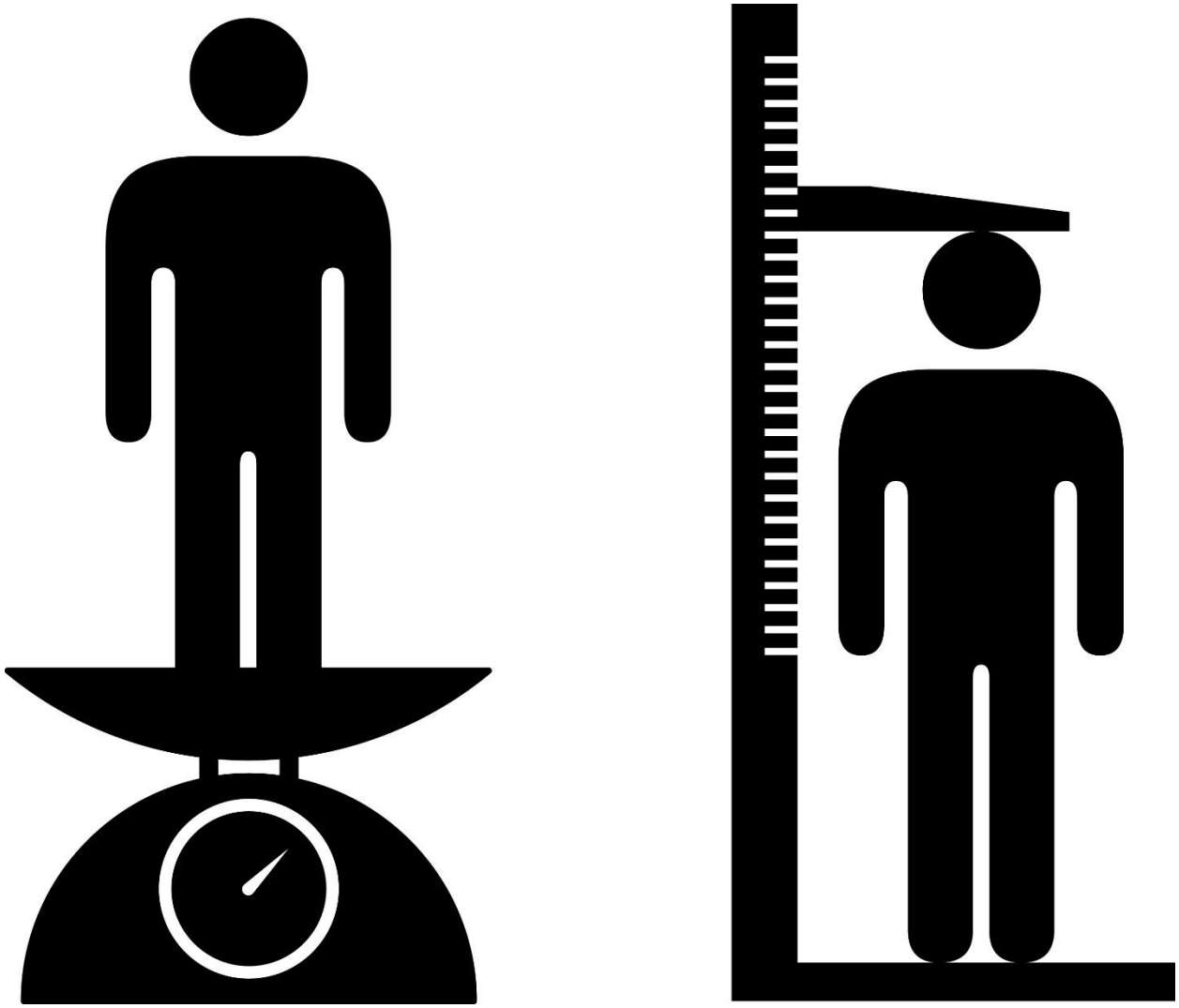
MODEL FOR PREDICTING WIEGHT USING HEIGHT



INTRODUCTION

According to most of the researchers, weight and height have a very deep connection when it comes to real life evaluation. Thus, here we make an effort to create a model which can predict weight by giving height as an input. Mostly you see that on average, men have more height than women. This theory can differ from region to region, country to country and state to state. Also, there is challenge to predict weight using height when there is no factor / feature of age, work, region etc. Thus, here we are focusing on pure statistical modeling and insights to create a relation between these two values (Height and Weight). Also, we can think in another perspective, such as what would be the gender of the candidate if his/her weight and height are given as input. Should we use some different model on that? So, what we try to achieve here is to implement multiple machine learning models on same dataset in order to create multiple insights in form of prediction which can be display in form of values or graphs.

