Untitled7

June 7, 2019

In [1]:
```
#a) Use the linspace() method to generate ten integers between 1 and 10
#b) Display the numbers
import numpy as np
```

In [8]:
```
x1 = np.linspace(1, 10, 10)
```

In [9]:
```
print('x1:',x1)
```

```
x1: [ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
```

In [37]:
```
#question 2
x= [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]
```

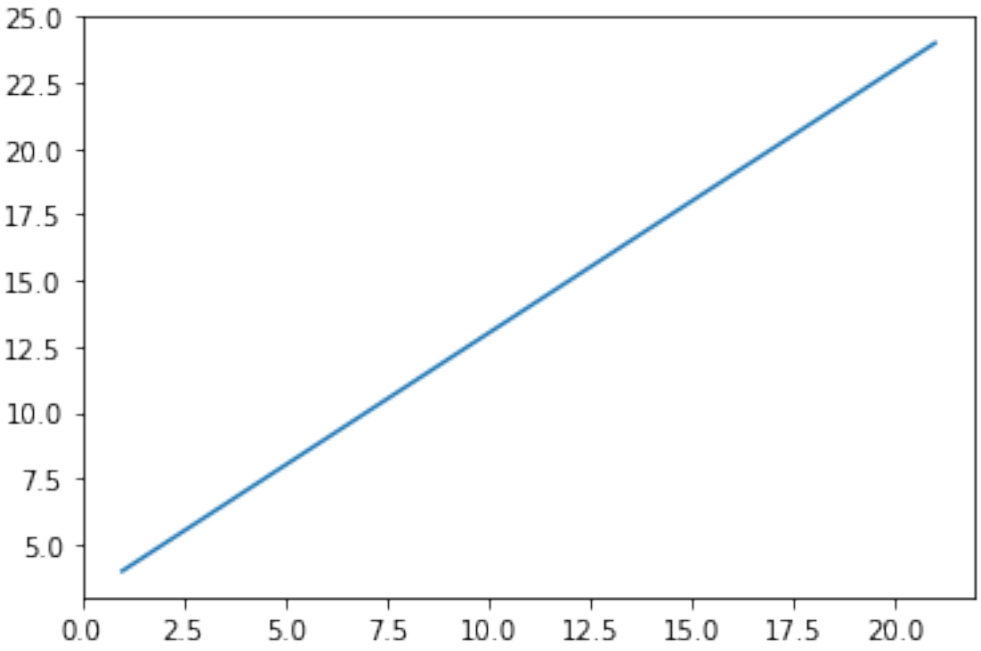
In [38]:
```
y=[4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]
```

In [39]:
```
import matplotlib.pyplot as plt
```

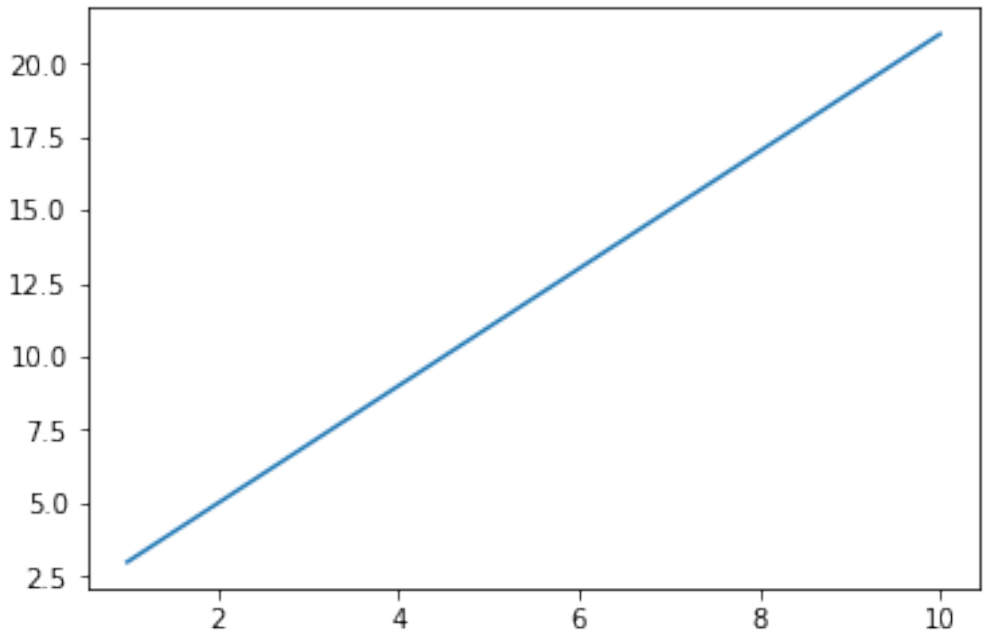
In [40]:
```
slope,intercept = np.polyfit(x,y,1)
```

In [41]:
```
plt.plot(x,y)
```

Out[41]:
```
[<matplotlib.lines.Line2D at 0x261bedd82b0>]
```



```
In [42]: #question 3
         x = [1,2,3,4,5,6,7,8,9,10]

In [120]: y = [3,5,7,9,11,13,15,17,19,21]
          #only first 10 values are taken using y=mx+c, becaiuse polyfit needs to have same le

In [121]: import matplotlib.pyplot as plt
          import numpy as np

In [122]: slope,intercept = np.polyfit(x,y,1)

In [123]: print(slope)

1.9999999999999996


In [124]: print(intercept)

0.9999999999999959


In [125]: print(intercept.round(0))

1.0


In [126]: plt.plot(x,y)

Out[126]: [<matplotlib.lines.Line2D at 0x261bf338d30>]
```



```
In [127]: # question 3
          import numpy as np

In [128]: from sklearn.linear_model import LinearRegression

In [62]: from scipy import stats
         import numpy as np

In [77]: #question 4
         #x = [1,2,3,4,5,6,7,8,9,10]
         from scipy import stats

In [110]: from sklearn.linear_model import LinearRegression
          import numpy as np

In [111]:
          x = [1,2,3,4,5,6,7,8,9,10]

In [112]: y = [3,5,7,9,11,13,15,17,19,21]

In [113]: slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)

In [114]: print("r-squared:", r_value**2)

r-squared: 1.0

In [115]: print(slope)

2.0

In [116]: print(intercept)

1.0

In [117]: print(r_value)

1.0

In [118]: print(p_value)

4.375000000000076e-80

In [119]: print(std_err)

0.0

In [107]:

In [108]:

In [ ]:
```