Test Notebook - James Amidei

This is a test jupyter notebook for Advanced Lab 1.

In a markdown cell, we can shockingly format our text using markdown.

We can bold things by adding `**` on either side of our text like below.

`**Here is some bolded text**` = **Here is some bolded text**

Similarly, we can italicize text by adding a single `*` on either side of text like below.

`*Here is some text that's italicized*` = *Il corsivo non significa che il testo è tradotto in italiano.*

We can add equations by using `$` on either side of some math. For example

$$E = \gamma m_0 c^2$$

We've already seen a couple of times above, we can seperate commands by adding ``` around them. If add three on either side, we can use that same character to seperate chunks of code. For example:

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

Speaking of code, let's run that code in a code cell below.

In [7]:

```python
def factorial(n): # Function that allows us to find the factorial of a nu
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

factorial(500)
```

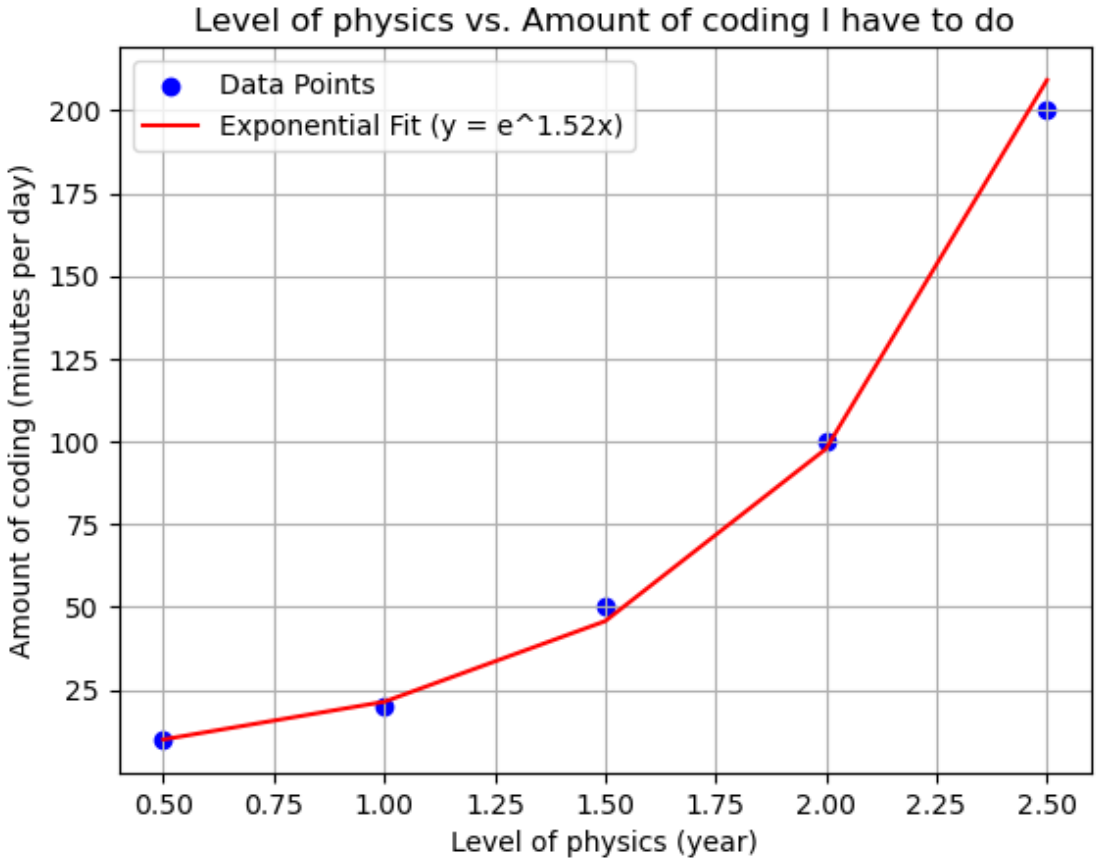
Out[7]:

```
122013682599111006870123878542304692625357434280319284219241358838584537
315388199760549644750220328186301361647714820358416337872207817720048078
520515932928547790757193933060377296085908627042917454788242491272634430
567017327076946106280231045264421887878946575477714986349436778103764427
403382736539747138647787849543848959553753799042324106127132698432774571
554630997720278101456108118837370953101635632443298702956389662891165897
476957208792692887128178007026517450776841071962439039432253642260523494
585012991857150124870696156814162535905669342381300885624924689156412677
565448188650659384795177536089400574523894033579847636394490531306232374
906644504882466507594673586207463792518420045936969298102226397195259719
094521782333175693458150855233282076282002340262690789834245171200620771
464097945611612762914595123722991334016955236385094288559201872743379517
301458635757082835578015873543276888868012039988238470215146760544540766
353598417443048012893831389688163948746965881750450692636533817505547812
864000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000
```

As a final note, here's a plot.

In [3]:

```
import matplotlib.pyplot as plt
import numpy as np

physics = [0.5, 1, 1.5, 2, 2.5]
coding = [10, 20, 50, 100, 200]

fig, ax = plt.subplots()

plt.scatter(physics, coding, label='Data Points', color='blue')

params = np.polyfit(physics, np.log(coding), 1)
a, b = params
y_fit = np.exp(a * np.array(physics) + b)

plt.plot(physics, y_fit, label=f'Exponential Fit (y = e^{a:.2f}x)', color=

plt.xlabel('Level of physics (year)')
plt.ylabel('Amount of coding (minutes per day)')
plt.title('Level of physics vs. Amount of coding I have to do')
plt.legend()

plt.grid(True)
plt.show()
```



In [ ]: