CS 237 Fall 2018, Homework 11

Due date: PDF file due Friday December 6th @ 11:59PM in GradeScope with 12-hour grace period

No late submissions accepted

General Instructions

Please complete this notebook by filling in solutions where indicated. Be sure to "Run All" from the Cell menu before submitting.

You may use ordinary ASCII text to write your solutions, or (preferably) Latex. A nice introduction to Latex in Jupyter notebooks may be found here: http://data-blog.udacity.com/posts/2016/10/latex-primer/ (http://data-blog.udacity.com/posts/2016/10/latex-primer/)

As with previous homeworks, just upload a PDF file of this notebook. Instructions for converting to PDF may be found on the class web page right under the link for homework 1.

In [1]:

```
# General useful imports
import numpy as np
from numpy import arange,linspace,mean, var, std, corrcoef, transpose, ones,log
from numpy.linalg import inv
from scipy.stats import pearsonr
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.mlab as mlab
from numpy.random import random, randint, uniform
import math
from collections import Counter
import pandas as pd
%matplotlib inline


# Basic Numpy statistical functions

X = [1,2,3]

# mean of a list
mean(X)             # might need to use np.mean, np.var, and np.std

# population variance
var(X)

# sample variance    ddof = delta degrees of freedom, df = len(X) - ddof
var(X,ddof=1)

# population standard deviation
std(X)

# sample standard deviation
std(X,ddof=1)


# Scipy statistical functions

# Scipy Stats Library Functions, see:
# https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.statistics.html


# Calculate the correlation coefficient rho(X,Y)


def rho(X,Y):
    return corrcoef(X,Y)[0][1]

def R2(X,Y):
    return corrcoef(X,Y)[0][1] ** 2

# Utility functions


# Round to 2 decimal places
def round2(x):
    return round(float(x)+0.00000000001,2)

# Round to 4 decimal places
def round4(x):
```

```
    return round(float(x)+0.00000000001,4)

def round4List(X):
    return [round(float(x)+0.00000000001,4) for x in X]

def probToPercent(p):
    pc = p*100
    if round(pc) == pc:
        return str(round(pc)) + "%"
    else:
        return str(round(pc,2))+ "%"
```

Homework 11 General Instructions

This homework contains programming problems with some commentary on interpreting the results. Its goal is simply to exercise your understanding of regression and Monte Carlo algorithms, as covered in lecture before Thanksgiving break.

Problem One: Linear Regression

For this problem, in Part A you will fill in the template for a function that draws a scatterplot, the linear regression line, the midpoint, and the relevant statistics, as shown in lecture. In lab 9, problem 1, you drew a diagram just to practice using the matplotlib library, and you may want to refer to your solution to get started.

For Part B you will simply draw the regression line and residual plot for some data related to our class (anonymous of course!). For Part C you will think about what you are seeing and answer some questions about the data and whether a linear model is appropriate for this data.

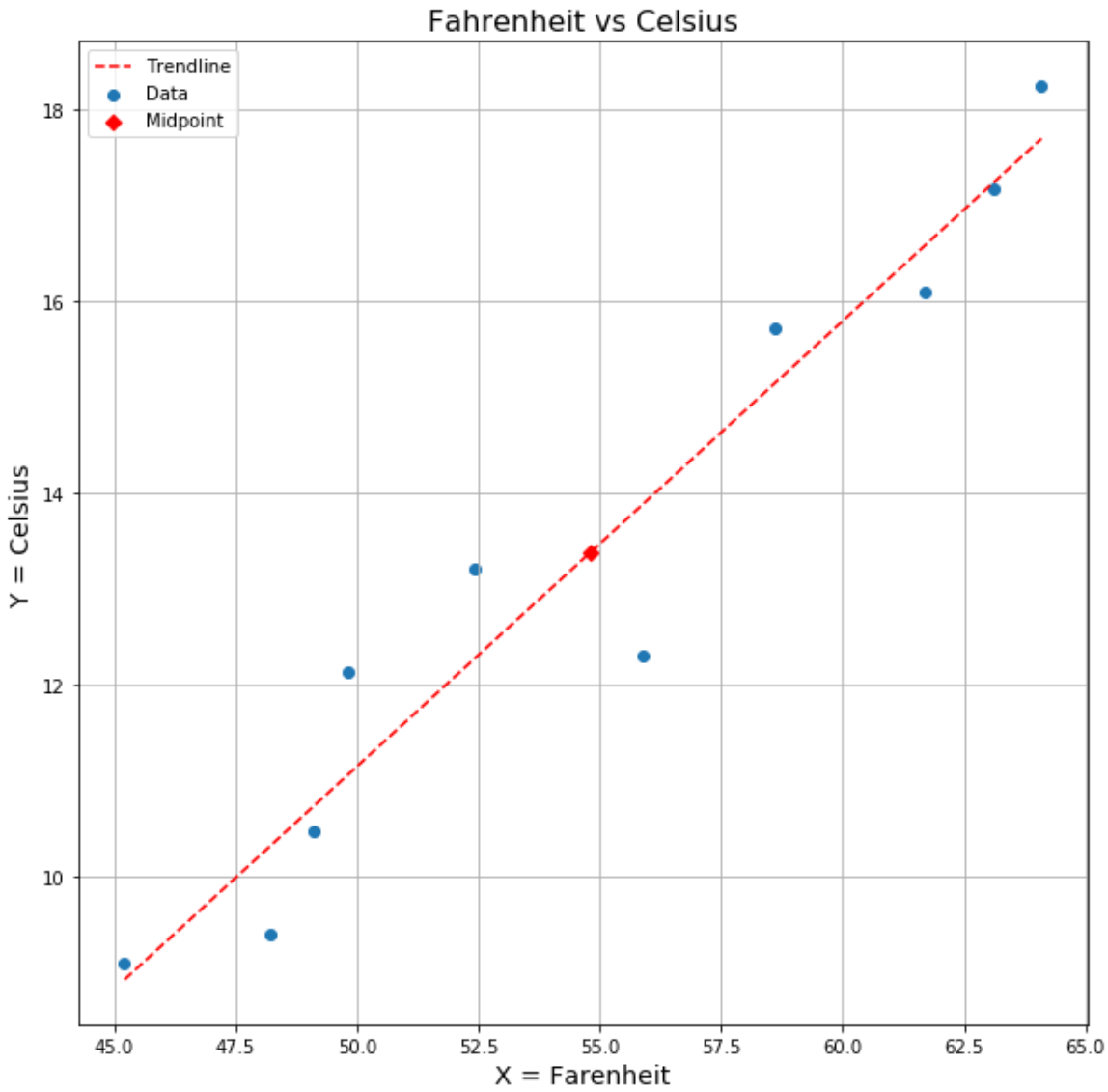
Part A (Farentheit vs Celsius)

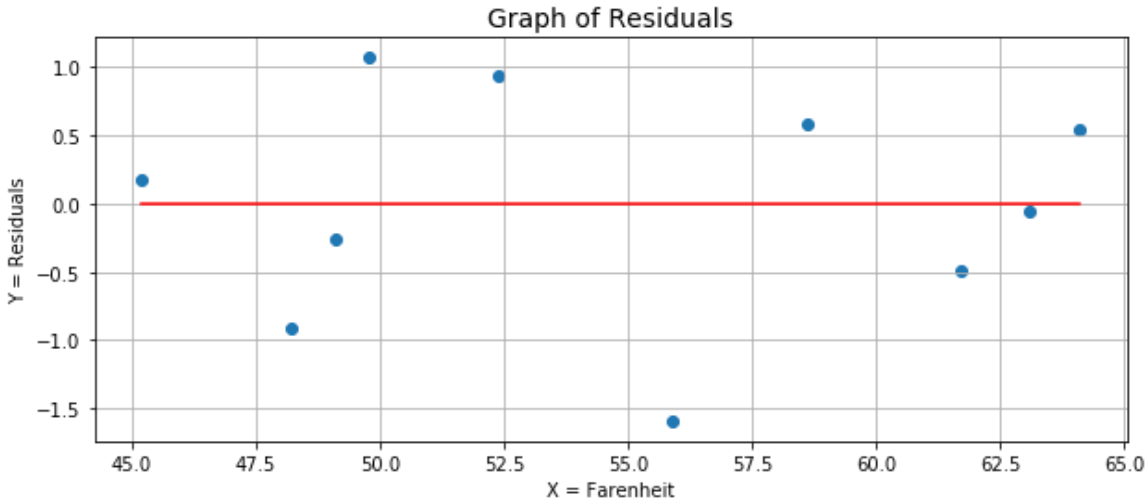
Complete the following template to draw the example Farenheit/Celsius data from lecture

In [2]:

```
# Draw scatterplot for bivariate data and draw linear regression line
# with midpoint (mux,muy)

def LinearRegression(X,Y,titl="Linear Regression", xlab="X",ylab="Y"):

    n = len(X)

    # Calculate stats and values m (theta1) and b (theta0)

    mux = np.mean(X)     # your code here, use statistical functions as shown ab
ove
    muy = np.mean(Y)
    stdx = np.std(X)
    stdy = np.std(Y)
    r = rho(X,Y)      # rho
    r2 = R2(X,Y)     # r^2



    # Predicted values from regression line
    m = r * stdy / stdx
    b = muy - m*mux

    Yhat = [(m*X[i]+b) for i in range(n)]


    E = [(Y[i] - Yhat[i]) for i in range(n)]

    # residual sum of squares -- deviations of data from line
    rss = sum( [ e**2 for e in E])

    # explained sum of squares -- deviation of line from mean of y
    ess = sum( [ (Yhat[i] - muy)**2 for i in range(n)])

    # total sum of squares -- deviation of data from mean of y
    tss = sum( [ (Y[i] - muy)**2 for i in range(n)] )

    plt.figure(figsize=(10,10))
    plt.grid()

    plt.scatter(X,Y,label="Data")
    plt.plot(X,Yhat,'r--',label="Trendline")
    plt.scatter([mux],[muy],marker='D',color='red',label="Midpoint")
    plt.title(titl,fontsize=16)
    plt.legend()
    plt.xlabel(xlab,fontsize=14)
    plt.ylabel(ylab,fontsize=14)
    plt.show()

    plt.figure(figsize=(10,4))
    plt.title("Graph of Residuals",fontsize=14)
    plt.grid()
    plt.xlabel(xlab)
    plt.ylabel("Y = Residuals")
    Yhat = [0 for x in X ]
    plt.scatter(X,E)
    plt.plot(X,Yhat,color='red')
    plt.show()
```

```
    print("\nmean(x):\t" + str(round4(mux)) + "\tstd(x):\t" + str(round4(stdx)))
    print("mean(y):\t" + str(round4(muy)) + "\tstd(y):\t" + str(round4(stdy)))
    print("\nrho:   " + str(round4(r)) + "\tR^2:   " + str(round4(r2)))
    print("\nResidual SS:   " + str(round4(rss)) + "\tExplained SS: " + str(roun
d4(ess)) + "\tTotal SS:   " + str(round4(tss)))

    if(b >= 0):
        print("\nRegression Line: y = " + str(round4(m)) + " * x + " + str(round
4(b)))
    else:
        print("\nRegression Line: y = " + str(round4(m)) + " * x - " + str(round
4(-b)))

Xfarenheit = [45.2, 48.2, 49.1, 49.8, 52.4, 55.9, 58.6, 61.7, 63.1, 64.1]
Ycelsius = [9.0994, 9.4023, 10.4809, 12.132, 13.2032, 12.303, 15.7304, 16.1, 17.
1773, 18.2468]


LinearRegression(Xfarenheit,Ycelsius,"Fahrenheit vs Celsius",xlab="X = Farenhei
t",ylab="Y = Celsius")
```






```
mean(x):         54.81   std(x): 6.4623
mean(y):         13.3875 std(y): 3.1037

rho:   0.9664   R^2:   0.9339

Residual SS:   6.3631   Explained SS: 89.9635   Total SS:   96.3265

Regression Line: y = 0.4641 * x - 12.0519
```

Now we will test our regression skills on some data from our class....

