`.

Unlike the exponential, the standard normal cdf Φ must be computed numerically, and hence so must its inverse Φ^{-1}. You have to identify the appropriate `SciPy` function to transform the `Uniform` column. Do not add more lines of code.

[58]:
```
standard_normals = stats.norm.ppf(sim.column('Uniform'))
sim = sim.with_columns('Standard Normal',
                       standard_normals)
sim.hist('Standard Normal', bins=25)
```



At this point, go back and look through Part 4. Notice that the only time you generated random numbers was when you simulated 100,000 uniform (0, 1) values. All the other variables were deterministic transformations of the uniform random numbers.

4.2.7 Note on Uniform Random Numbers

Since uniform $(0, 1)$ random numbers are central to all simulations, their quality is very important for the accuracy and reliability of simulations. Testing and assessing uniform random number generators is serious business, because random number generators don't really produce random numbers. They follow deterministic processes that produce results that have properties that resemble those of random numbers. That is why they are called Pseudo Random Number Generators or PRNGs. Python uses the Mersenne Twister, one of the most tested and reliable PRNGs.

4.3 Section 5: Densities of Transformed Variables

You now know how to transform a uniform $(0,1)$ random variable into a random variable with any specified distribution. But often, we start with some random variable X, not necessarily uniform, and our analysis requires us to transform the variable.

To find the density of the transformed variable, we can sometimes use the methods of Chapter 16. In this part of the lab you will examine how to do this.

Before you begin, please review Parts 1 and 2 to recall how to use `SymPy` to work with densities.

As in Part 2, let X have density $f_X(x) = 6x^5$ for $0 < x < 1$. Run the cell below to define the density function `f_X`.

[88]:
```
x = Symbol('x', positive=True)
f_X = 6 * x**5
f_X
```

[88]: $6x^5$

4.3.1 5a) 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

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

See Section 16.2.1 for the derivation, and notice how the derivation depends on the cdf of X. The cdf was also the principal function you used to create the transformations in Parts 3 and 4 of the lab.

Each time you use the change of variable 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 already declared `x` and `f_X(x)` at the start of this part of the lab.

[89]:
```
g = pi * x**2
g
```

[89]: 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`

[90]:
```
pi, np.pi
```

[90]: $(\pi, \ 3.14159265358979)$

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 are $V \geq 0$, as X represents the radius of a disc, which cannot have a negative value.

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.

[91]:
```
v = Symbol('v', positive=True)
g_inverse = solve(g - v, x)
g_inverse
```

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

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.

[92]:
```
g_inverse = g_inverse[0]
g_inverse
```

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

The change of variable formula has another factor: the derivative in the denominator. Since we are working only on the interval $(0, 1)$, g is an increasing function. So 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`.

[93]:
```
deriv_g = diff(g, x)
deriv_g
```

[93]: $2\pi x$

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.**

[94]:
```
"""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
```

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

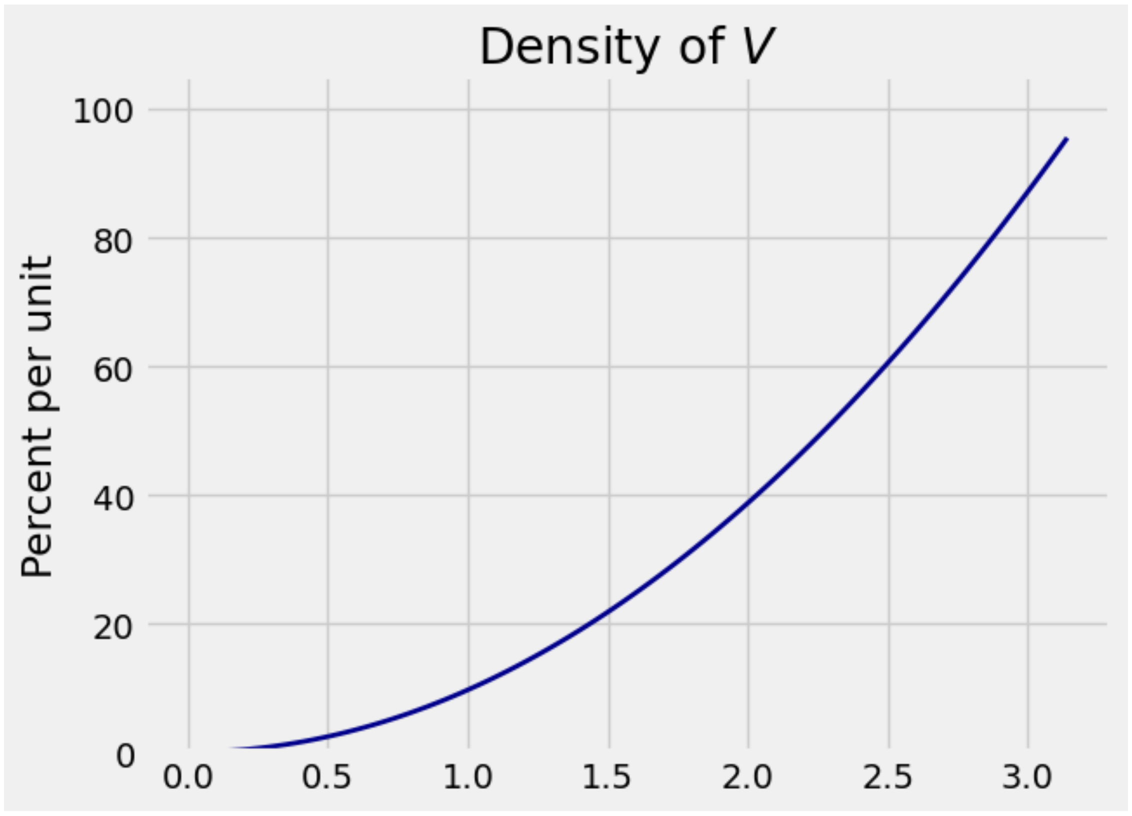
Check that the function `f_V` is a density.

[95]:
```
integration = integrate(f_V, (v,0,pi))
print(integration)
```

```
1
```

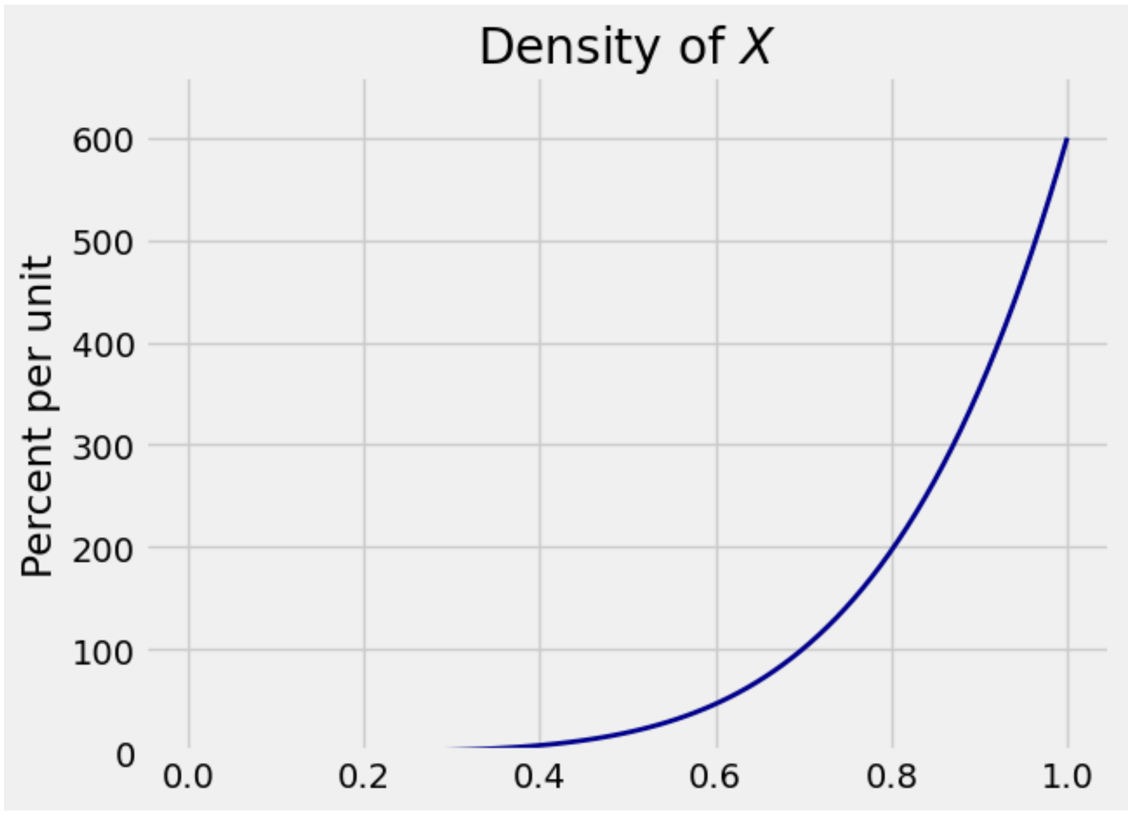
Plot the graph of the density of V. Recall from Part **2a** that the first argument of `Plot_continuous` is a list of two elements: the left and right ends of the interval over which to plot the graph. They must be numerical expressions, not symbolic.

