Math 151- Python Lab 9

November 17, 2022

0.1 MATH 151 Lab 8

Section Number: 568

Members:

- Brighton Sikarskie
- Colton Hesser
- Gabriel Gonzalez
- Gabriel Cuevas

[1]:
```python
from sympy import *
from sympy.plotting import (plot,plot_parametric)
```

0.1.1 Question 1

1a

[2]:
```python
# given: y = (1 + (26 / x))^x
# rewrite in form y = f(x)/g(x)
# f(x) = ln(1 + (26 / x))
# g(x) = (1 / x)
x = Symbol('x')
f = ln(1 + (26 / x))
g = (1 / x)
y = f / g

print(f"y = {(1 + (26/x))**x} can be rewritten as y = {exp(y)}")
print(f"We can check for the limit of the exponent as x approaches infinity as␣
↪f(x) / g(x) where f(x) = ln(1 + (26 / x)) and g(x) = (1 / x)")
```

```
y = (1 + 26/x)**x can be rewritten as y = exp(x*log(1 + 26/x))
We can check for the limit of the exponent as x approaches infinity as f(x) /
g(x) where f(x) = ln(1 + (26 / x)) and g(x) = (1 / x)
```

1b

[3]:
```python
# find the limits of f and g as x -> oo
f_lim = limit(f, x, oo)
g_lim = limit(g, x, oo)
print(f"Limit of {f} as x -> oo: {f_lim}")
```

```
print(f"Limit of {g} as x -> oo: {g_lim}")
```

```
Limit of log(1 + 26/x) as x -> oo: 0
Limit of 1/x as x -> oo: 0
```

1c

[4]:
```
# use L'Hopital's rule to find the limit of y as x -> oo
print(f"Since the limit of {f} as x -> oo is {f_lim} and the limit of {g} as x␣
↪-> oo is {g_lim}, we can use L'Hopital's rule to find the limit of y as x ->␣
↪oo.")

f_diff = diff(f, x)
g_diff = diff(g, x)
y_before = y
y = simplify(f_diff / g_diff)

print(f"For y = f(x) / g(x) where f(x) = {f} and g(x) = {g}, the derivative of␣
↪f(x) is {f_diff} and the derivative of g(x) is {g_diff}.")
print(f"We can rewrite y = {y_before} as y = {f_diff} / {g_diff} or simplified␣
↪y = {y} in order to take the limit as stated by L'Hopital's rule.")

numerator = numer(y)
denominator = denom(y)
print(f"The numerator of y is {numerator} and the denominator of y is␣
↪{denominator}.")

numerator_lim = limit(numerator, x, oo)
denominator_lim = limit(denominator, x, oo)

print(f"The limit of the numerator as x -> oo is {numerator_lim} and the limit␣
↪of the denominator as x -> oo is {denominator_lim}.")

numerator_diff = diff(numerator, x)
denominator_diff = diff(denominator, x)
print(f"The derivative of the numerator is {numerator_diff} and the derivative␣
↪of the denominator is {denominator_diff}.")

numerator_diff_lim = limit(numerator_diff, x, oo)
denominator_diff_lim = limit(denominator_diff, x, oo)

print(f"The limit of the derivative of the numerator as x -> oo is␣
↪{numerator_diff_lim} and the limit of the derivative of the denominator as x␣
↪-> oo is {denominator_diff_lim}.")

exponent = numerator_diff_lim / denominator_diff_lim
print(f"The exponent of the limit of y as x -> oo is {exponent}.")
```

```
print(f"The limit of y as x -> oo is {exp(exponent)}.")
```

```
Since the limit of log(1 + 26/x) as x -> oo is 0 and the limit of 1/x as x -> oo
is 0, we can use L'Hopital's rule to find the limit of y as x -> oo.
For y = f(x) / g(x) where f(x) = log(1 + 26/x) and g(x) = 1/x, the derivative of
f(x) is -26/(x**2*(1 + 26/x)) and the derivative of g(x) is -1/x**2.
We can rewrite y = x*log(1 + 26/x) as y = -26/(x**2*(1 + 26/x)) / -1/x**2 or
simplified y = 26*x/(x + 26) in order to take the limit as stated by L'Hopital's
rule.
The numerator of y is 26*x and the denominator of y is x + 26.
The limit of the numerator as x -> oo is oo and the limit of the denominator as
x -> oo is oo.
The derivative of the numerator is 26 and the derivative of the denominator is
1.
The limit of the derivative of the numerator as x -> oo is 26 and the limit of
the derivative of the denominator as x -> oo is 1.
The exponent of the limit of y as x -> oo is 26.
The limit of y as x -> oo is exp(26).
```