In [3]:

```
studs = pd.read_csv('http://www.cs.bu.edu/fac/snyder/cs237/Homeworks,%20Labs,%20and%20Code/classdata.csv')
print(studs[:10])
```

```
    GPA  Midterm  HWAvg
0  3.51     86.0  86.29
1  3.34     91.0  51.18
2  3.94     92.0  94.06
3  3.20     78.0  38.82
4  3.91     97.0  89.18
5  3.59     92.0  88.76
6  3.17     97.0  68.65
7  3.45     97.0  55.06
8  3.45     84.0  92.47
9  3.50     94.0  89.76
```

In [4]:

```
GPA = studs['GPA']
MID = studs['Midterm']
HWS = studs['HWAvg']
```

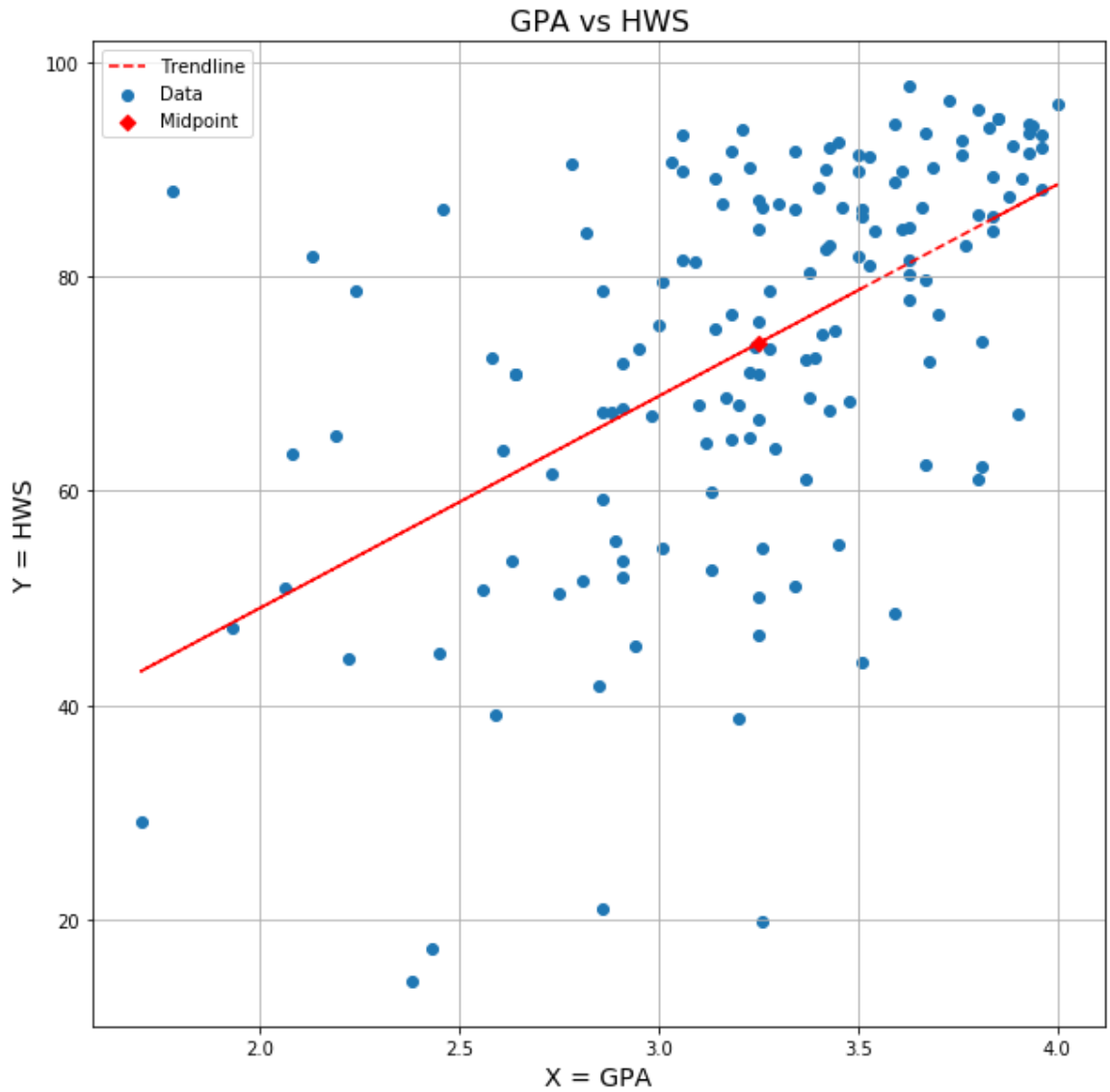
Part B (GPA vs HWS)

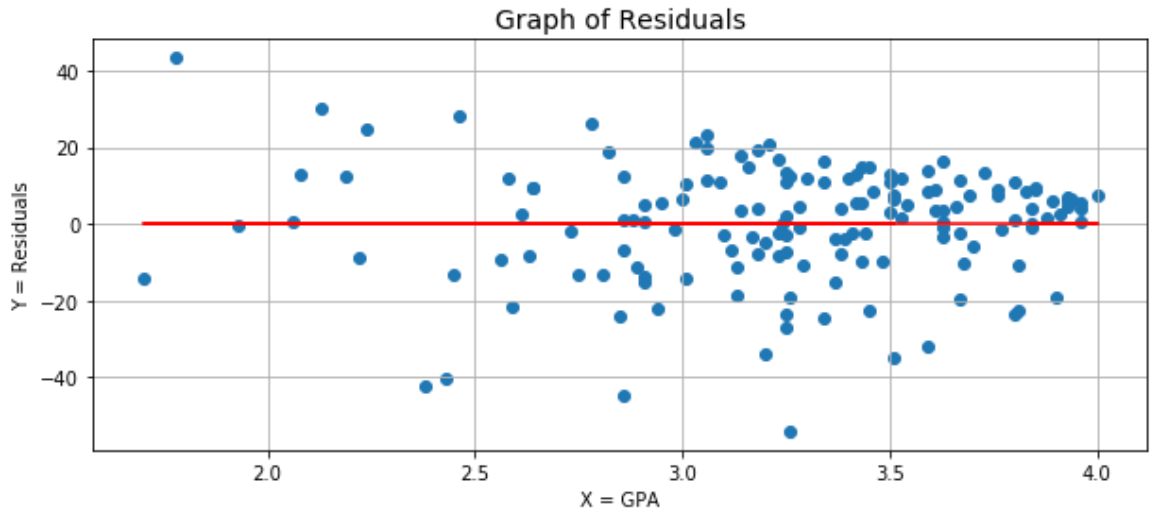
Apply the function from Part A to the following data:

```
X = GPA
Y = HWS
```

In [5]:

```
LinearRegression(GPA,HWS,"GPA vs HWS",xlab="X = GPA",ylab="Y = HWS")
```


GPA vs HWS
Trendline
Data
Midpoint
Y = HWS
X = GPA
100
80
60
40
20
2.0
2.5
3.0
3.5
4.0




```
mean(x):         3.249    std(x): 0.4903
mean(y):         73.7222 std(y): 18.0138

rho:    0.5371    R^2:    0.2885

Residual SS:    35093.5105        Explained SS: 14229.7885         Tota
l SS:    49323.299

Regression Line: y = 19.7359 * x + 9.6
```

Part C

Looking at the data and the statistics for Part B, answer the following questions to the best of your ability:

> (i) Just looking at the graphical display of the data, would you think there is a linear trend to this data, i.e., is there a correlation between GPA and homework scores in CS 237? What do you see?

> (ii) Now, what does the R^2 value tell you about fitting a linear model to this data?

Linear regression isn't always appropriate, and not only because of the R^2 score. Please read through the following page outlining the principal conditions necessary for linear regression:

https://www.statisticshowto.datasciencecentral.com/assumptions-conditions-for-regression/ (https://www.statisticshowto.datasciencecentral.com/assumptions-conditions-for-regression/)

> (iii) Do you see any other problems, related to the conditions you read about above? (Hint: look at the residual plot).) Can you think of any reason why this might be the case for this data set? (Hint: Just answer these by "eyeballing" the data, don't worry about doing a precise analysis.)

In [6]:

```
print("Part I")
print("There is a linear correlation between GPA and homework scores in CS 237 e
ven though there are some exceptions.")
print("This is because for most values, as the GPA increases, the homework score
s are better.")
print("Logically, if you think about this, homework counts as some portion of th
e GPA and therefore, if GPA is higher, hoemwork scores should be higher as wel
l.")

print()

print("Part II")
print("The R value squared tells that the model does not fit the data to be line
ar.")
print("The value of R squared is 0.2885, which is close to 0 than 100, means tha
t the model does not fit the data and therefore does not fit the data to be line
ar")

print()

print("Part III")
print("There exists other problems after reading about the conditions.")
print("There exists a pattern within the residual plot.")
print("As the value for GPA increases, the homework scores seem to be closer to
 the GPA and therefore the residual decreases, which is a pattern that should no
t be included.")
```

```
Part I
There is a linear correlation between GPA and homework scores in CS
237 even though there are some exceptions.
This is because for most values, as the GPA increases, the homework
scores are better.
Logically, if you think about this, homework counts as some portion
of the GPA and therefore, if GPA is higher, hoemwork scores should b
e higher as well.

Part II
The R value squared tells that the model does not fit the data to be
linear.
The value of R squared is 0.2885, which is close to 0 than 100, mean
s that the model does not fit the data and therefore does not fit th
e data to be linear

Part III
There exists other problems after reading about the conditions.
There exists a pattern within the residual plot.
As the value for GPA increases, the homework scores seem to be close
r to the GPA and therefore the residual decreases, which is a patter
n that should not be included.
```

Problem Two: Linear Regression

Now we will look at pairing the Midterm Score data with GPA. Again, you will apply your function from Problem One (A) to display the data and statistics, and then think about what you are seeing.

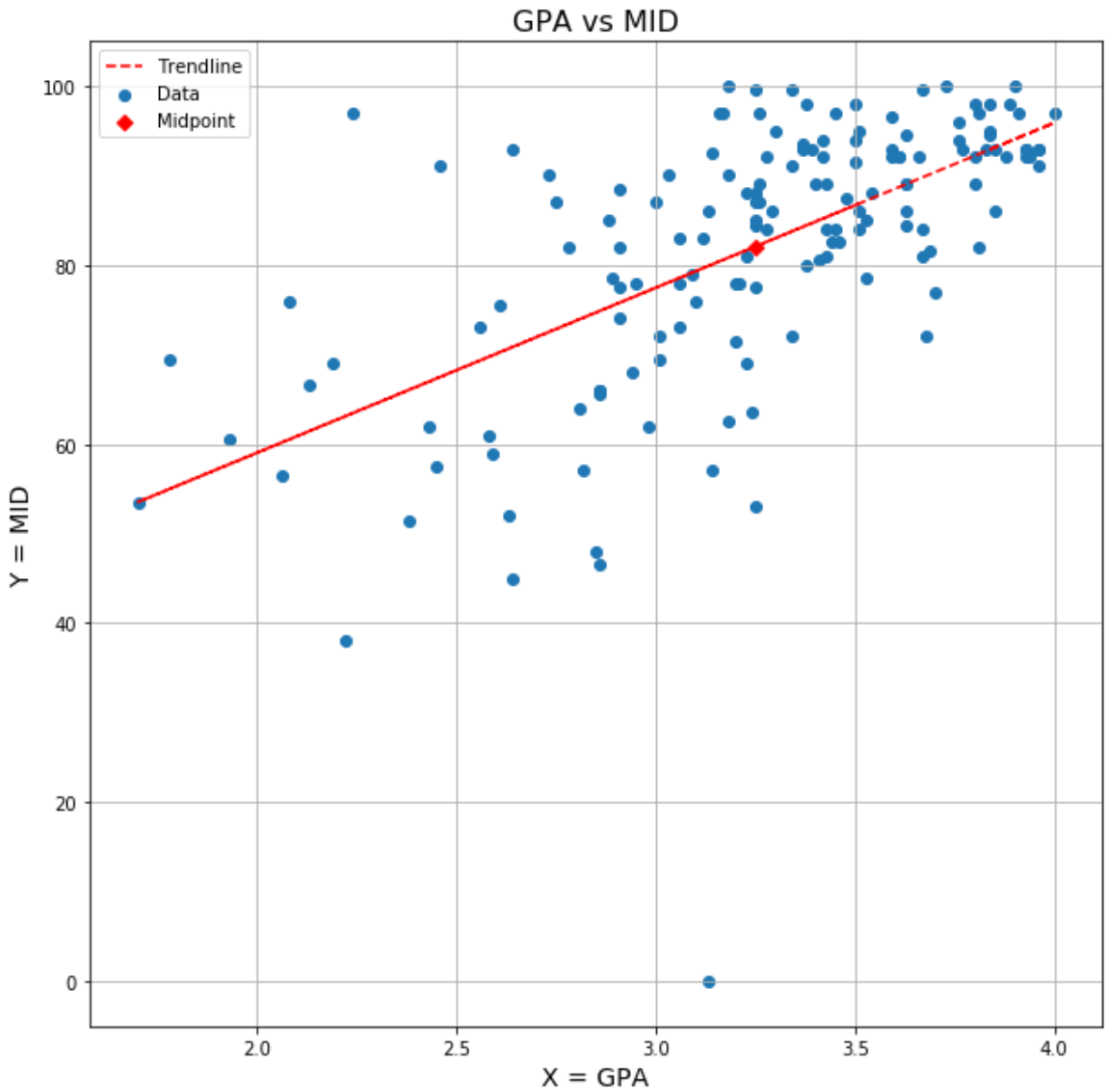
Part A (GPA vs MID)

