Fall25-1110-Lab-7

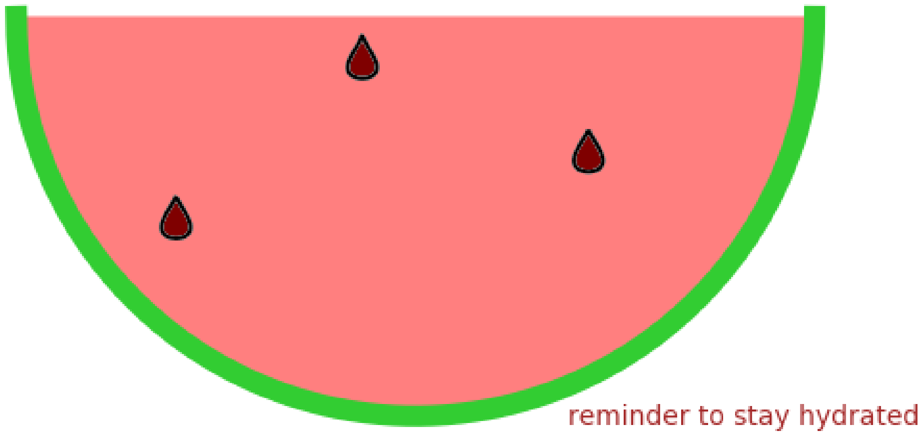
December 30, 2025

[53]:
```python
# remember to include these two lines of code at the start of your document!
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

1 MATH1110 Lab 7: Summations

[16]:
```python
x = var('x')
g1 = plot( -sqrt(3^2-x^2), -3, 3, color = 'limegreen', thickness = 8, fill =␣
↪True, fillcolor = 'red')
a=140
z = var("z")
s1 = implicit_plot( a*(x-1.3)^2-(1-(z+1)*6)^3*(1+(z+1)*6)==0, (x, 0, 2.6), (z,␣
↪-2, 0), color = 'black', fill = True, fillcolor = 'black')
s2 = implicit_plot( a*(x+1.8)^2-(1-(z+1.5)*6)^3*(1+(z+1.5)*6)==0, (x,-2.4, 0.
↪2), (z, -2, 0), color = 'black', fill = True, fillcolor = 'black')
s3 = implicit_plot( a*(x+0.4)^2-(1-(z+0.3)*6)^3*(1+(z+0.3)*6)==0, (x, -2, 0),␣
↪(z, -2, 0), color = 'black', fill = True, fillcolor = 'black')
seeds = s1+s2+s3
tt = text('reminder to stay hydrated!', (2.5, -3), color = 'brown')

show((g1+seeds+tt), ymin = -4, ymax = 1, axes = False)
```



Objectives for today:

1. How does Σ work?
2. Computing summations
3. Riemann sums

1.1 1. Translating summations into Sage

This is how we translate summation notation to syntax:

$$\sum_{i=1}^{n} f(i)$$

In Sage, the $f(i)$ is what goes in the first argument. That is the function being summed. The second argument is the *index*, which is given as the variable beneath the capital sigma letter. The third and fourth entries are where the index will start, and where it will stop. Note that in the general notation above, I started i at 1 (often it's 1 or 0) and used a variable n for the last value.

[19]:
```
# this is the generic form for some function f:
n,i = var('n, i')
f = function('f')(i)
show( sum(f, i, 1, n))
# same thing as above!
```

```
sum(f(i), i, 1, n)
```

How could the calculation $3 + 3 + 3 + 3$ be rewritten as a sum?

[ ]:

1.2 2. Computing summations

The number at the top of our Σ symbol could be any number of things; a whole number, a variable, or even infinity.

Try finding $\sum_{k=1}^{50} k$ and then $\sum_{k=1}^{\infty} k$. What response do you get from Sage?

[ ]:

If the ending point of our function is a *variable*, then the output will likely be some function of that variable.

Simplify the following:

$$\sum_{k=1}^{n} \frac{4}{k^2 + 2k}$$

.

[ ]:

What about the limit of this summation as $n \to \infty$? Remember that Sage accepts the value the variable approaches as '$n = \_$' in the second argument of the *limit* command.

[ ]:

1.2.1 Exercise questions

Compute the following summations. Reminder that *N(_)* gives a decimal approximation if your answer is hard to read.

1. $$\sum_{x=0}^{42} \sin\left(\frac{\pi}{2}x\right)$$

2. $$\sum_{i=2}^{70} \frac{1}{\ln(i)}$$

Be warned that if you try to sum to a sufficiently large number, Sage will start taking a long time. This is signified by the * symbol in the brackets to the left. If it doesn't go away after a minute, you'll probably want to stop/refresh your kernel and lower the number.

[ ]:

1.3 3. Riemann Sums

One method of finding the area beneath a curve without any calculus is to approximate the area by slicing it into rectangles. This is the idea behind Reimann Sums. We'll proceed with a variation of sorts called the Right-Hand Rule:

$$\int_a^b f(x)dx = \lim_{n\to\infty}\sum_{i=1}^{n}\Delta x\Delta y = \lim_{n\to\infty}\sum_{i=1}^{n}(\frac{b-a}{n})f(a+i(\frac{b-a}{n}))$$

Here $\Delta x = \frac{b-a}{n}$ is the change in x (or the width of each rectangle) and Δy is the height of the rectangle at an indicated spot. n is the number of rectangles we're dividing the area under the curve into. > Question: Suppose we started with an estimation of area beneath a curve by dividing it into $n = 10$ sections. Will our estimation of area get better or worse as n increases towards infinity?

Let's practice with the sum of $y = 2x$ from 0 to 4. Start by plotting the curve and shade the region between it and x-axis using the argument '*fill = True*'.

$$\lim_{n\to\infty}\sum_{i=1}^{n}(\frac{4-0}{n})f(0+i(\frac{4-0}{n}))$$

[2]:

```
clear_vars()
x=var('x')
func = 2*x
# first make the plot



i = var('i')
a=0
b=4
n=8
show( (b-a)/n*func(0+i*(b-a)/n) )
# the sum ...
```

1/2*i

Given that the region is a triangle, what is the true area? Now take the accurate Riemann Sum as $n \to \infty$.

[ ]:

```

```

Plot the function $\arctan(x) \cdot e^{-x}$ for $x \geq 0$. Do you think the area is finite or infinite and why?

Then approximate the area beneath the curve using the right hand rule Riemann sum.

[ ]:

```

```

1.3.1 Exercise questions

For both questions, plot the function and shade the region between the curve and x-axis. The compute the appropriate summation.

3. Plot $g(x) = (x - 3)(x^2 - 1)$ and use the Right-Hand Rule to approximate the enclosed area beneath the curve (and above the x-axis) with $n = 13$ rectangles.

4. Find the exact area within $f(x) = \frac{1}{5}e^x$, $x = -2$, $x = 2$ and $y = 0$ using the Riemann Sum approach.

[ ]:

```
```

[ ]:

```
```

[ ]:

```
```