[98]:
```
Plot_continuous([0,np.pi], f_V)
plt.title('Density of $V$');
```



For comparison, run the cell below to see the graph of f_X again.

```
[97]: Plot_continuous((0, 1), f_X)
      plt.title('Density of $X$');
```



4.4 Section 6: Transforming an F Distribution

The density you started with in Section 5 was deliberately chosen to be simple, so that you could try out the calculations by hand if you wanted to check. 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 the cdf. It is named for Sir Ronald Fisher whose contributions include what is now the standard method of testing statistical hypotheses.

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. You will become closely acquainted with gamma distributions later in the course.

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.

4.4.1 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

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

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

4.4.2 6a) 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 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.

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

```
[68]: 6
```

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

[69]:
```
11.6317283965674
```

[70]:
```
gamma(5)
```

[70]:
```
24
```

[71]:
```
r = Symbol('r', positive=True)
gamma(r)
```

[71]: $\Gamma(r)$

Start 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.

[72]:
```
def constant_F(n, d):
    return (gamma(n/2 + d/2) / (gamma(n/2) * gamma(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.

[73]:
```
constant_F(4, 4)
```

[73]:
```
6.0
```

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

[74]:
```
constant_F(2, 4)
```

[74]:
```
2.0
```

4.4.3 6b) 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 $f_X(x)$.

[75]:
```
x = Symbol('x', positive=True)
f_X = constant_F(6,4) * (6/4)**(6/2)  * x**((6/2) - 1) * (1 + (6/4)*x) **␣
↪(-(6+4)/2)
f_X = nsimplify(f_X) # simplifies expression above to avoid a bug from previous␣
↪semesters
f_X
```

[75]:
$$\frac{16x^2}{3\left(x+\frac{2}{3}\right)^5}$$

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 ∞.