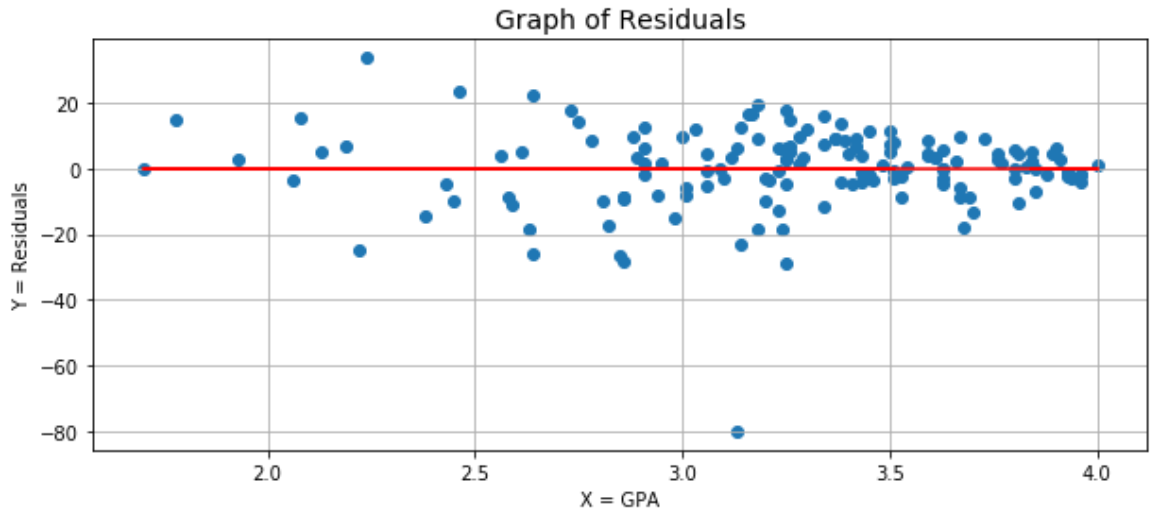
Apply the function from Part A to the following data:

```
X = GPA
Y = MID
```

In [7]:

```
LinearRegression(GPA,MID,"GPA vs MID",xlab="X = GPA",ylab="Y = MID")
```


GPA vs MID
Trendline
Data
Midpoint
Y = MID
X = GPA
0
20
40
60
80
100
2.0
2.5
3.0
3.5
4.0




```
mean(x):         3.249    std(x): 0.4903
mean(y):         82.0296 std(y): 15.1626

rho:    0.5952    R^2:    0.3542

Residual SS:    22567.1516        Explained SS: 12378.4652         Tota
l SS:    34945.6168

Regression Line: y = 18.4073 * x + 22.2239
```

Part B (Dealing with Outliers: GPA2 vs MID2)

In Part A there is clearly a violation of one of the conditions for linear regression, in that there is an outlier in the Midterm data, because of a student in the data that didn't take the midterm and then at some point dropped the class. It is always a serious question whether an inconvenient data point is really an outlier, but in this case it is clear that if a student dropped the class, then the data on homeworks AND midterm are not really relevant to what we want to know, which is the relationship between GPA and performance in two parts of the class marks for those who complete the class.

The outlier occurs at index 128:

In [8]:

```
print(studs[125:132])
```

```
      GPA  Midterm  HWAvg
125  3.81     82.0  62.29
126  3.84     95.0  84.24
127  2.95     78.0  73.24
128  3.13      0.0  52.59
129  3.30     95.0  86.82
130  3.59     96.5  48.53
131  2.85     48.0  41.82
```

Eliminate this data point from ALL three lists and call the new sets GPA2, HWS2, and MID2. [Hint: check out the `del` function in Python.]

Now display the data for GPA2 vs MID2 (as in Part A, but with the outlier removed).

In [9]:

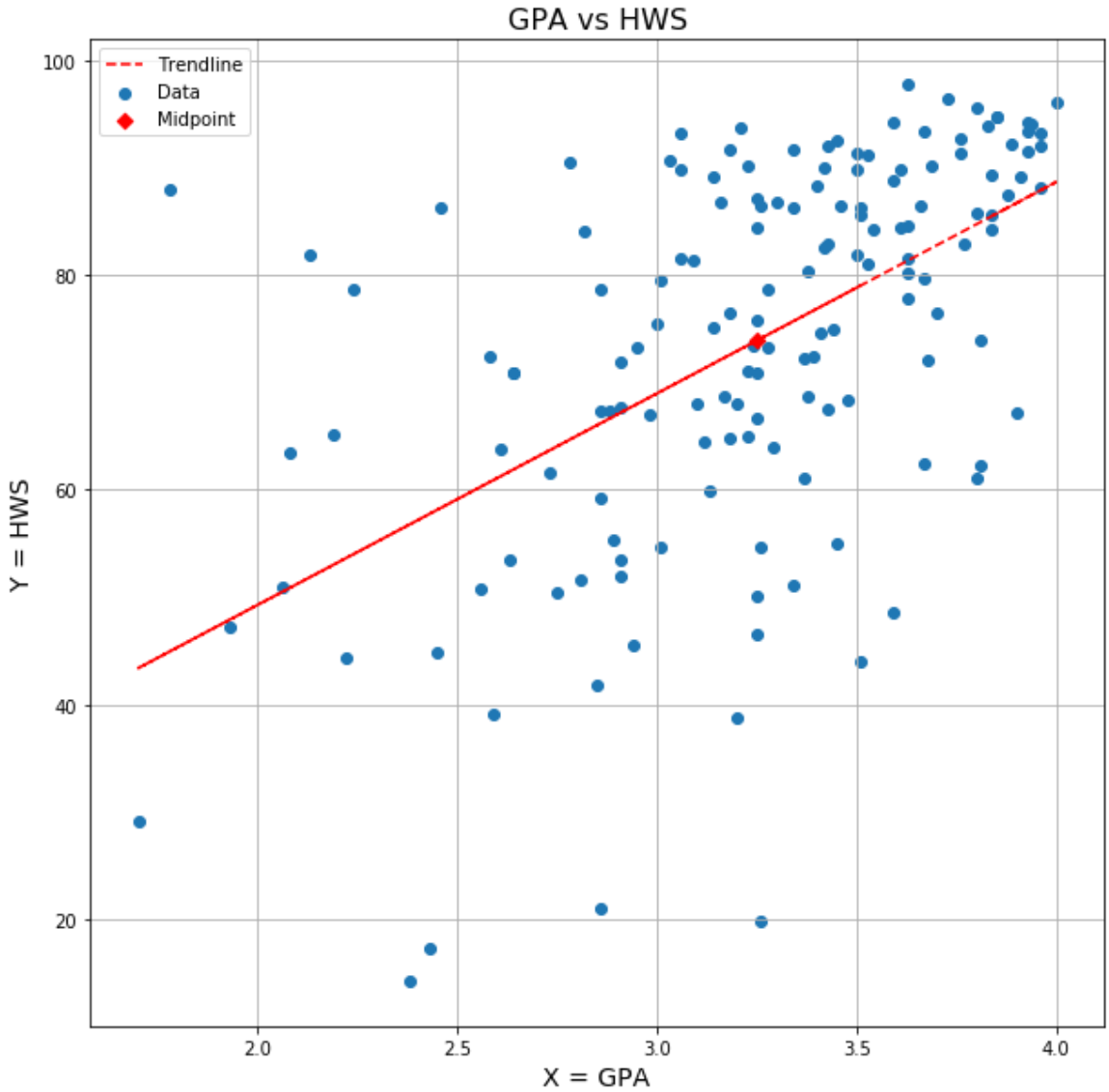
```
GPA2 = list(GPA)
del GPA2[128]
MID2 = list(MID)
del MID2[128]
HWS2 = list(HWS)
del HWS2[128]
```

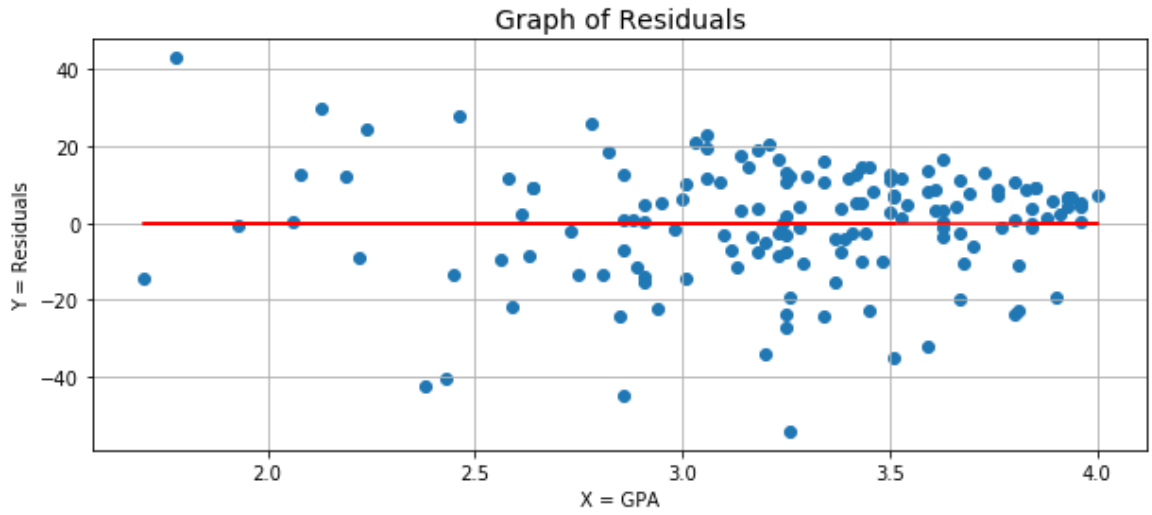
Part C (GPA2 vs HWS2)

Well, now we could wonder whether this new data (with one student eliminated who dropped the class) would change our results in Problem One. Redo the data display from Problem One (B) with this new data.

In [10]:

```
LinearRegression(GPA2,HWS2,"GPA vs HWS",xlab="X = GPA",ylab="Y = HWS")
```


GPA vs HWS
Trendline
Data
Midpoint
100
80
60
40
20
Y = HWS
2.0
2.5
3.0
3.5
4.0
X = GPA




```
mean(x):         3.2498  std(x): 0.4918
mean(y):         73.8621 std(y): 17.9908

rho:   0.5378   R^2:   0.2892

Residual SS:   34738.2215       Explained SS: 14135.5514        Tota
l SS:   48873.7729

Regression Line: y = 19.6743 * x + 9.9246
```

Part D

Now answer the following questions:

(i) How much did removing the outlier affect our results with GPA vs MID (Problem 2 A compared with 2 B)? In particular, look at $var(y)$ and the R^2 value.

(ii) What does this say (with this one example as evidence) about the sensitivity of R^2 to outliers in the data set?

(iii) How what you saw in 2 C differ with what you saw with Problem One (B)?

In [11]:

```
print("Part I")
print("The removal of the outlier affected the GPA vs MID by increasing the valu
e of R squared by 0.0007 and decreasing the value of standard deviation of Y by
 0.0023.")

print()

print("Part II")
print("Even though the value of R sqaured decreased, the amount it decreased was
only 0.0007 by the outlier, which shows that R sqaured does not get affected eas
ily by removing one data set.")
print()

print("Part III")
print("As stated in part I, the standard deviation for y has decreased and the v
alue of residual squared increased, which means that it is stronger linear fit t
han the graph in problem one section b")
print("However, erasing one value did not change much.")
print("The slope and the y-intercept also changed but did not change much as wel
l.")
```

```
Part I
The removal of the outlier affected the GPA vs MID by increasing the
value of R squared by 0.0007 and decreasing the value of standard de
viation of Y by 0.0023.

Part II
Even though the value of R sqaured decreased, the amount it decrease
d was only 0.0007 by the outlier, which shows that R sqaured does no
t get affected easily by removing one data set.

Part III
As stated in part I, the standard deviation for y has decreased and
the value of residual squared increased, which means that it is stro
nger linear fit than the graph in problem one section b
However, erasing one value did not change much.
The slope and the y-intercept also changed but did not change much a
s well.
```

Problem Three: Linear Regression

In this last problem on linear regression you will finish your analysis by comparing HWS2 and MID2 data, using the regression line to do some prediction, and thinking about what you have seen in all the problems so far.

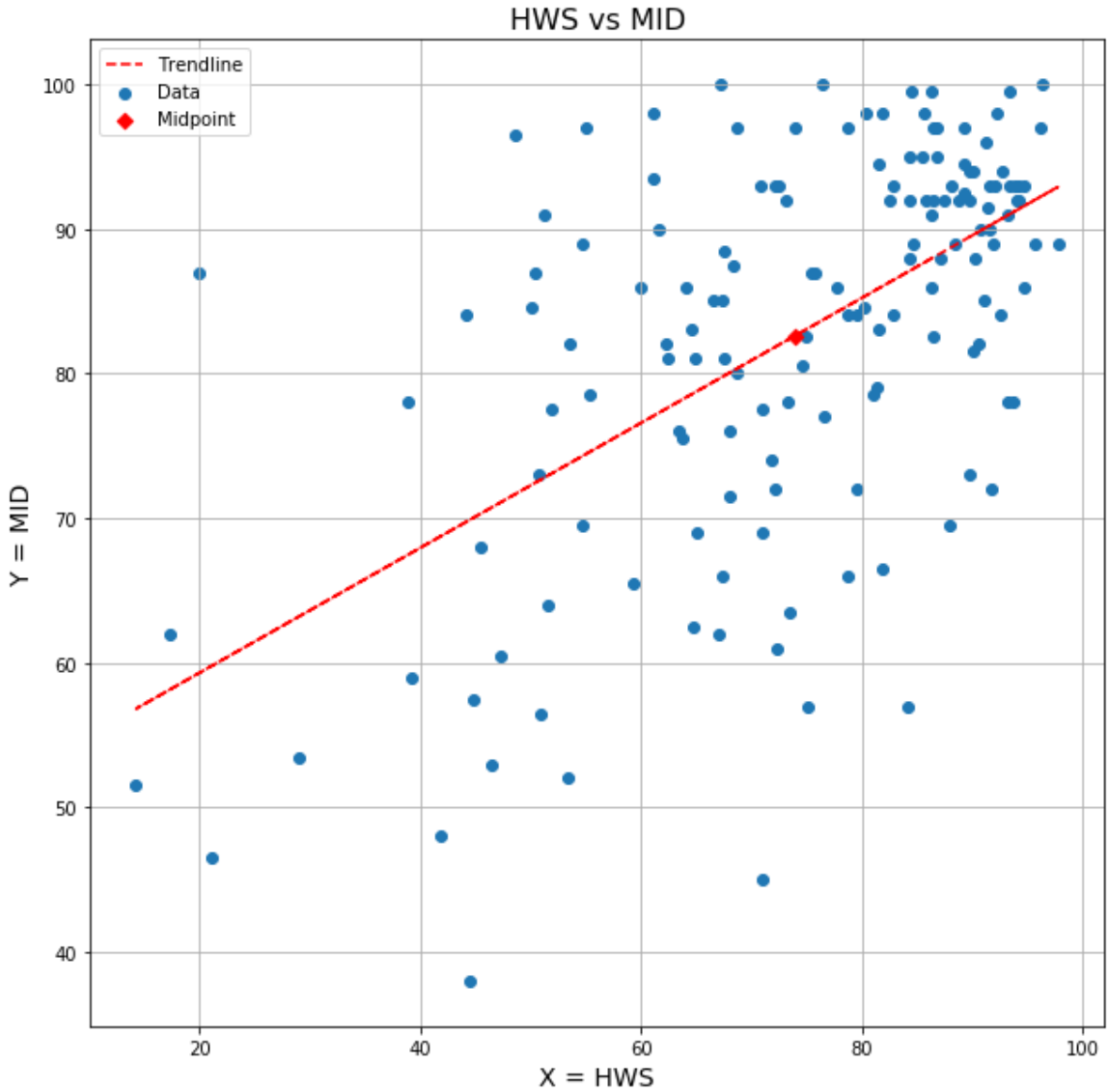
Part A

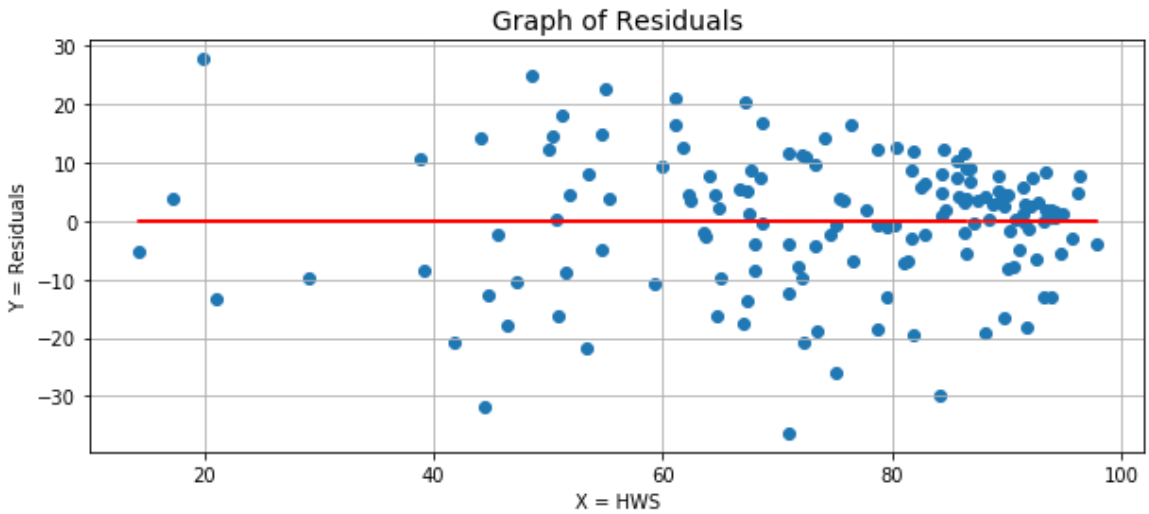
Apply the function from Problem One A to the following (modified) data:

```
X = HWS2
Y = MID2
```

In [12]:

```
LinearRegression(HWS2,MID2,"HWS vs MID",xlab="X = HWS",ylab="Y = MID")
```


HWS vs MID
Trendline
Data
Midpoint
Y = MID
X = HWS
100
90
80
70
60
50
40
20
40
60
80
100


Graph of Residuals
Y = Residuals
X = HWS
30
20
10
0
−10
−20
−30
20
40
60
80
100

```
mean(x):        73.8621 std(x): 17.9908
mean(y):        82.5728 std(y): 13.6591

rho:   0.5688   R^2:   0.3235

Residual SS:   19058.132        Explained SS: 9114.0667 Total SS:
28172.1987

Regression Line: y = 0.4318 * x + 50.6766
```

Part B

We eliminated a student from the class who dropped the course before the midterm. Hm, I wonder how our model would have predicted his/her score for homeworks (for which we have a score) and midterm (for which we don't)? We will use the corrected values, after we removed the outlier (HWS2, MID2, GPA2).

The outlier student had the following values:

```
GPA:            3.13
Midterm:        0.0
HWAvg:          52.59
```

> (i) Calculate, based on the appropriate regression equation, what the midterm score would be predicted to be for this student based on the GPA.

> (ii) Using the regression line for HWS2 vs MID2, what would be the prediction for the midterm score be for this student based on his/her homework average? How well does this compare with the value from (i)?

(iii) Do you have any explanation for the results? (Hint: remember that the student dropped the class.)

In [13]:

```
print("Part I")
print("By applying x = 52.59, the y value, or the midterm score, would be about
 73.3850")

print()

print("Part II")
print("Using the regression line, the x value of 52.59 is about 73 to 74")
print("Compared to the regression equation that we found, the value is not much
 different, but the regression equation gives a specific and more detailed numbe
r because you cannot calculate the exact value by simply looking at the regressi
on line to find the approximate value.")

print()

print("Part III")
print("BY applying x = 52.59 or by approxmiating the value by looking at the reg
ression line, we were able to find out that the expected value for the guy who s
cored 52.59 in homework will receive about 73% in the midterm.")
print("However, the exact value is actually 0 since the student dropped the clas
s.")
print("This shows that even though the student dropped the class, his scores can
be estimated by simply looking at the homework average grades and shows that the
re can exist a huge error by using the linear regression line.")
```

```
Part I
By applying x = 52.59, the y value, or the midterm score, would be a
bout 73.3850

Part II
Using the regression line, the x value of 52.59 is about 73 to 74
Compared to the regression equation that we found, the value is not
much different, but the regression equation gives a specific and mor
e detailed number because you cannot calculate the exact value by si
mply looking at the regression line to find the approximate value.

Part III
BY applying x = 52.59 or by approxmiating the value by looking at th
e regression line, we were able to find out that the expected value
for the guy who scored 52.59 in homework will receive about 73% in t
he midterm.
However, the exact value is actually 0 since the student dropped the
class.
This shows that even though the student dropped the class, his score
s can be estimated by simply looking at the homework average grades
and shows that there can exist a huge error by using the linear regr
ession line.
```

Part C

Answer the following questions in a short sentence or two:

> (i) Did you see any other problems with the necessary conditions for linear regression in the other pairings of data from this set?

> (ii) Let us consider these three pairs of data, after eliminating the outlier but not worrying about any of the other conditions for linear regression. Using the R^2 value, in what order do you put them from weakest to strongest linear relationship? What do you think might be the causes for them being ordered in this way? Is there anything surprising about this? How certain are you of your conclusions, based on the R^2 values?

In [14]:

```
print("Part I")
print("One problem I recognized was that GPA counts for both homework assignment
s and midterm grades and therefore, by simply matching them with pair, exact cor
relation would not be calculated and therefore the correlation with pairing is s
imply not enough.")

print()

print("Part II")
print("We should put the value of GPA and average homework scores as the indepen
dent variables as two x (x1 and x2)")
print("This is because the GPA vs Midterm and HWS vs Midterm showed stronger R s
quared value.")
print("Therefore, by putting HWS and GPA as independent variables as two xs, the
y will have stronger correlation against midterm because they had the strongest
 correlation with each other, which will be combined to maximize the value of r
 squared for homework scores.")
```

```
Part I
One problem I recognized was that GPA counts for both homework assig
nments and midterm grades and therefore, by simply matching them wit
h pair, exact correlation would not be calculated and therefore the
correlation with pairing is simply not enough.

Part II
We should put the value of GPA and average homework scores as the in
dependent variables as two x (x1 and x2)
This is because the GPA vs Midterm and HWS vs Midterm showed stronge
r R squared value.
Therefore, by putting HWS and GPA as independent variables as two x
s, they will have stronger correlation against midterm because they
had the strongest correlation with each other, which will be combine
d to maximize the value of r squared for homework scores.
```

Problem Four Multiple Linear Regression

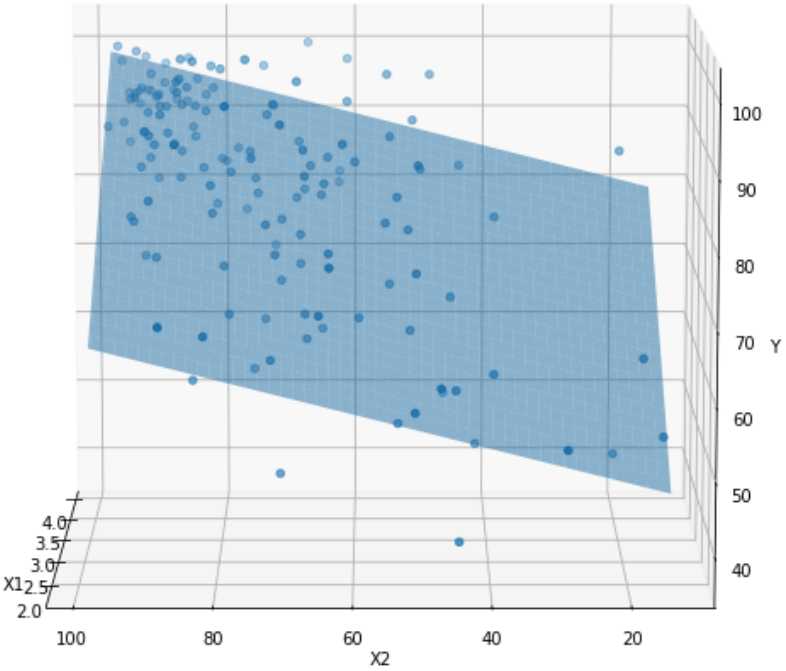
In this problem we will apply the technique of multiple linear regression to some data from a class and predict some values from the regression plane. The next cell contains the code for displaying the regression plane.

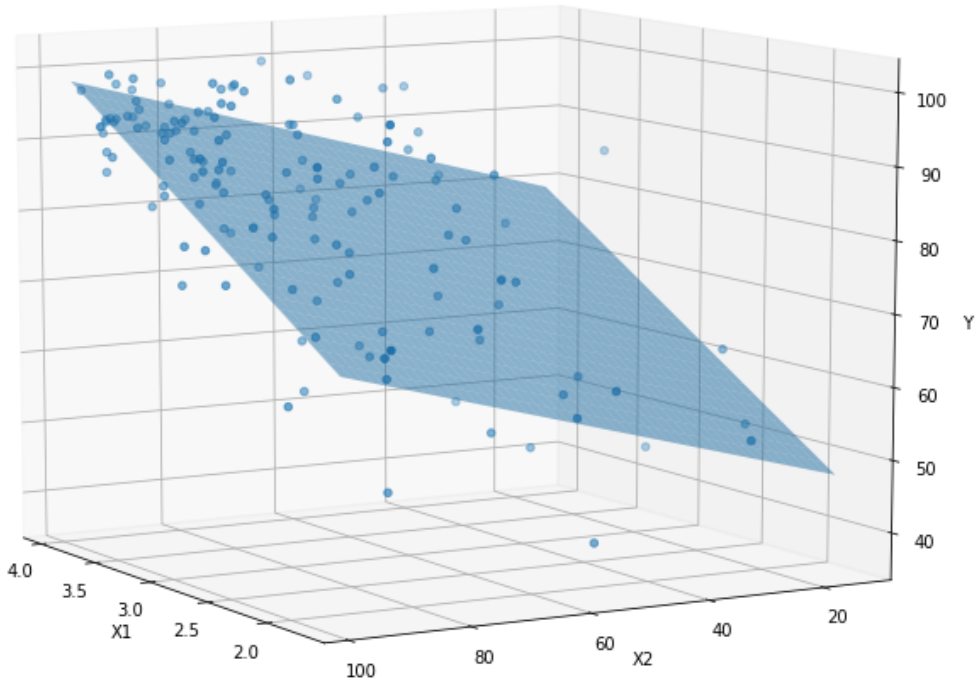
Run this cell and observe how it creates a series of 3D views of the linear regression plane.

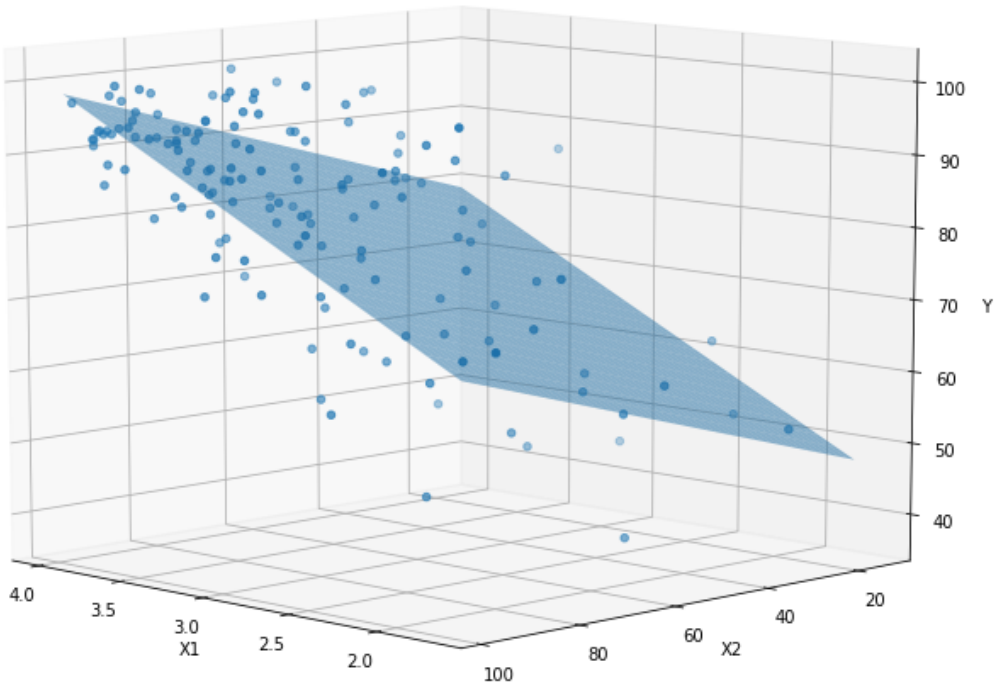
In [15]:

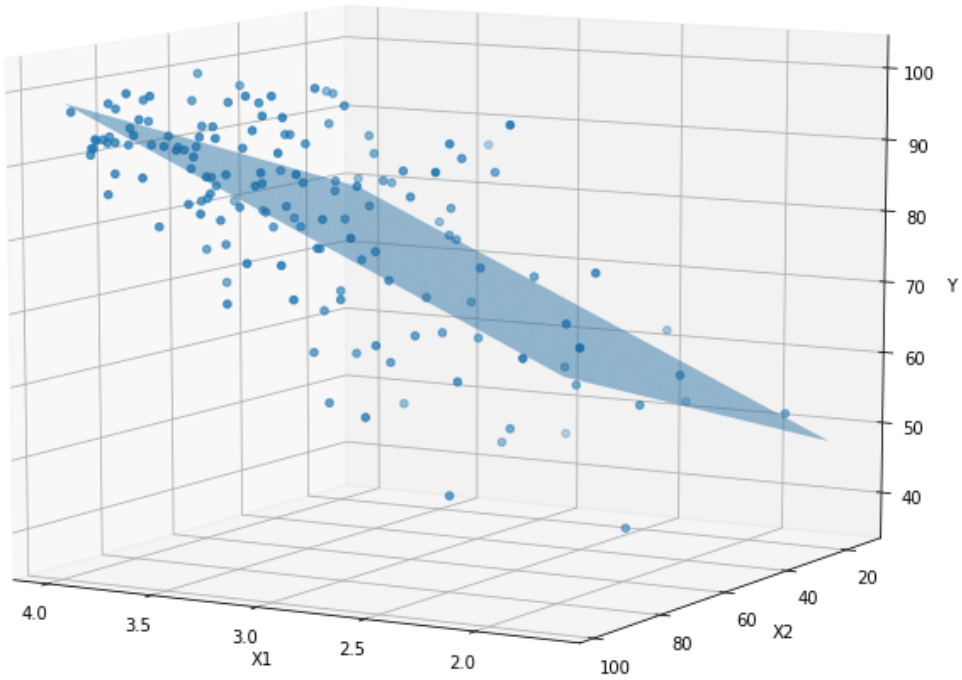
```
# Given data vectors X and Y (must be same length), calculate the vector Theta
# which contains the parameters produced by regression.

def getTheta(X,Y):
    return inv(X.T @ X) @ X.T @ Y

def multiple_regression(X1,X2,Y,verbose=False,el=10,az=180):

    # make sure they are numpy arrays
    X1 = np.array(X1)
    X2 = np.array(X2)
    Y = np.array(Y)

    # Collect the dependent variables and add the bias term 1 to front
    X = transpose([ones(len(X1)),X1,X2])

    Theta = getTheta(X,Y)

    if verbose:
        print("\nRegression Plane:  y = " + str(round4(Theta[0])) + " + "
                                            + str(round4(Theta[1])) + " * x1 + "
                                            + str(round4(Theta[2])) + " * x2")

    # Plot the surface.
    Xplot = np.arange(min(X1), max(X1)+0.1, 0.1)
    Yplot = np.arange(min(X2), max(X2)+0.1, 0.1)
    Xplot, Yplot = np.meshgrid(Xplot, Yplot)
    Zplot = np.zeros_like(Xplot)
    for r in range(len(Zplot)):
        for c in range(len(Zplot[0])):
            Zplot[r][c] = Theta[0] + Theta[1]*Xplot[r][c] + Theta[2]*Yplot[r][c]


    fig = plt.figure(figsize=(12,10))
    ax = fig.gca(projection='3d')
    ax.view_init(elev=el, azim=az)

 #   ax.set_xlim(min(X),)
 #   ax.set_ylim(lo,hi)
 #   ax.set_zlim(lo,hi)
    ax.set_xlabel("X1")
    ax.set_ylabel("X2")
    ax.set_zlabel("Y")
    ax.scatter(X1,X2,Y)
    ax.plot_surface(Xplot,Yplot,Zplot,alpha=0.5)

    plt.show()

X1 = GPA2
X2 = HWS2
Y = MID2


multiple_regression(X1,X2,Y,True,el=10,az=180)
multiple_regression(X1,X2,Y,el=10,az=150)
multiple_regression(X1,X2,Y,el=10,az=135)
multiple_regression(X1,X2,Y,el=10,az=120)
multiple_regression(X1,X2,Y,el=10,az=90)
```

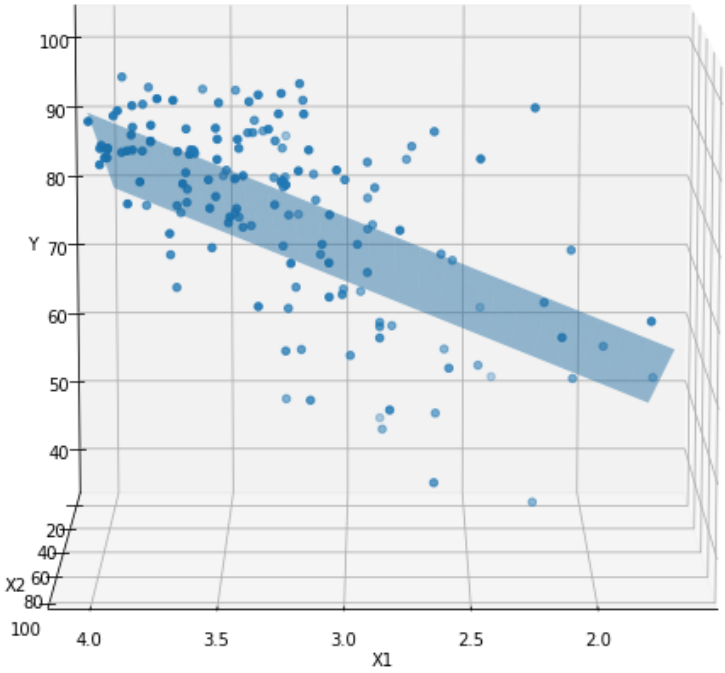
```
Regression Plane:  y = 21.2987 + 13.5759 * x1 + 0.2323 * x2
```




100
90
80
70
Y
60
50
40
4.0
3.5
3.0
2.5
2.0
X1
100
80
60
40
20
X2


100
90
80
70
Y
60
50
40
4.0
3.5
3.0
X1
2.5
2.0
100
80
60
X2
40
20


100
90
80
70
Y
60
50
40
4.0
3.5
3.0
X1
2.5
2.0
100
80
60
X2
40
20


100
90
80
Y 70
60
50
40
20
40
X2 60
80
100
4.0
3.5
3.0
2.5
2.0
X1

Part A

Now we will determine which pair of data best predict the other two. Using the formula for ESS, TSS, and R^2:

$$ESS = \sum_{i=0}^{N} (\hat{y}_i - \mu_Y)^2$$

$$TSS = \sum_{i=0}^{N} (y_i - \mu_Y)^2$$

$$R^2 = \frac{ESS}{TSS}$$

and the regression equation determined for this data set:

$$\hat{Y} = 21.2987 + 13.5759 \cdot X_1 + 0.2323 \cdot X_2$$

calculate the R^2 value for

```
X1 = GPA2
X2 = HWS2
Y = MID2
```

In [16]:

```
def rsquared(X1,X2,Y):
    n = len(X1)


    mux = np.mean(X1)
    muy = np.mean(X2)
    muz = np.mean(Y)
    stdx = np.std(X1)
    stdy = np.std(X2)
    stdz = np.std(Y)
    X = transpose([ones(len(X1)),X1,X2])
    Theta = getTheta(X,Y)
    one = round4(Theta[0])
    two = round4(Theta[1])
    three = round4(Theta[2])



    Yhat = [(one + two*X1[i]+ three * X2[i]) for i in range(n)]



    # explained sum of squares -- deviation of line from mean of y
    ess = sum( [ (Yhat[i] - muz)**2 for i in range(n)])

    # total sum of squares -- deviation of data from mean of y
    tss = sum( [ (Y[i] - muz)**2 for i in range(n)] )
    return round4(ess/tss)

value = rsquared(GPA2,HWS2,MID2)
print(value)
```

```
0.4934
```

Part B

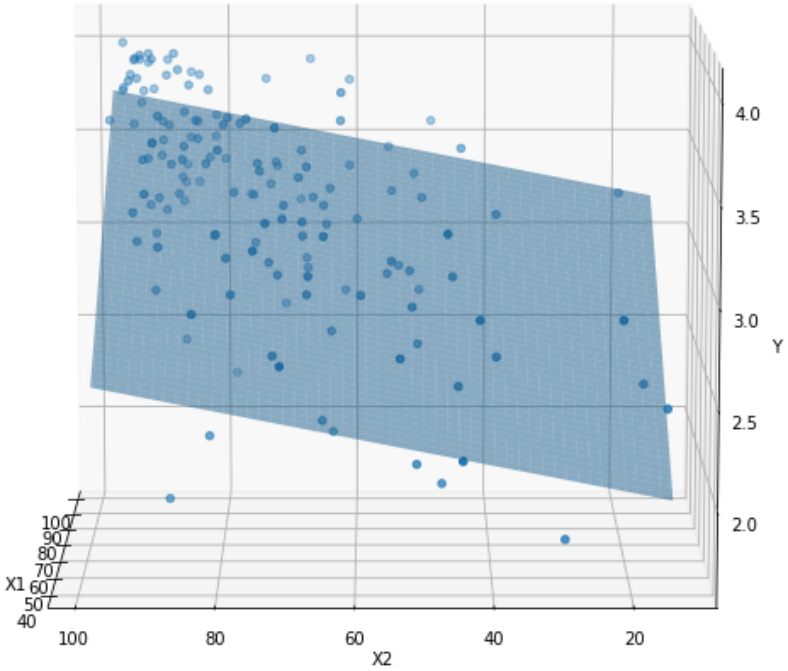
Now consider the other 2 possibilities for two of these data predicting the other one (note that you don't need to consider reordering X1 and X2), print out your results, and then answer the following question: Which had the highest correlation and was it relevant?

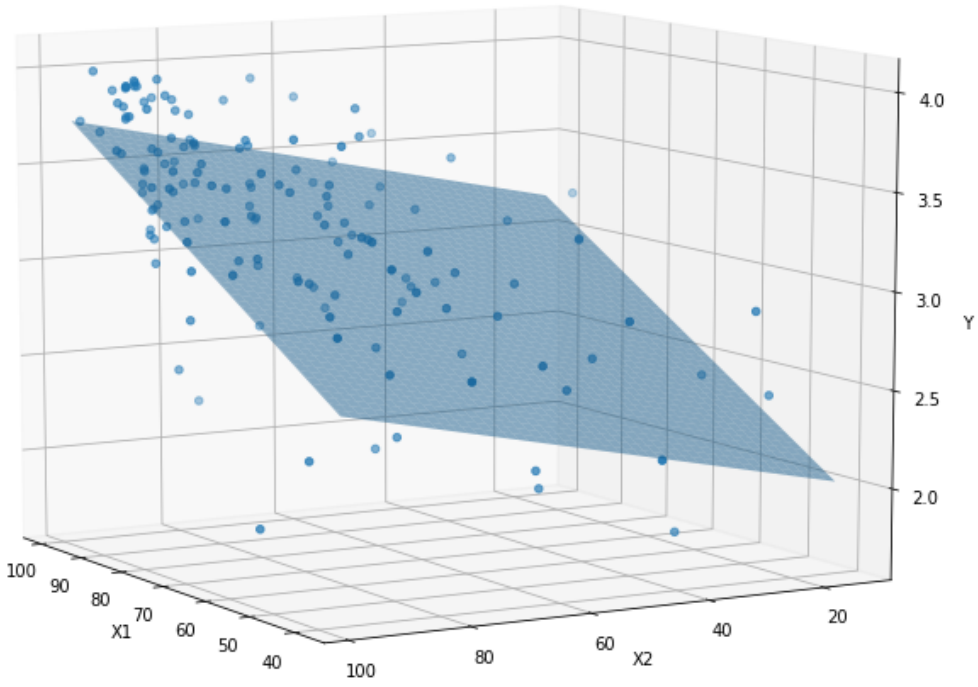
Hint: You will need to recalculate the regression line, look at how it is done in the function `multiple_regression(...)`.

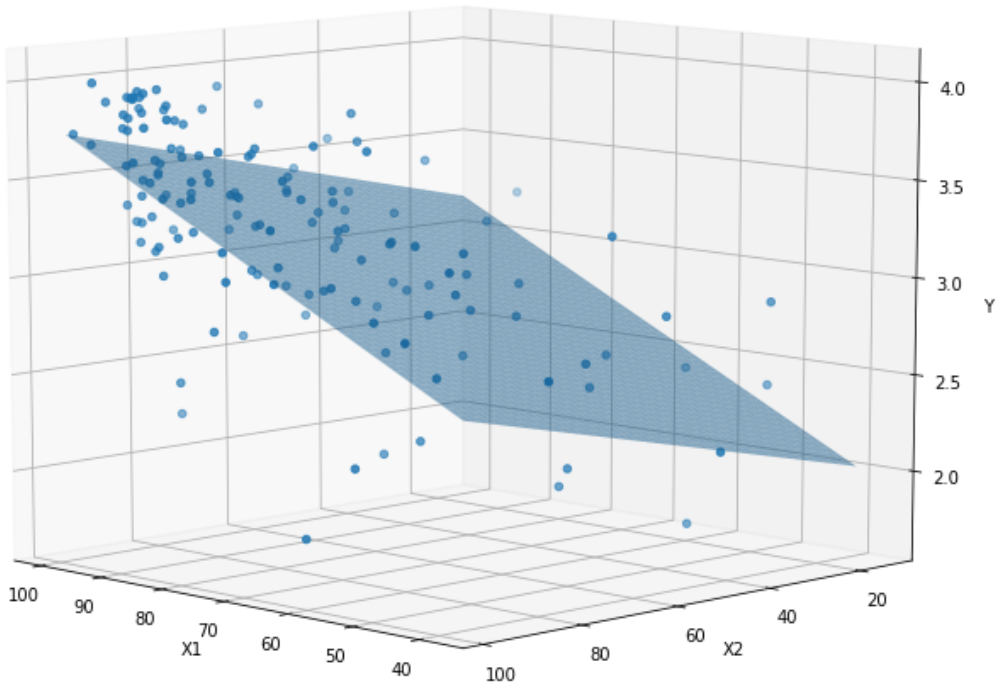
In [17]:

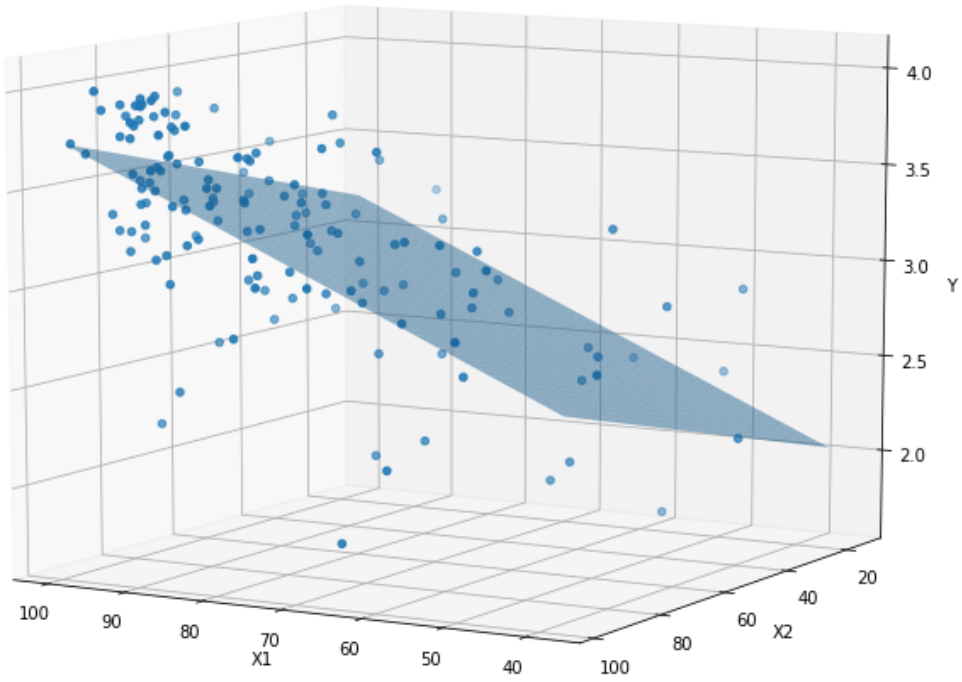
```
X1 = MID2
X2 = HWS2
Y = GPA2


multiple_regression(X1,X2,Y,True,el=10,az=180)
multiple_regression(X1,X2,Y,el=10,az=150)
multiple_regression(X1,X2,Y,el=10,az=135)
multiple_regression(X1,X2,Y,el=10,az=120)
multiple_regression(X1,X2,Y,el=10,az=90)

value = rsquared(MID2,HWS2,GPA2)
print(value)

print()


X1 = GPA2
X2 = MID2
Y = HWS2


multiple_regression(X1,X2,Y,True,el=10,az=180)
multiple_regression(X1,X2,Y,el=10,az=150)
multiple_regression(X1,X2,Y,el=10,az=135)
multiple_regression(X1,X2,Y,el=10,az=120)
multiple_regression(X1,X2,Y,el=10,az=90)

value = rsquared(GPA2,MID2,HWS2)
print(value)
```

