In [1]:
```
                                        ###### Homework 4 - Part 3 ###########
```

In [2]:
```
#Note the following:

#1. Each question is worth 1 point
#2. Write your name and student ID below
#3. Remember to save your file and upload to Canvas
```

In [3]:
```
#Student Name: Julia Troni
```

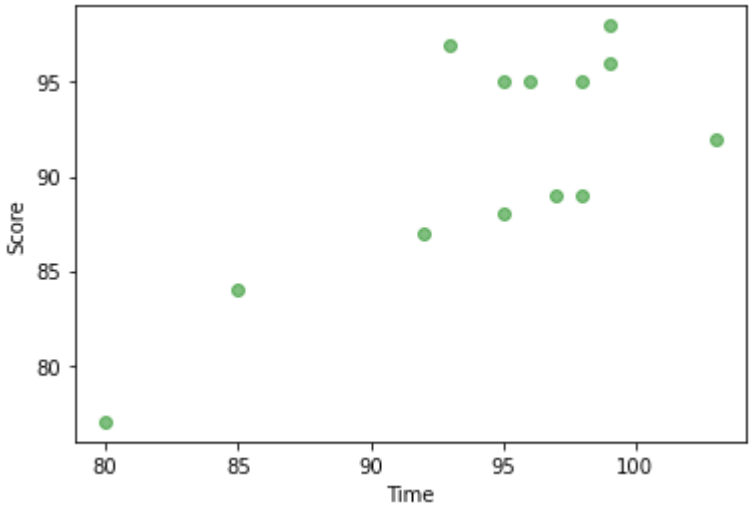
In [4]:
```
#Student ID: 109280095
```

In [ ]:
```

```

In [5]:
```
#Read the context below to answer Questions 1 - 10 below:

#It is plausible to think that the longer a student studies mathematics the better they

#Suppose you set out to prove/disprove the theory above. You talk to 10 randomly select
#mathematics in Fall of 2022 across several universities and obtain the following raw d
# which you store in the following variables:


#Time: defined as number of hours a friend spent studying mathematics in one semester

#Score: defined as the raw score in points (out of a maximum of 100) that a friend earn

#Your goal is to use simple linear regression to test the following hypotheses:

#Null hypothesis: There is no statistically significant relationship between Score and
#Alternative hypothesis: There is a statistically significant relationship between Scor

#You choose to test the Null against the Alternative at 5% (0.05) significance level (a

#The next cell shows the data and shows a scatter plot for the data you collected
```

In [6]:
```
#MAKE SURE YOU RUN THIS CELL FIRST BEFORE YOU ANSWER ANY QUESTIONS BELOW

#Import relevant libraries

import matplotlib.pyplot as plt
from scipy import stats

#Data you collected for all 10 students

Time = [99,96,93,98,97,92,103,85,80,98,95,95,99]
Score = [96,95,97,95,89,87,92,84,77,89,88,95,98]

#Show scatterplot of your data
fig, ax = plt.subplots()
```

```
ax.scatter(Time, Score, c="green", alpha=0.5, label="Scatterplot of Score vs. Time Spen

ax.set_xlabel("Time")
ax.set_ylabel("Score")

plt.show()
```



In [7]:

```
#Question 1

#Suppose we define the following variables to hold outputs from our linear regression a

#slope: variable to hold the slope coefficient from the regression
#intercept: variable to hold the intercept from the regression
#corr_coefficient: variable to hold the correlation coefficient from the regression
#p_value: variable to hold the p-value of the slope coefficient from the regression
#std_err: variable to hold the standard error of the regression

#Using stats.linregress from the stats module you imported above,
#Write Python code to obtain the following (in the order written) from a regression of

#slope, intercept, corr_coefficient, p_value, std_err


slope, intercept, corr_coefficient, p_value, std_err= stats.linregress(Time,Score)

print(" Slope: {:2f}\n Intercept: {:.2f}\n Corr_coeff: {:2f}\n pval: {:2f}\n stderr: {:
```

```
Slope: 0.750676
Intercept: 19.90
Corr_coeff: 0.766112
pval: 0.002259
stderr: 0.189879
```

In [ ]:

In [18]:

```
#Question 2:

#Write Python code to print slope

#Write your answer below:
```

```
print(" Slope: {:.2f}".format(slope))
```

```
Slope: 0.75
```

In [ ]:

In [9]:
```
#Question 3:

#Write Python code print p_value (the p-value of the coefficient on the variable Time)

print(" p_value: {:.5f}".format(p_value))
```

```
p_value: 0.00226
```

In [ ]:

In [10]:
```
#Question 4:

#Given the null and alternative hypothesis you set out to test above,
# do you think that the coefficient on Time is statistically significant? Why or Why no


#Write your answer below:
```

The null hypothesis is that there is no statistically significant realationship between score and time (i.e. slope=0). For this model, the p value = 0.002 < alpha = 0.05, thus we can say that that the coefficient on Time IS statistically significant. In other words, we reject the null hypothesis because pvalue<alpha.

In [ ]:

In [11]:
```
#Question 5

#Write Python code to print the intercept of the regression above

#Write your answer below:


print(" Intercept: {:.3f}".format(intercept))
```

```
Intercept: 19.898
```

In [ ]:

In [12]:
```
#Question 6

#Given your regression of Score on Time, write Score as a linear function of Time using
# the intercept and slope coefficients (round each to 2 decimal places) you obtained in
```

```
#Write your answer below:

#Score= 0.75*Time+ 19.90

#Create a function that uses the slope and intercept values to return a new value.
#This new value represents where on the y-axis the corresponding x value will be placed

#Score= 0.75*Time+ 19.90
def mymodel(time):
  return slope.round(2) * time + intercept.round(2)
```

In [ ]:

In [13]:
```
#Question 7

#How do you interpret the coefficient estimate (i.e. slope) of the variable Time?


#Write your answer below:

    #The slope is 0.75 which indicates that for each additional hour that a student stu
     #their score will increase by 0.75 points.
```

In [ ]:

In [14]:
```
#Question 8


#One of your friends says they plan to spend 90 hours studying mathatics this Fall
# and wants a prediction for their likely score.

#Using your model/function in Question 7 above, what would be your prediction for
# Score when Time = 90 hours?


#Write your answer below (round to 1 decimal places):



friendsScore=mymodel(90)
print("If friend spends 90 hours studying math, their likely score is {:.1f}".format(fr
```

```
If friend spends 90 hours studying math, their likely score is 87.4
```

In [ ]:

In [15]:
```
#Question 9

#Looking at the orginal data you collected for Scores and Time above,
#the student that spent 103 hours studying, actually scored 92 points in mathematics.

#1. Use your model in Question 6 to obtain a predicted Score given that Time = 103 hour
```

```
#2. Calculate what the error (actual Score value minus predicted Score value) is when T
#3. What is the absolute value of your calculated error (round to two decimal places?

#Write your answer below:

#1
pred= mymodel(103)
print("Predicted score: ", pred)
#2
error= abs(92-pred)
#3
print("Absolute Error: ",round(error,2))
```

```
Predicted score:  97.15
Absolute Error:  5.15
```

In [ ]:

In [16]:

```
#Question 10


#Using the information from the original data you collected from your friends and your
#calculate the root mean squared error (RMSE) for your model using the last three Time


#Round your answer to two decimal places

#Write your answer below:


from sklearn.metrics import mean_squared_error
import numpy as np

last3Time = [95,95,99]
last3Score = [88,95,98]

last3predictions=  list(map(mymodel, last3Time))


#obtain RMSE for linear model
rmse = np.sqrt(mean_squared_error(last3Score, last3predictions))

#display RMSE
print("The root mean squared error is : ", rmse.round(2))
```

```
The root mean squared error is :  3.63
```

In [ ]:

In [17]:

```
                                        ####End of Homework 3 - Part 3###
```