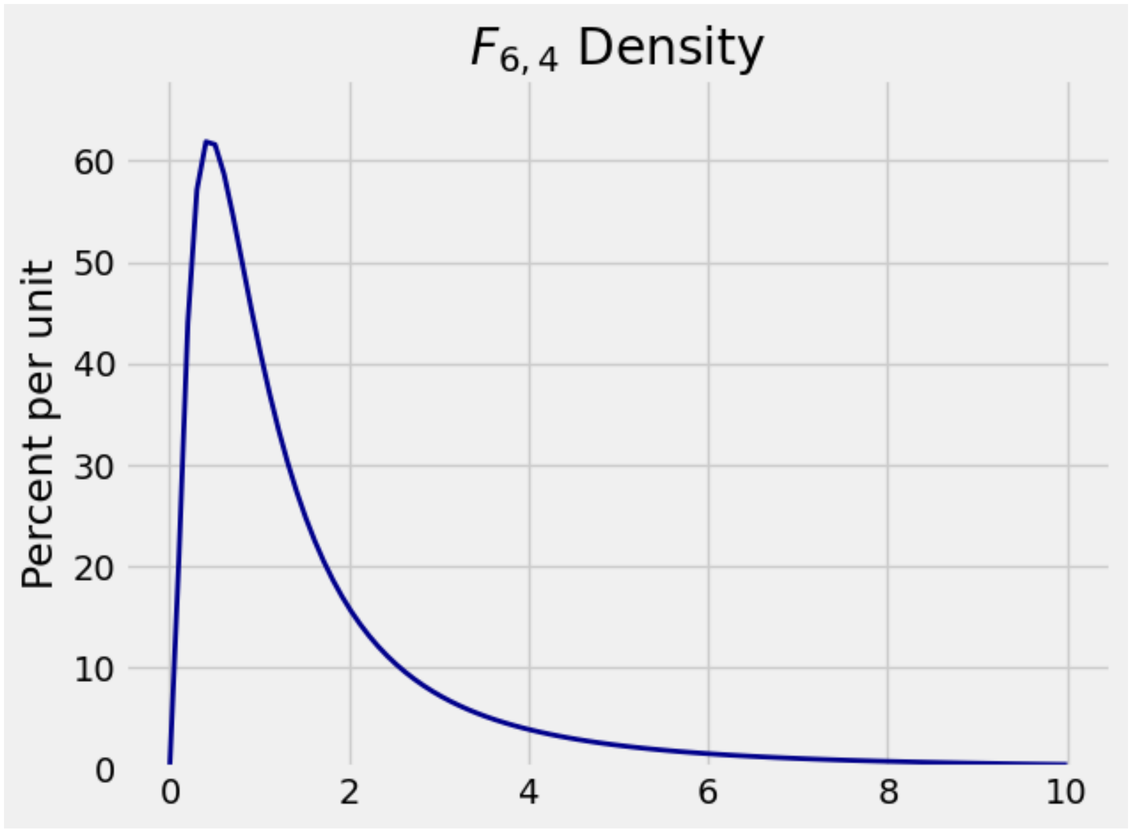
[76]:
```
Integral(f_X, (x, 0, oo)).doit()
```

[76]:

1

Run the cell below to plot the density.

[77]:
```
Plot_continuous([0, 10], f_X)
plt.title('$F_{6,4}$ Density');
```



4.4.4 6c) A Transformation and its Inverse

In the rest of the lab, you will find the density of a transformation of X. **Review Part 5** 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 values 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.

[78]:
```
g = ((6*x)/4) / (1 + ((6*x)/4))
g
```

[78]:
$$\frac{3x}{2\left(\frac{3x}{2}+1\right)}$$

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

[It helps to write the function as 1 minus something.]

The function g is monotonically increasing because its value increases as x increases.

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

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

[79]:
```
v = Symbol('v', positive=True)
g_inverse = solve(g-v,x)[0]
g_inverse
```

[79]:
$$-\frac{2v}{3v-3}$$

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.

work on paper

4.4.5 6d) A Derivative

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

[80]:
```
deriv_g = diff(g)
deriv_g
```

[80]:
$$-\frac{9x}{4\left(\frac{3x}{2}+1\right)^2}+\frac{3}{2\left(\frac{3x}{2}+1\right)}$$

Explain why this derivative is positive for all positive x.

The term on the left has a squared denominator, causing it to decrease more quickly as x increases, while the term on the right remains positive. As a result, for larger values of x, the term on the right will always be larger than the term on the left, ensuring that the derivative remains positive for all positive values of x.

4.4.6 6e) The Density of the Transformation

Use the change of variable formula in Part 5 to find f_V, the density of $V$. Start by filling in the possible values of $V$ in the comment line.

[81]:
```
# 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
```

[81]:

$$\frac{64v^2}{3\left(3v-3\right)^2\left(-\frac{2v}{3v-3}+\frac{2}{3}\right)^5\left(\frac{9v}{2(3v-3)\left(-\frac{3v}{3v-3}+1\right)^2}+\frac{3}{2\left(-\frac{3v}{3v-3}+1\right)}\right)}$$

Yikes! That looks like a horrible mess.

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

[82]:
```
f_V = simplify(f_V)
f_V
```

[82]:

$$12v^2\left(1-v\right)$$

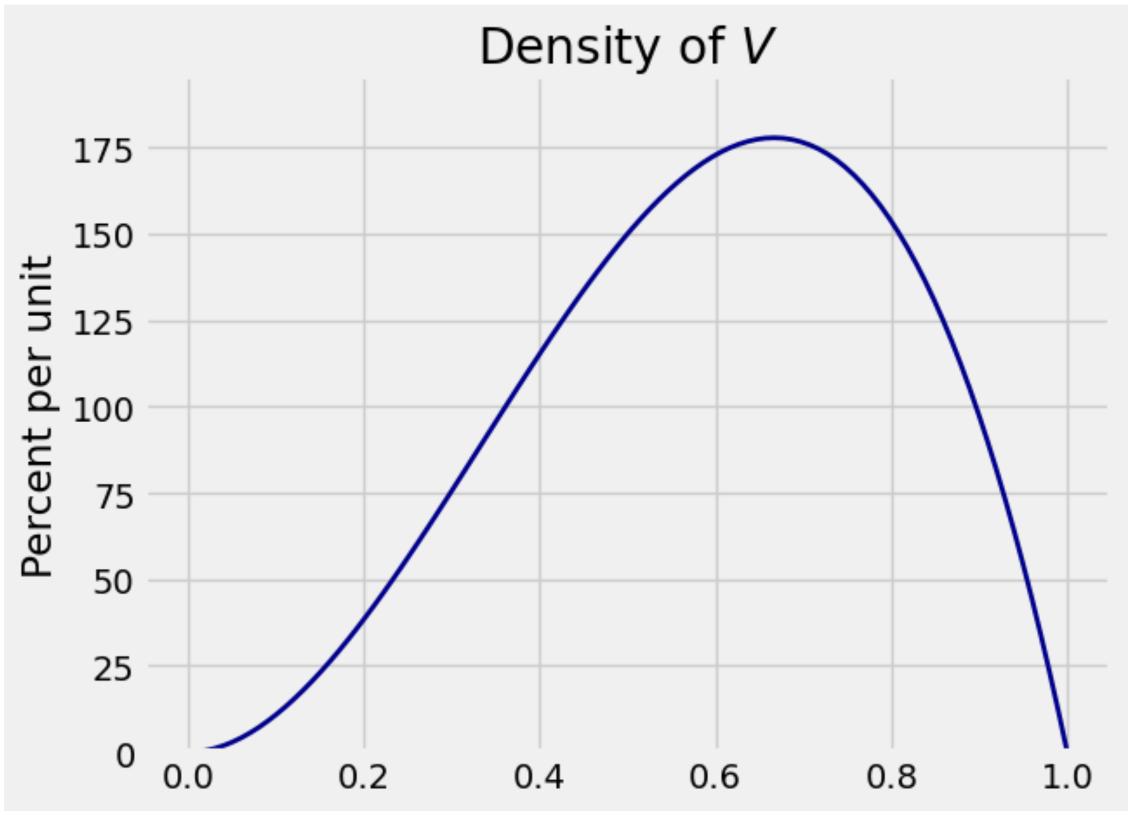
Algebra is truly wonderful. Especially when `SymPy` does it for you.

Recognize the density of $V$ as one of the famous ones and state its name and parameters. If you are stuck, you may find this section of the textbook helpful.

$V$ ~ beta(3,2) density function

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

[83]:
```
Plot_continuous([0,1], f_V)
plt.title('Density of $V$');
```



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 Part B of the lab ends here. It is due on Monday, March 31st at 5 PM.

5.1 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
- How to use uniform random numbers to generate random variables with any specified distribution
- How to find the density of a transformed random variable

5.2 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.

5.2.1 Written Portion

- Scan all the pages into a PDF. You can use any scanner or a phone using applications such as CamScanner. 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 in CamScanner 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.
- If you used LaTeX to do the written portions, you do not need to do any scanning; you can just download the whole notebook as a PDF via LaTeX.

5.2.2 Code Portion

- Save your notebook using `File > Save Notebook`.
- Generate a PDF file using `File > Save and Export Notebook As > PDF`. This might take a few seconds and will automatically download a PDF version of this notebook.
  - If you have issues, please post a follow-up on the general Lab 6B Ed thread.

5.2.3 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 6B 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.**

If you are having difficulties scanning, uploading, or submitting your work, please read the Ed Thread on this topic and post a follow-up on the general Lab 6B Ed thread.

5.3 We will not grade assignments that do not have pages selected for each question.