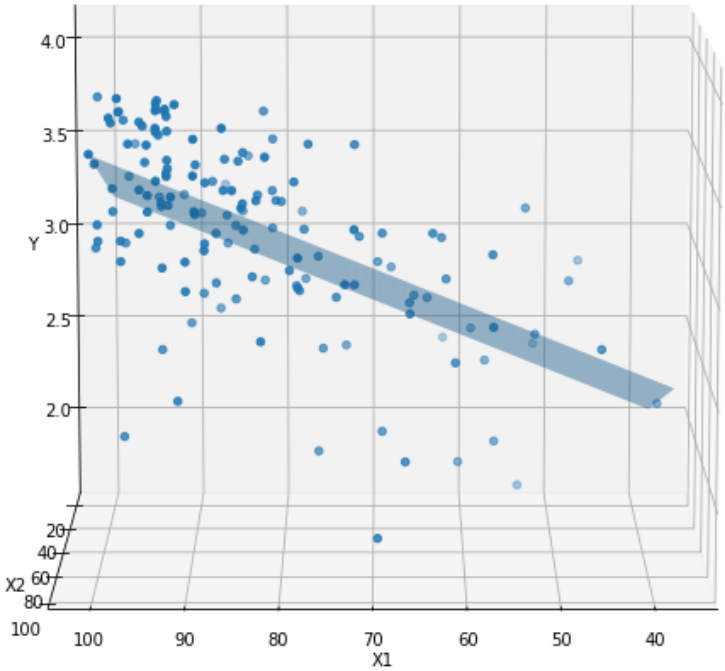
```
Regression Plane:  y = 1.227 + 0.0185 * x1 + 0.0067 * x2
```




4.0
3.5
3.0
2.5
2.0
Y
100
90
80
70
60
50
40
X1
100
80
60
40
20
X2

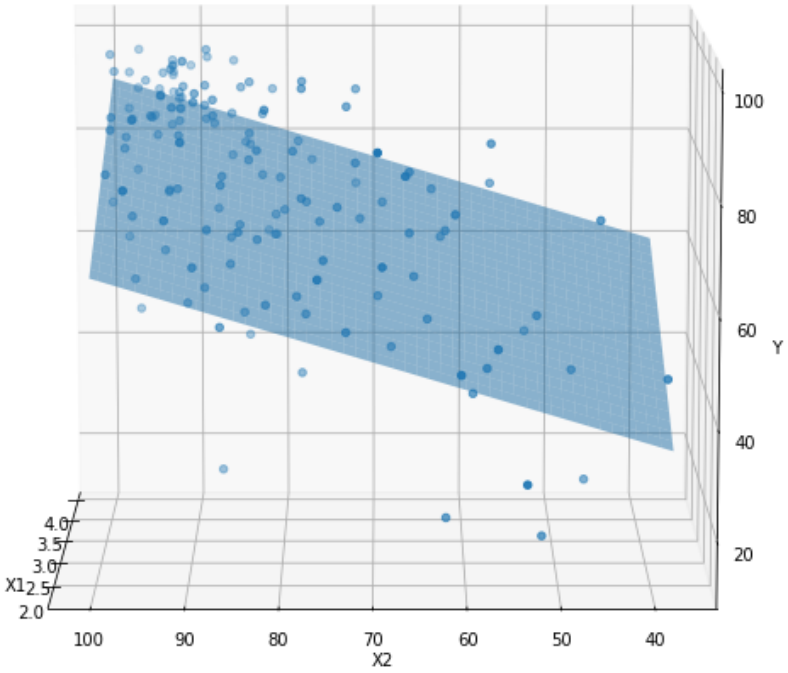

4.0
3.5
3.0
2.5
2.0
Y
100
90
80
70
60
50
40
X1
100
80
60
40
20
X2

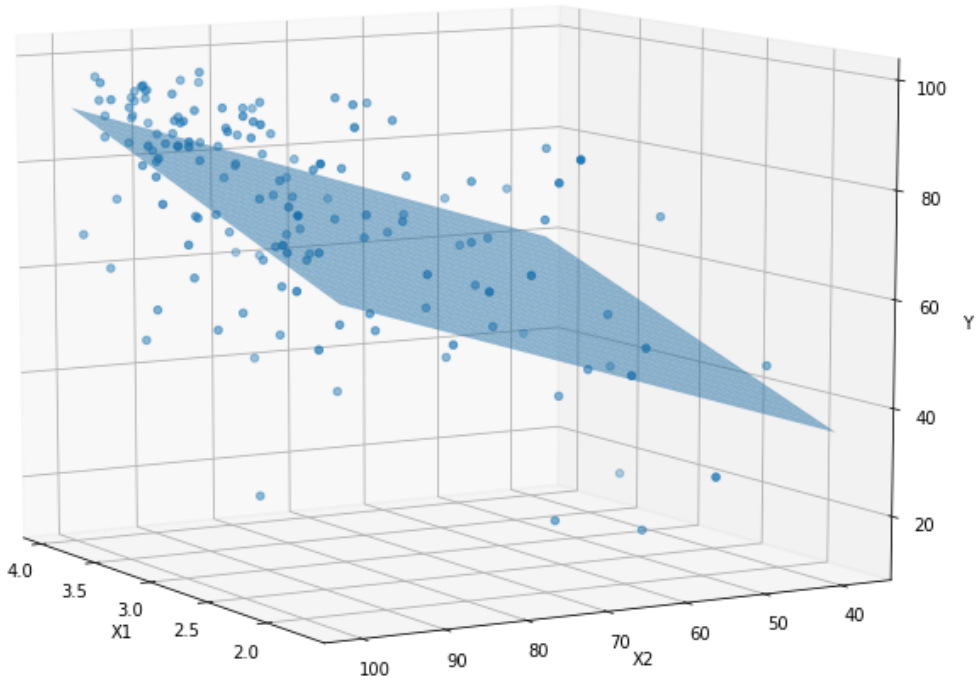

4.0
3.5
3.0
2.5
2.0
Y
100
90
80
70
60
50
40
X1
100
80
60
40
20
X2

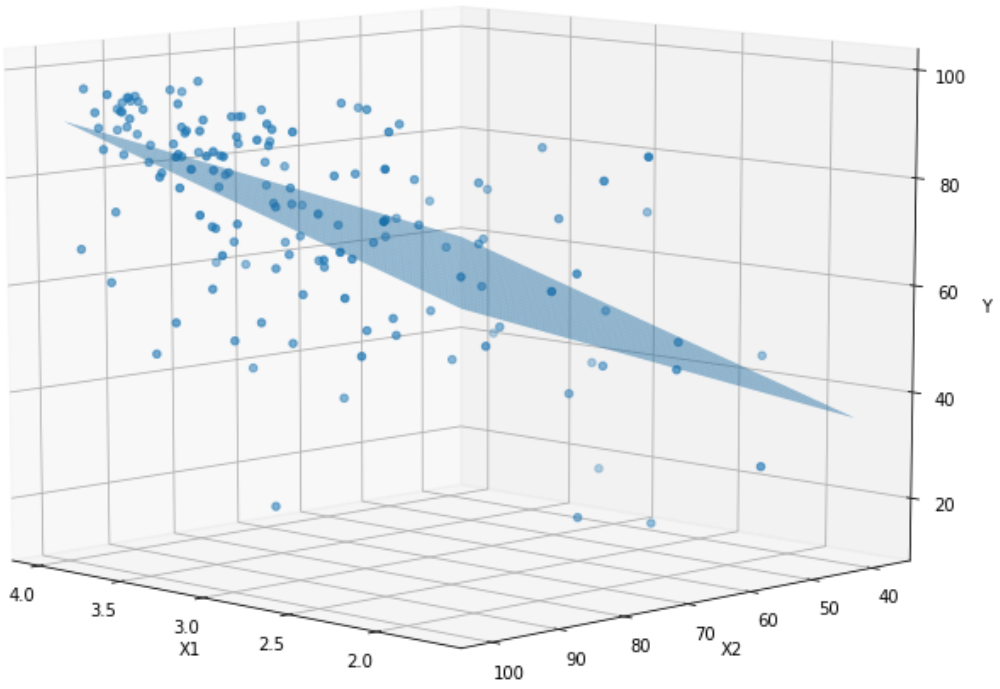


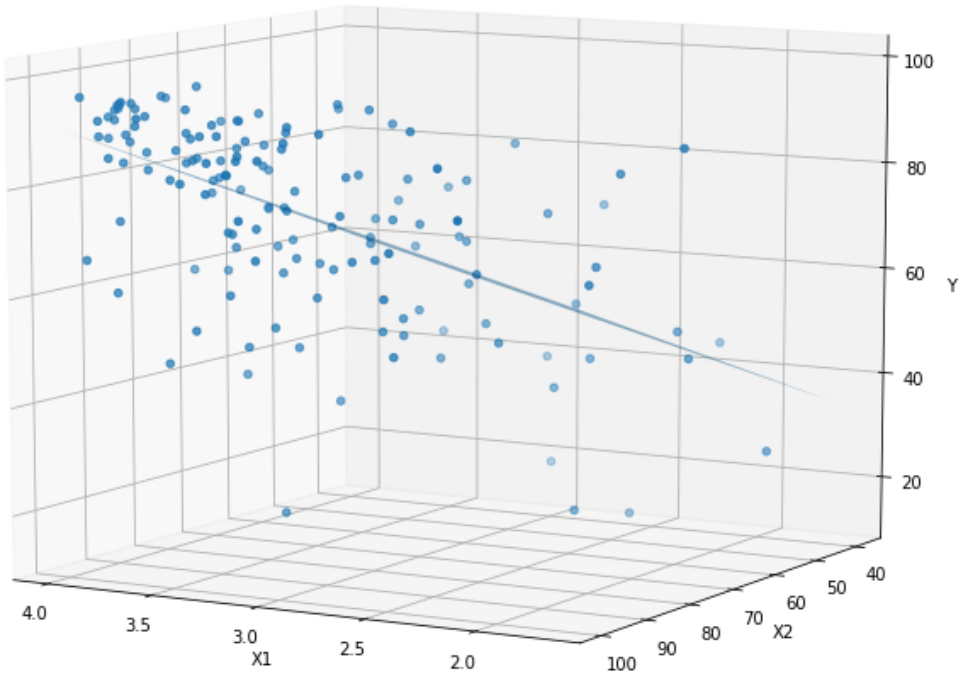

```
0.4674
```

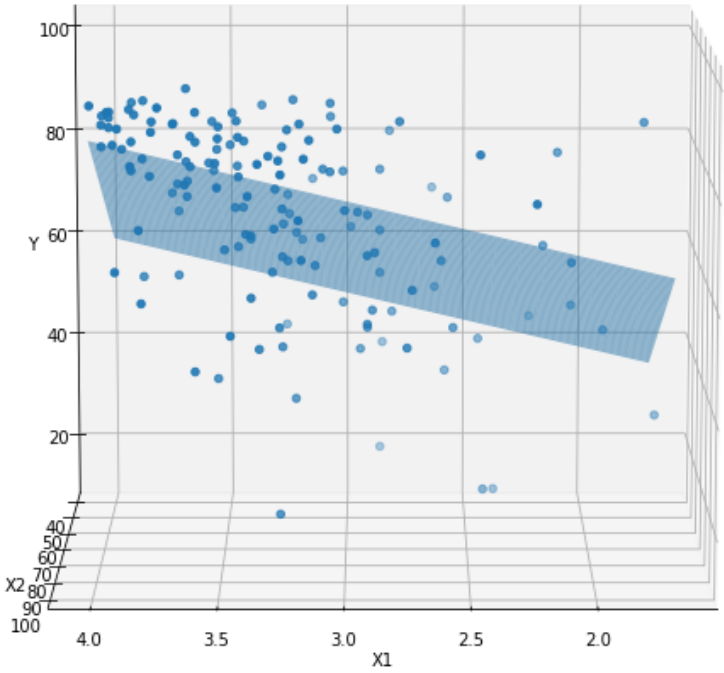
```
Regression Plane:  y = -1.8688 + 10.6081 * x1 + 0.4996 * x2
```


100
80
60
Y
40
20
4.0
3.5
3.0
X1
2.5
2.0
100
90
80
70
60
50
40
X2


100
80
60
40
20
Y
4.0
3.5
3.0
2.5
2.0
X1
100
90
80
70
60
50
40
X2


100
80
60
40
20
Y
4.0
3.5
3.0
2.5
2.0
X1
100
90
80
70
60
50
40
X2


100
80
60
40
20
Y
4.0
3.5
3.0
2.5
2.0
X1
100
90
80
70
60
50
40
X2




```
0.3717
```

Part C

Now answer the following questions:

(i) Which produced a better linear model?