1d

[5]:
```
y_lim = limit(y, x, oo)
print(f"Using the limit function in sympy, the limit of y as x -> oo is {y_lim}.
↪")
print(f"We can see that the limit of y as x -> oo is {exp(exponent)} and the␣
↪limit of y as x -> oo using the limit function in sympy is {y_lim}.")
print(f"Since they are the same value, we can conclude that the limit of y as x␣
↪-> oo is {exp(exponent)}.")
```

```
Using the limit function in sympy, the limit of y as x -> oo is 26.
We can see that the limit of y as x -> oo is exp(26) and the limit of y as x ->
oo using the limit function in sympy is 26.
Since they are the same value, we can conclude that the limit of y as x -> oo is
exp(26).
```

0.1.2 Question 2

2a

[6]:
```
"""
Suppose a photograph with a width of 42 inches and a
height of 50 inches is placed in the blue frame of the

round billboard shown. The margins between the rect-
angular frame and the picture are 10 inches at the top

and bottom and 4 inches on the sides. What would the
radius of the billboard be that would fit a photograph
of such dimensions?
```

```
"""

# given:
# width = 42 inches
# height = 50 inches
# top and bottom margins = 10 inches
# left and right margins = 4 inches

# the outer most rectangle
w = 42 + (2 * 4)
h = 50 + (2 * 10)
# from the center of the rectangle to the top right corner
# get the distance, which is the radius
d = sqrt((w / 2)**2 + (h / 2)**2)
print(f"The radius of the billboard is {d} inches.")
```

```
The radius of the billboard is 43.0116263352131 inches.
```

2b

[7]:
```
"""
A billboard with radius 55 inches is designed like the
figure to the left. Determine the dimensions, a and b,
of the largest picture that could fit in the frame.
"""

# R^2 = (b / 2)^2 + (a / 2)^2
# R^2 = 1/4 (b^2 + a^2)
# 4R^2 = b^2 + a^2
# b = sqrt(4R^2 - a^2)
a = Symbol('a')
b = sqrt(4 * (55 ** 2) - a ** 2)
# A = (a - 8)(b - 20)
A = (a - 8) * (b - 20)
A_diff = diff(A, a)
a_ = nsolve(A_diff, a, 70)
b = b.subs(a, a_)
a = a_

# check the radius
assert(sqrt((a / 2)**2 + (b / 2)**2) == 55)

print(f"The dimensions of the largest picture that could fit in the frame are␣
↪{a} inches by {b} inches.")
```

```
The dimensions of the largest picture that could fit in the frame are
74.4148590423809 inches by 81.0088189872101 inches.
```

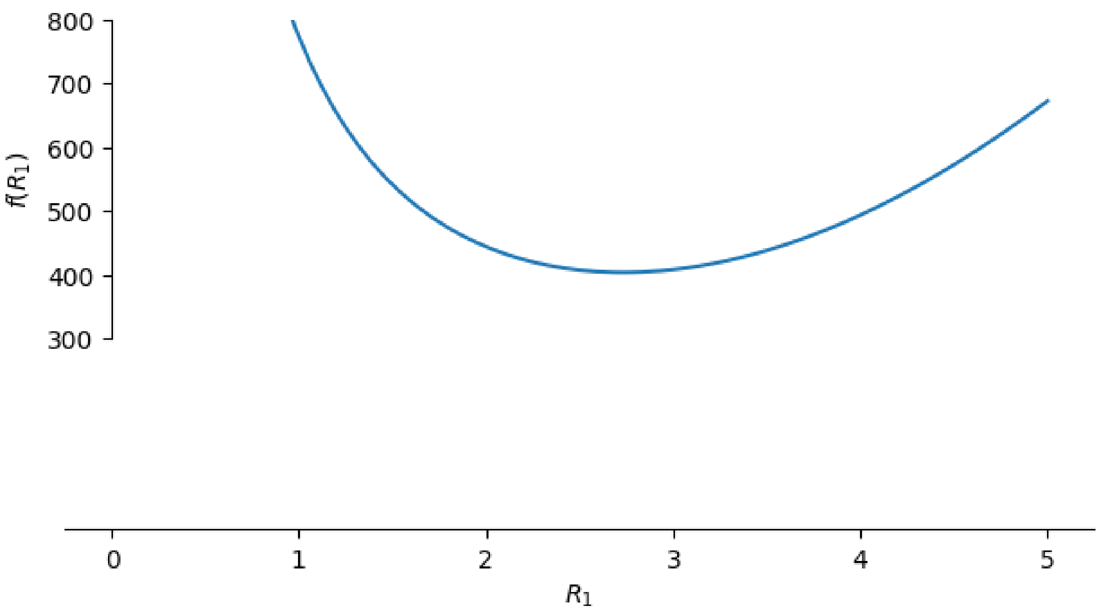
0.1.3 Question 3

3a

[8]:
```
# R_2 = 2R_1
# V = 1/3*pi*h*(R_1^2 + R_2^2 + R_1*R_2)
# SA = pi*(R_1 + R_2)*sqrt((R_2-R_1)^2 + h^2)+pi*(R_1^2 + R_2^2)

# V = 590
# minimize SA
R_1, R_2, h = symbols('R_1 R_2 h')
V = 1/3 * pi * h * (R_1**2 + R_2**2 + R_1 * R_2)
SA = pi * (R_1 + R_2) * sqrt((R_2 - R_1)**2 + h**2) + pi * (R_1**2 + R_2**2)
# R_2 = 2 * R_1
V = V.subs(R_2, 2 * R_1)
h_ = solve(V - 590, h)[0]
SA = SA.subs(h, h_).subs(R_2, 2 * R_1)
SA_diff = diff(SA, R_1)

# plot the optimization function on R_1 x = [0, 5] with y [300, 800]
plot(SA, (R_1, 0, 5), ylim=(300, 800))
print(f"The graph of the optimization function is shown above.")
```




```
The graph of the optimization function is shown above.
```

3b

[9]:
```
R_1_ = nsolve(SA_diff, R_1, 3)
R_2_ = 2 * R_1_
```

```
h_ = h_.subs(R_1, R_1_).subs(R_2, R_2_)
print(f"The radius of the top of the cone is {R_1_} inches, the radius of the bottom of the cone is {R_2_} inches, and the height of the cone is {h_}.")
```

```
The radius of the top of the cone is 2.73235307040265 inches, the radius of the
bottom of the cone is 5.46470614080531 inches, and the height of the cone is
10.7808181056803.
```

0.1.4 Question 4

4a

[10]:
```
# given f''(x) = 5 / (x + 1)^2
# f'(0) = 3
# f(0) = 9
# find f'(x) and f(x)

f_double_prime = 5 / ((x + 1) ** 2)
f_prime = integrate(f_double_prime, x)

# solve for C
C = symbols('C')
C = solve(f_prime.subs(x, 0) - 3 + C, C)[0]
f_prime = f_prime + C

print(f"The anti-derivative of f''(x) is f'(x) = {f_prime}.")

f = integrate(f_prime, x)

# solve for C
C = symbols('C')
C = solve(f.subs(x, 0) - 9 + C, C)[0]
f = f + C

print(f"The anti-derivative of f'(x) is f(x) = {f}.")
```

```
The anti-derivative of f''(x) is f'(x) = 8 - 5/(x + 1).
The anti-derivative of f'(x) is f(x) = 8*x - 5*log(x + 1) + 9.
```

4b

[11]:
```
# f(1) = f(4) = 10
# find f(x)

f_prime_prime = 5 / ((x + 1) ** 2)
f_prime = integrate(f_prime_prime, x)
x, C, D = symbols('x C D')
f = integrate(f_prime, x) + C*x + D
```

```python
# system of equations
eq1 = f.subs(x, 1) - 10
eq2 = f.subs(x, 4) - 10

# solve eq1 in terms of C
eq1 = solve(eq1, C)[0]

# plug eq1 into eq2
eq2 = eq2.subs(C, eq1)

# solve eq2 in terms of D
eq2 = solve(eq2, D)[0]

# plug eq1 and eq2 into f
f = f.subs(C, eq1)
f = f.subs(D, eq2)

print(f"The function f(x) is {f.simplify()}.")
```

```
The function f(x) is -5*x*log(2)/3 + 5*x*log(5)/3 - 5*log(x + 1) - 5*log(5)/3 +
20*log(2)/3 + 10.
```