```
[ ]:
```

4di. $F \sim$ continuous, increasing cdf

$$g = F^{-1}$$

$$\begin{aligned} \therefore \text{cdf} &= P(X \leq x) \\ &= P(g(U) \leq x) \\ &= P(F^{-1}(U) \leq x) \\ &= P(U \leq F(x)) \\ &= F(x) \end{aligned}$$

4dii. $g(U) = F^{-1}(U)$

$$\begin{aligned} F_x(x) &= 1 - e^{-\lambda x} \qquad x \geq 0 \\ U &= 1 - e^{-\lambda x} \\ U - 1 &= -e^{-\lambda x} \\ 1 - U &= e^{-\lambda x} \\ \ln(1-U) &= \ln e^{-\lambda x} \\ \lambda x &= -\ln(1-U) \\ x &= -\frac{\ln(1-U)}{\lambda} \\ g(U) &= -\frac{\ln(U)}{\lambda} \qquad 0 < U < 1 \end{aligned}$$

$1 - U$ and U are both uniform on $(0,1)$

$\therefore \ln(1-U) \longrightarrow \ln(U)$

6a. $n = d = 4$

$$\frac{\Gamma\left(\frac{4}{2}+\frac{4}{2}\right)}{\Gamma\left(\frac{4}{2}\right)\Gamma\left(\frac{4}{2}\right)}\left(\frac{4}{4}\right)^{\frac{4}{2}} = \frac{\Gamma(2+2)}{\Gamma(2)\Gamma(2)}(1)^2$$

$$= \frac{3!}{1!\,1!}$$

$$= 3 \times 2$$

$$= 6 \quad ✓$$

$n = 2,\ d = 4$

$$\frac{\Gamma\left(\frac{2}{2}+\frac{4}{2}\right)}{\Gamma\left(\frac{2}{2}\right)\Gamma\left(\frac{4}{2}\right)}\left(\frac{2}{4}\right)^{\frac{2}{2}} = \frac{\Gamma(1+2)}{\Gamma(1)\Gamma(2)}\left(\frac{1}{2}\right)^1$$

$$= \frac{2!}{0!\,1!}\left(\frac{1}{2}\right)$$

$$= 2\left(\frac{1}{2}\right)$$

$$= 1 \quad ✓$$

6c. $g(x) = \dfrac{\frac{6}{4}x}{1+\frac{6}{4}x}$

$$v = \frac{\frac{6}{4}x}{1+\frac{6}{4}x}$$

$$v\left(1+\frac{6}{4}x\right) = \frac{6}{4}x$$

$$v + \frac{6}{4}xv = \frac{6}{4}x$$

$$v = \frac{6}{4}x - \frac{6}{4}xv$$

$$v = \frac{3}{2}x(1-v)$$

$$x = \frac{2v}{3(1-v)}$$

$$g^{-1}(v) = \frac{2v}{3-3v}$$