(ii) Were the R^2 values high enough to reasonably use this as a predictor of student performance?

In [18]:

```
print("Part I")
print("Putting Midterm as the Y and GPA and average homework scores as X (x1 and
x2) produced the better linear model")

print()

print("Part II")
print("The R squared value of 0.4934, which was obtained for putting midterm as
 Y, is not a great value for r squared but is reasonable to use as the predicto
r")
print("Similarly, putting GPA as Y produced the R squared value of 0.4674, which
is reasonably high.")
print("However, putting homework scores as Y produced the R squared vlaue of 0.3
717, which is not high enough to ensure the linear model between the variables."
)
```

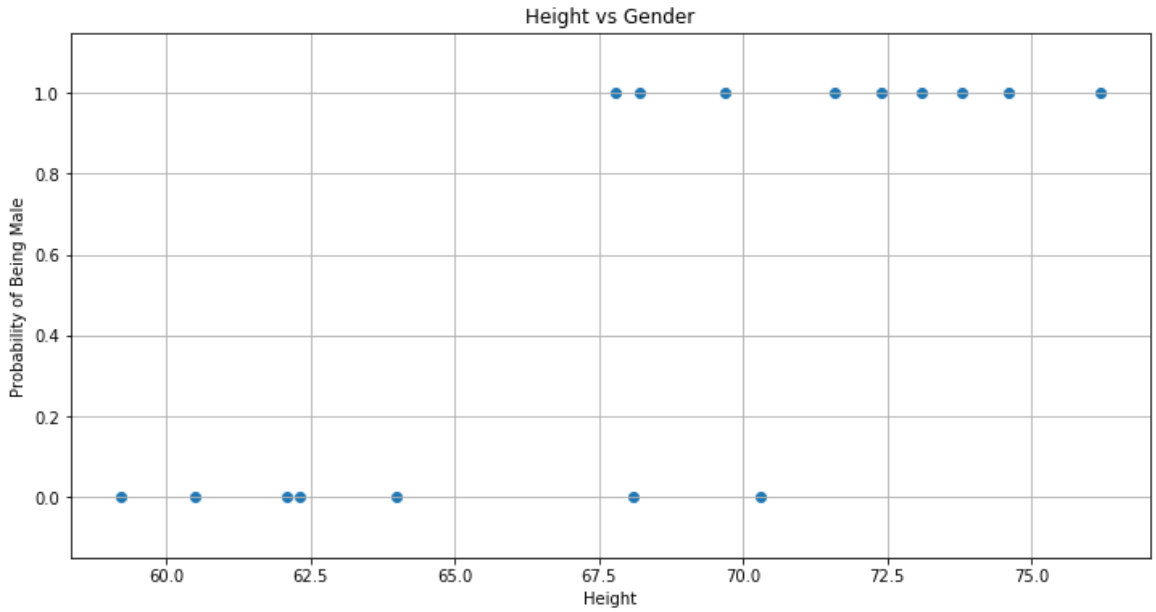
```
Part I
Putting Midterm as the Y and GPA and average homework scores as X (x
1 and x2) produced the better linear model

Part II
The R squared value of 0.4934, which was obtained for putting midter
m as Y, is not a great value for r squared but is reasonable to use
as the predictor
Similarly, putting GPA as Y produced the R squared value of 0.4674,
which is reasonably high.
However, putting homework scores as Y produced the R squared vlaue o
f 0.3717, which is not high enough to ensure the linear model betwee
n the variables.
```

Problem Five: Logistic Regression

This problem will review the process of logistic regression, using an example similar to what was shown in class.

In [19]:

```
Height = [59.2, 60.5, 62.1, 62.3, 73.8, 64.0, 71.6, 67.8,68.1,68.2,69.7,70.3, 7
2.4,73.1,74.6,76.2]

Gender = [0,0,0,0,1,0,1,1,0,1,1,0,1,1,1,1]

plt.figure(figsize=(12,6))
plt.grid()
plt.title("Height vs Gender")
plt.ylim([-0.15,1.15])
plt.xlabel("Height")
plt.ylabel("Probability of Being Male")
plt.scatter(Height,Gender)
plt.show()
```



In [20]:

```
def s(z):
    return 1/(1+np.exp(-z))

def Cost(y,yHat):
    if y == 1:
        return -log(yHat)
    else:
        return -log(1 - yHat)

def h(b,m,xi):
    return b + m*xi

def J(b,m,X,Y):
    N = len(X)
    res = 0
    for k in range(len(X)):
        res += Cost(Y[k],s(h(b,m,X[k])))

    return res/N

def update_weights(b,m, X, Y, blam,mlam,verbose=False):
    m_deriv = 0
    b_deriv = 0
    N = len(X)
    for i in range(N):
        # Calculate partial derivatives
        m_deriv += X[i] * (s(h(b,m,X[i])) - Y[i])
        b_deriv += (s(h(b,m,X[i])) - Y[i])
    m_deriv /= N
    b_deriv /= N
    # We subtract because the derivatives point in direction of steepest ascent
    m -= m_deriv * mlam
    b -= b_deriv * blam

    return b,m,b_deriv,m_deriv

def Bderiv(b,m,X,Y):
    (b,m,bd,md) = update_weights(b,m,X,Y,0.001,0.001)
    return bd

def Mderiv(b,m,X,Y):
    (b,m,bd,md) = update_weights(b,m,X,Y,0.001,0.001)
    return md

def gradient_descent(b,m,X,Y,blam,mlam,limit,verbose=False):
    blast = b
    mlast = m
    inc = int(limit/10)
    for k in range(limit):
        (b,m,bd,md) = update_weights(blast,mlast,X,Y,blam,mlam,verbose)
        if verbose and (k % inc == 0):
            #print("  b_deriv = ",bd," \tm_deriv = ",md)
            #print("  b_delta = ",(b-blast),"\tm_delta = ",(m-mlast))
            print("b = ",b,"\tm = ",m, "J = ", J(b,m,X,Y))
        blast = b
        mlast = m
    return (b,m)
```

Part A

Fill in the following code template, which returns the probability that a person is male, given the height and the parameters m and b as input. Hint: It will involve h and s.

In [21]:

```
# Given an x-axis value for height, what is the prediction that is male?
def predict(b,m,x):
    value = x * m + b
    return 1/(1+math.exp(value))
```

Part B

Now run the gradient descent algorithm to find the best values for m and b. Experiment with different values for limit, mlam, and blam. The code will print out the results ten times during the calculation.

You should be able to get values close to $m = \frac{2}{3}$ and $b = -45$.

In [22]:

```
limit = 10**5          # Try as large as you can on your machine!

mlam = 0.006
blam = 0.021

b = 0
m = 1

verbose = True
print("\nb = ",b,"\t\t\tm = ",m)
(b1,m1) = gradient_descent(b,m,Height,Gender,blam,mlam,limit,verbose)
print("\nb = ",b1,"\t\tm = ",m1)
```

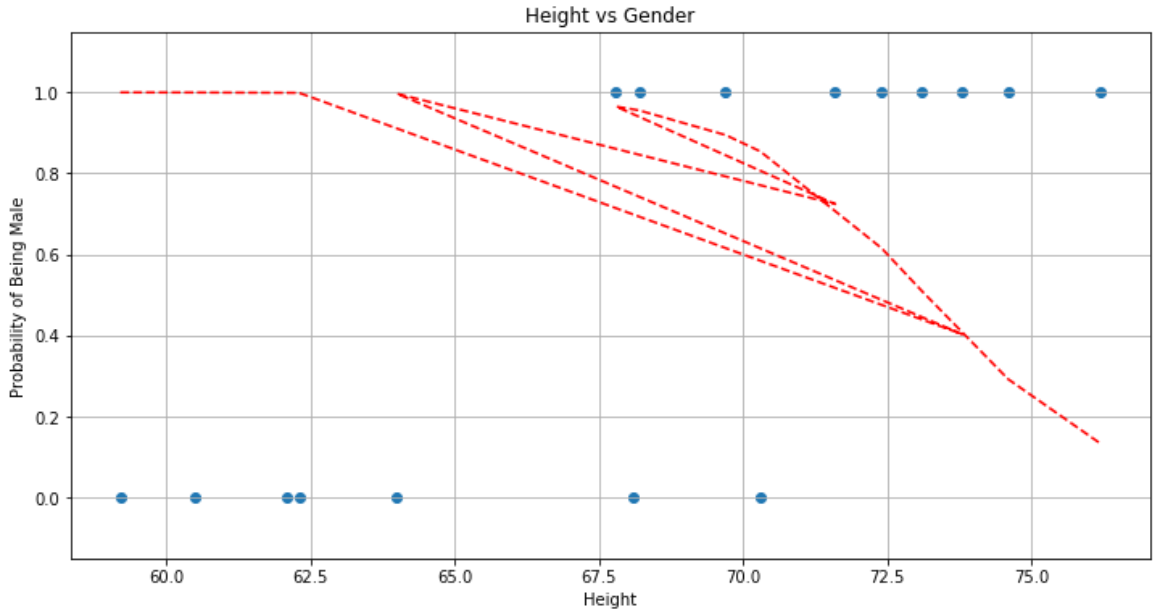
```
b =  0                    m =  1
b =  -0.009187500000000001      m =  0.8325625 J =  inf

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:8: Run
timeWarning: divide by zero encountered in log


b =  -5.958652471122045         m =  0.047571331071761924 J =  1.493
2743995104638
b =  -11.322201123569542        m =  0.16847288472522043 J =  0.4372
63433309852
b =  -16.18814400291426         m =  0.4069242794813842 J =  4.27347
4314805299
b =  -21.000226270298015        m =  0.3738019853010527 J =  1.29204
61092699893
b =  -25.298070834704575        m =  0.4138720375014291 J =  0.70139
86655883032
b =  -29.3830493700621  m =  0.4617339703252553 J =  0.4769354558203
461
b =  -33.33967480746126         m =  0.5313079328846753 J =  0.60612
65049701585
b =  -37.26823639271033         m =  0.611822666508911 J =  0.963711
720350458
b =  -41.29504363552937         m =  0.7079443171133911 J =  1.73789
14104135605

b =  -45.157501165191704                m =  0.6172046574270488
```

Part C

Now complete this template by adding code to show the prediction curve calculated by the algorithm. (Hint: this is just the function `predict(...)` .)

In [23]:

```
plt.figure(figsize=(12,6))
plt.grid()
plt.title("Height vs Gender")
plt.ylim([-0.15,1.15])
plt.scatter(Height,Gender)
plt.xlabel("Height")
plt.ylabel("Probability of Being Male")

x = []
for y in Height:
    x.append(predict(b1,m1,y))
plt.plot(Height,x,'r--',label="Trendline")
plt.show()
```



Part D

Now answer the following two questions:

(i) Your professor is 70 inches tall. What is the probability that he is male, according to this algorithm?

(ii) The decision rule associated with this algorithm is that an input height is classified as male if the probability is 0.5 or higher. Does this algorithm classify your gender correctly?

In [24]:

```
print("Part I")
print("If the professor is male, he is male according to this algorithm since th
e probability is higher than 0.5")

print()

print("Part II")
print("This algorithm does not classify the gender perfectly even though it clas
sifies most of them because there exists some outliers or some males that are ty
pically smaller than usual males and female that are taller than usual females."
)
print("However, most of the data points are classified correctly.")
```

```
Part I
If the professor is male, he is male according to this algorithm sin
ce the probability is higher than 0.5

Part II
This algorithm does not classify the gender perfectly even though it
classifies most of them because there exists some outliers or some m
ales that are typically smaller than usual males and female that are
taller than usual females.
However, most of the data points are classified correctly.
```