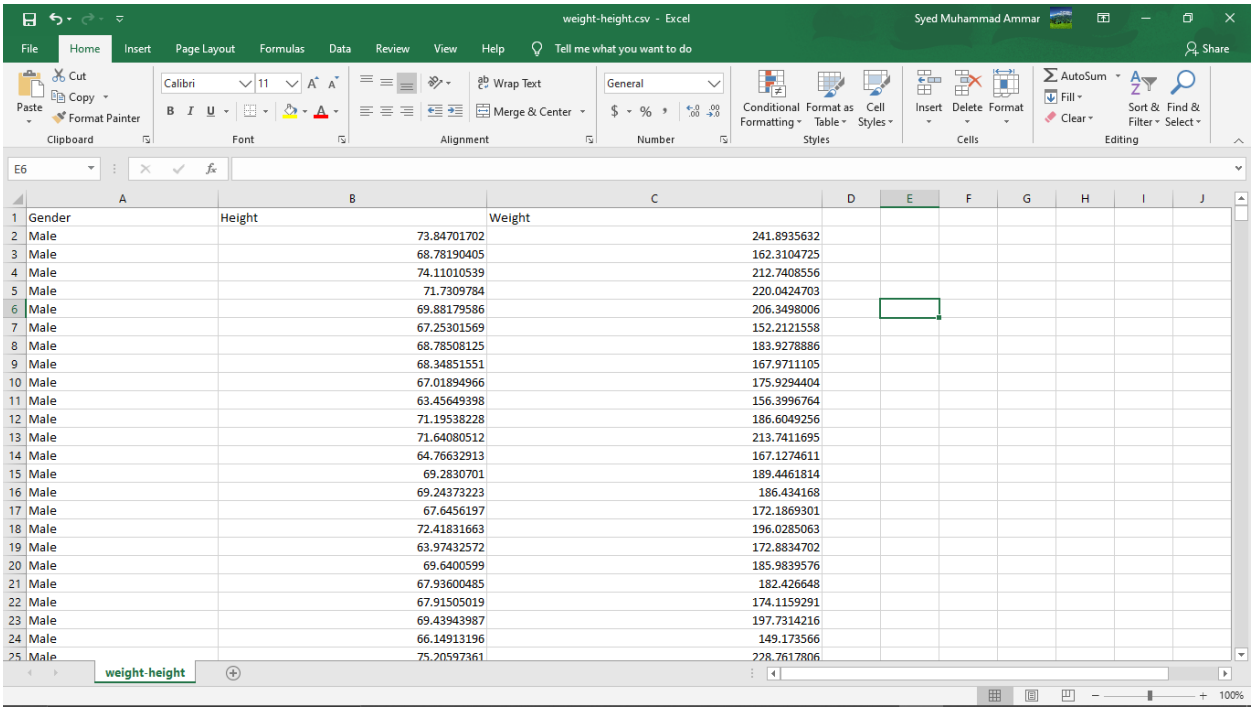
PROBLEM STATEMENT

Following are some main insights which we need to discover from the given dataset.

- What will be the weight of the candidate given height as input only?
- What will be the weight of candidate given gender and height of candidate as input?
- Is it possible to get gender of the candidate if weight and height is given as input?
- How we create connection between all these features?

DATASET

The data is collected online. It's a csv file with the size of 419 kb which holds total 10K records.



	Gender	Height	Weight
2	Male	73.84701702	241.8935632
3	Male	68.78190405	162.3104725
4	Male	74.11010539	212.7408556
5	Male	71.7309784	220.0424703
6	Male	69.88179586	206.3498006
7	Male	67.25301569	152.2121558
8	Male	68.78508125	183.9278886
9	Male	68.34851551	167.9711105
10	Male	67.01894966	175.9294404
11	Male	63.45649398	156.3996764
12	Male	71.19538228	186.6049256
13	Male	71.64080512	213.7411695
14	Male	64.76632913	167.1274611
15	Male	69.2830701	189.4461814
16	Male	69.24373223	186.434168
17	Male	67.6456197	172.1869301
18	Male	72.41831663	196.0285063
19	Male	63.97432572	172.8834702
20	Male	69.6400599	185.9839576
21	Male	67.93600485	182.426648
22	Male	67.91505019	174.1159291
23	Male	69.43943987	197.7314216
24	Male	66.14913196	149.173566
25	Male	75.20597361	228.7617806

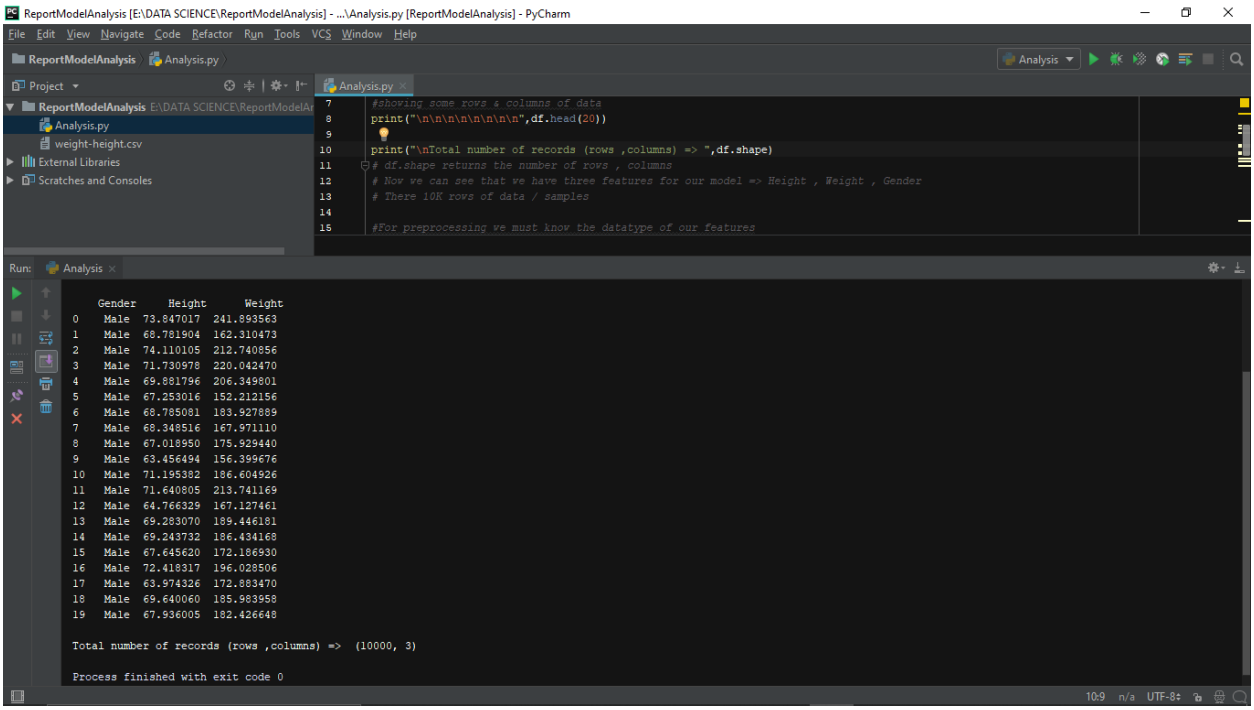
As you can see, three features are given

- Gender
- Weight (Pounds)
- Height (Inches)

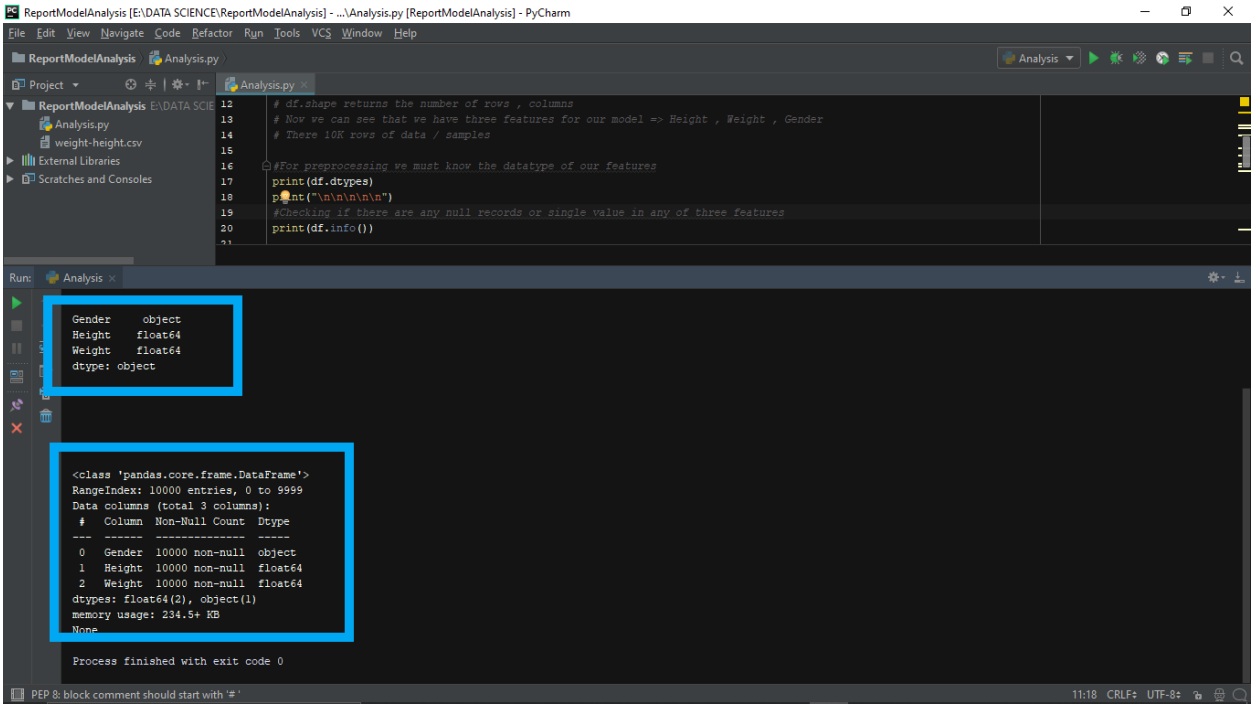
Now, to predict weight, we will make weight as our target class. Now what will be our input? Should we only go for height or both (height and gender). To know which model will suit better, we have to experiment both scenarios.

Also, we need to see if data need to clean or not. For this, we will evaluate data using **pandas** library in order to preprocess our data in a right and effective manner.

Here we import our data into data-frame using pandas library



To preprocess our data effectively, we must know the data types of every feature and we must check if there is any missing (Null) value in dataset.

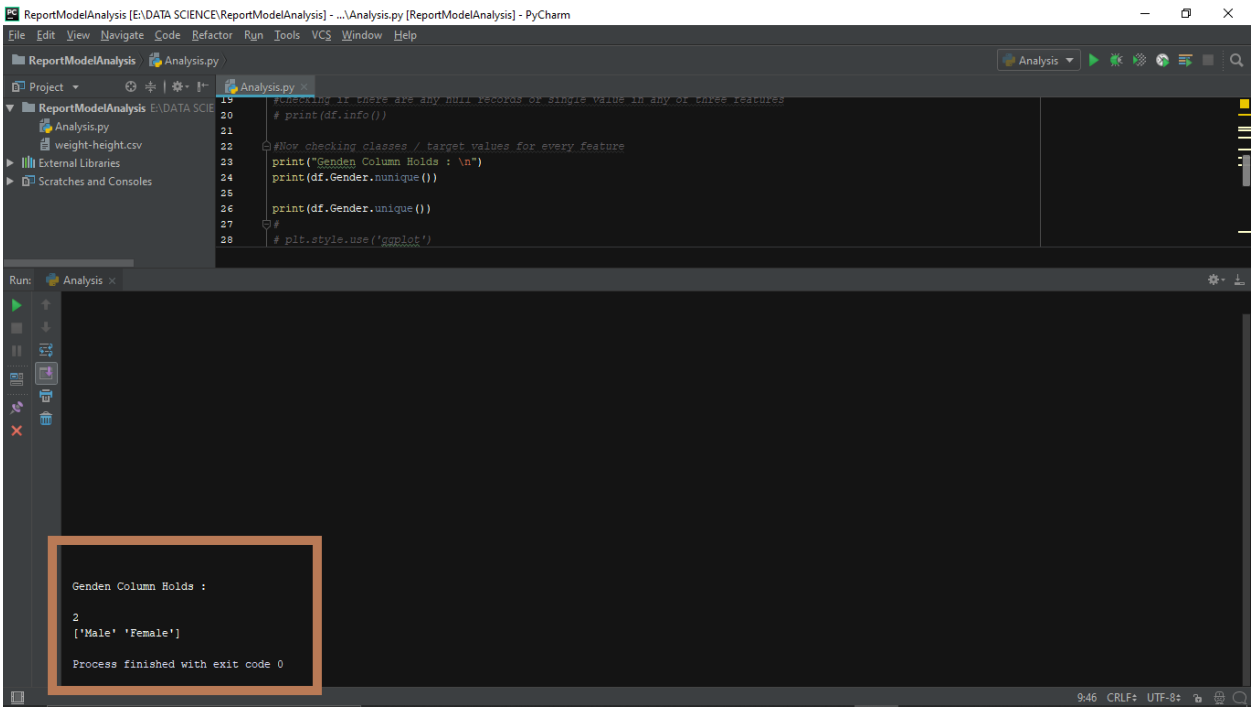


As no null values are present among all three features, so we are in good position to go ahead. So, now its time to study the features.

1. **Categorical Feature => Gender**

2. **Non-Categorical Features => Height, Weight**

Let's talk about gender. Its important that we should now how many values in terms of uniqueness, our feature holds.



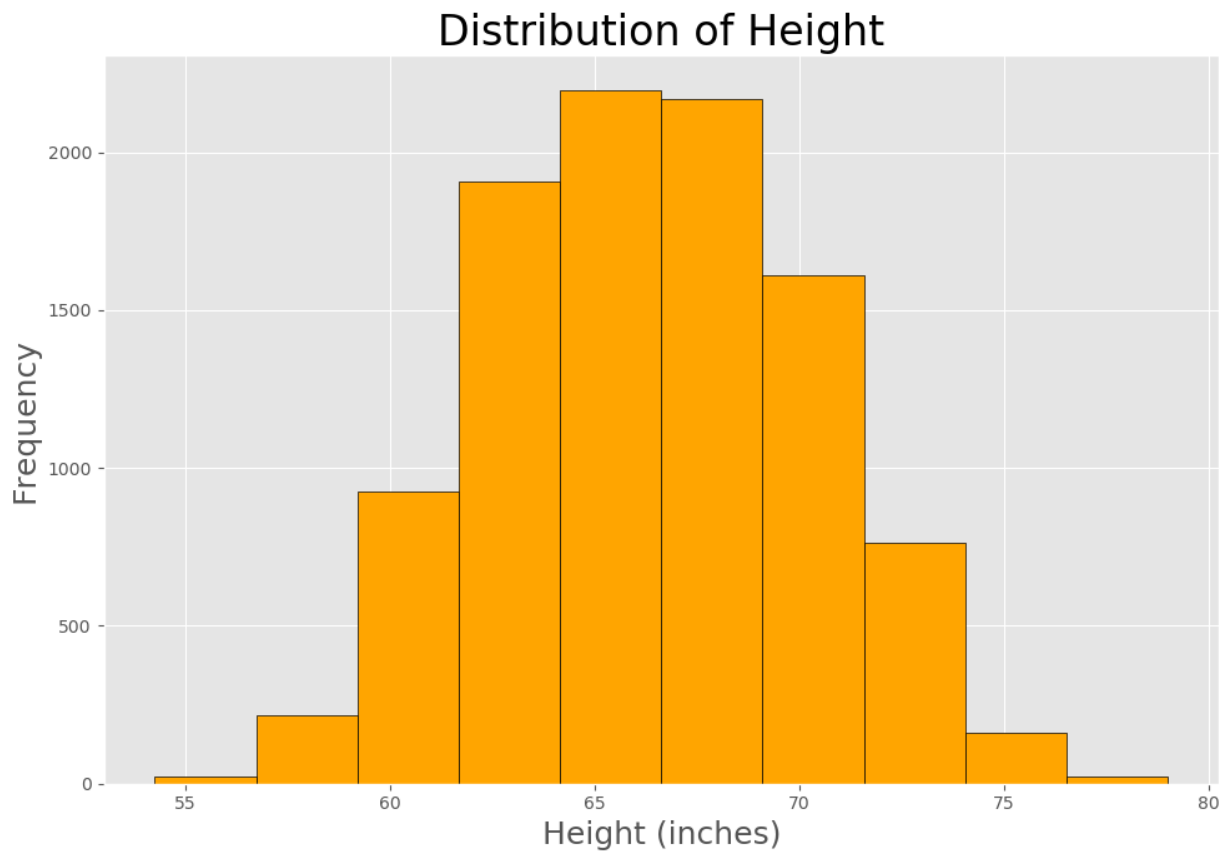
So, we see that our gender column holds two values:

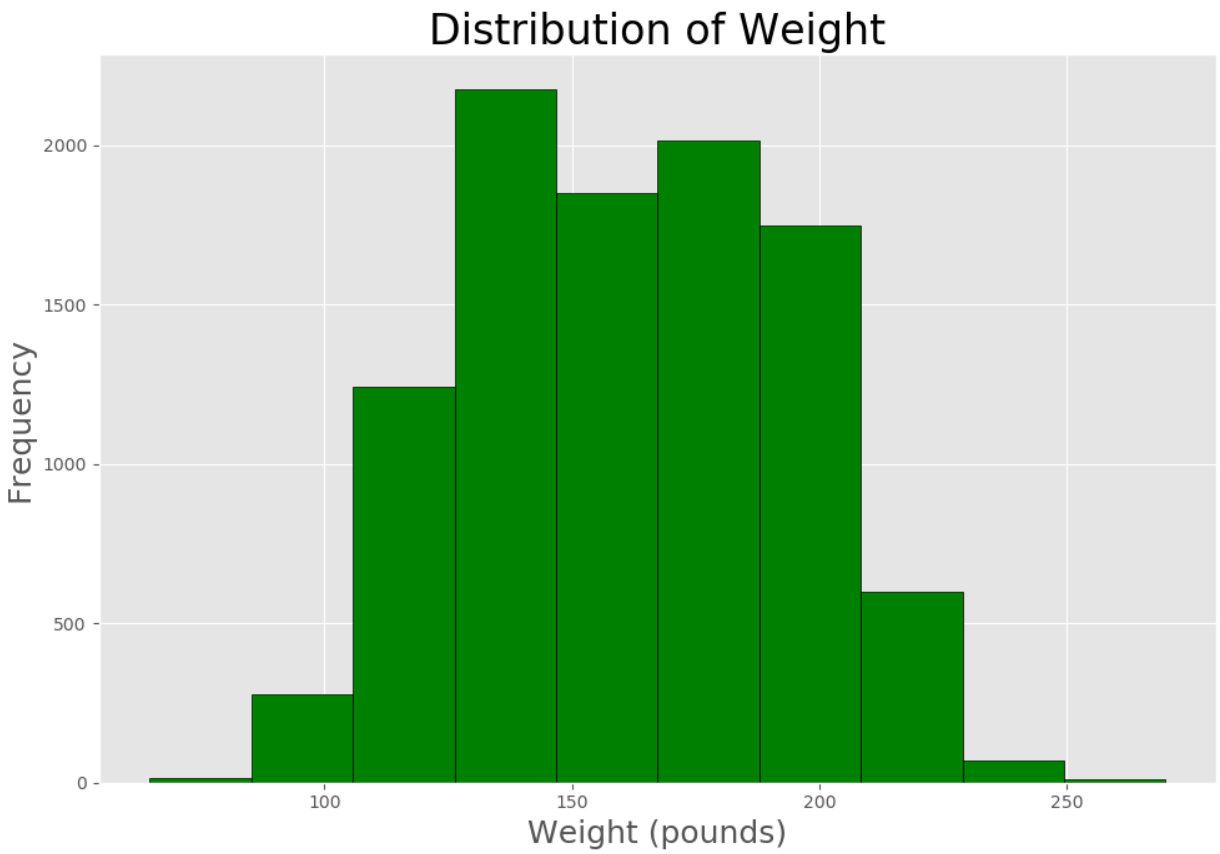
- Male
- Female

Now we must study the pattern of height and weight with respect to the gender. To better understand the distribution of the variables Height and Weight, we can

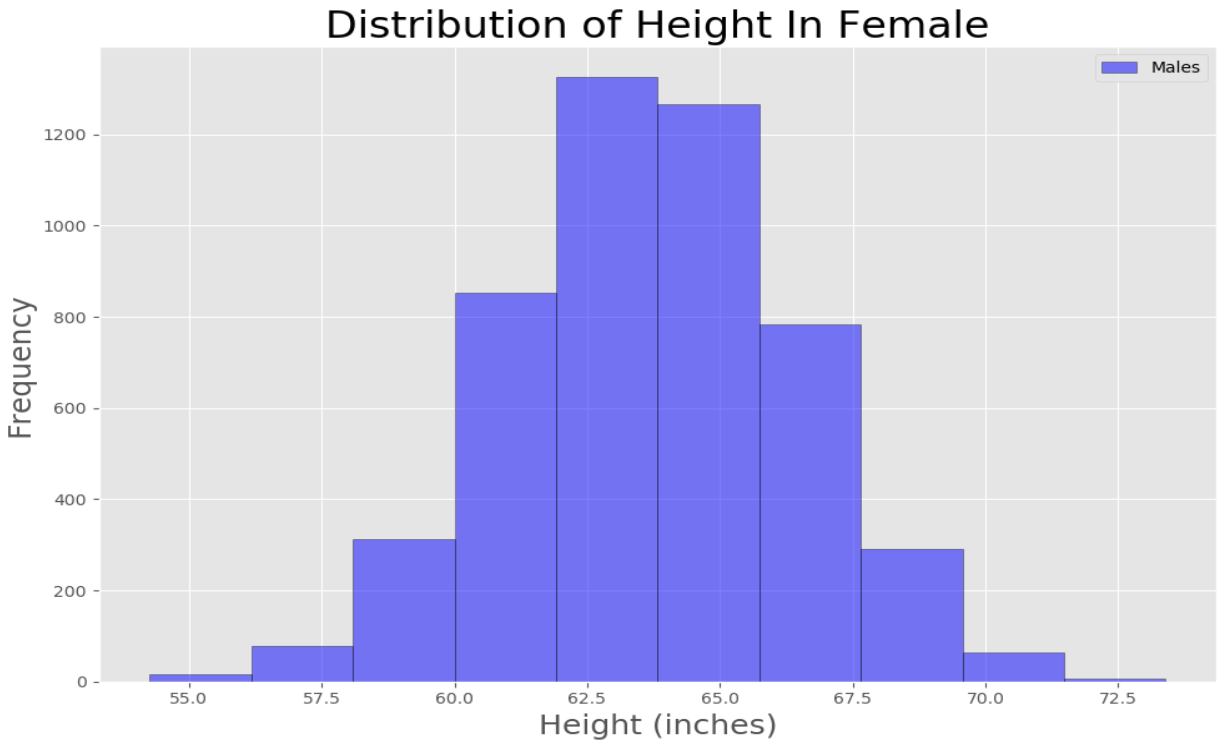
simply plot both variables using histograms. Histograms are plots that show the distribution of a numeric variable, grouping data into bins. The height of the bar represents the number of observations per bin.

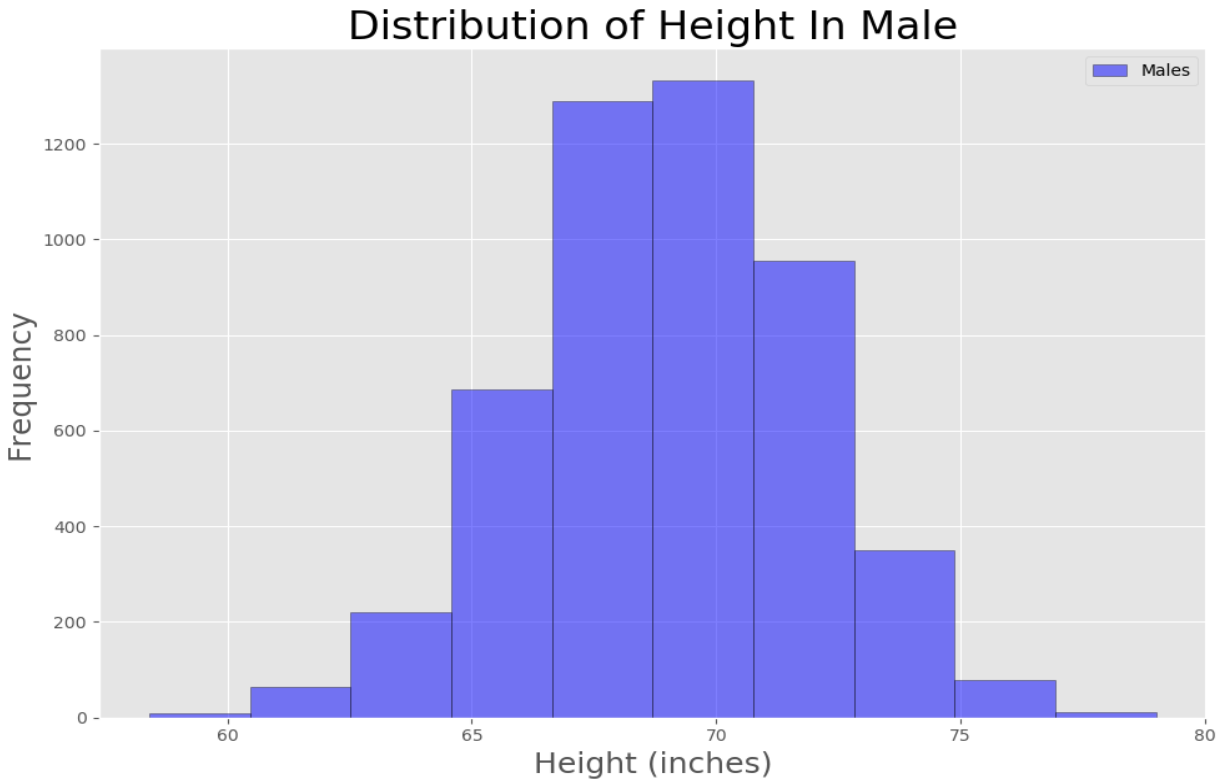
These are histograms:



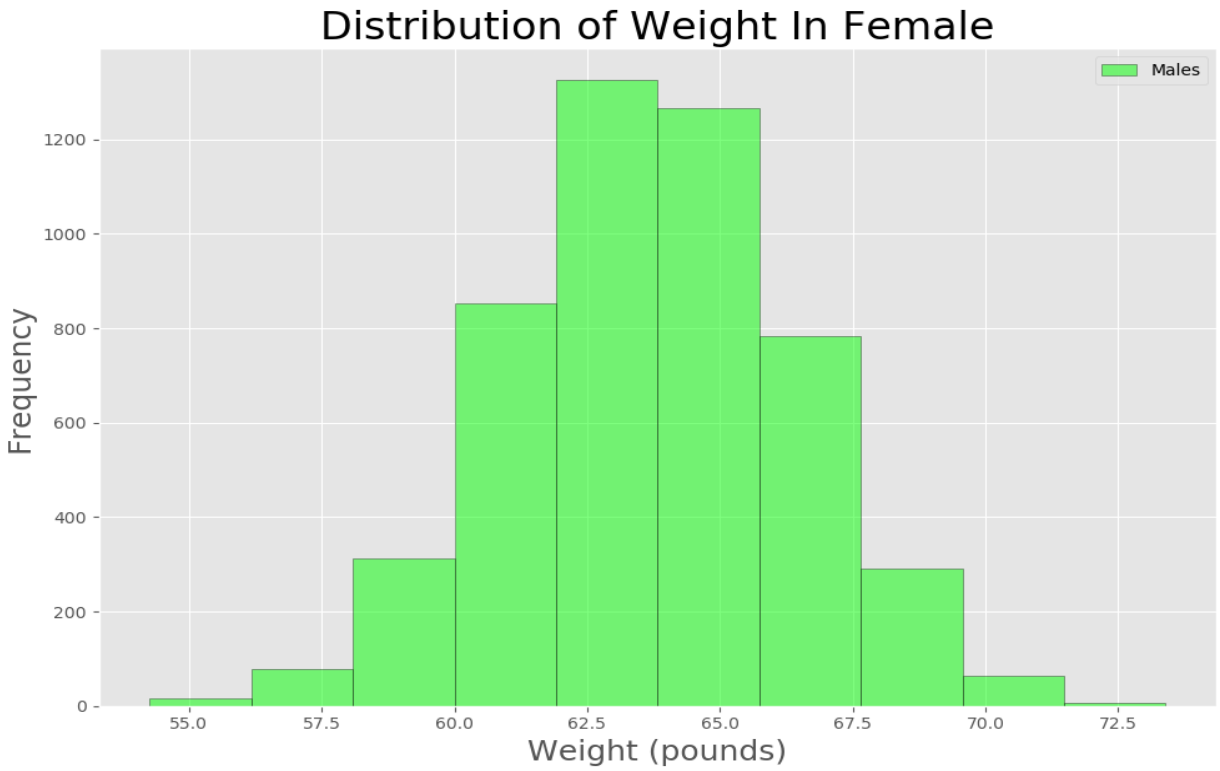


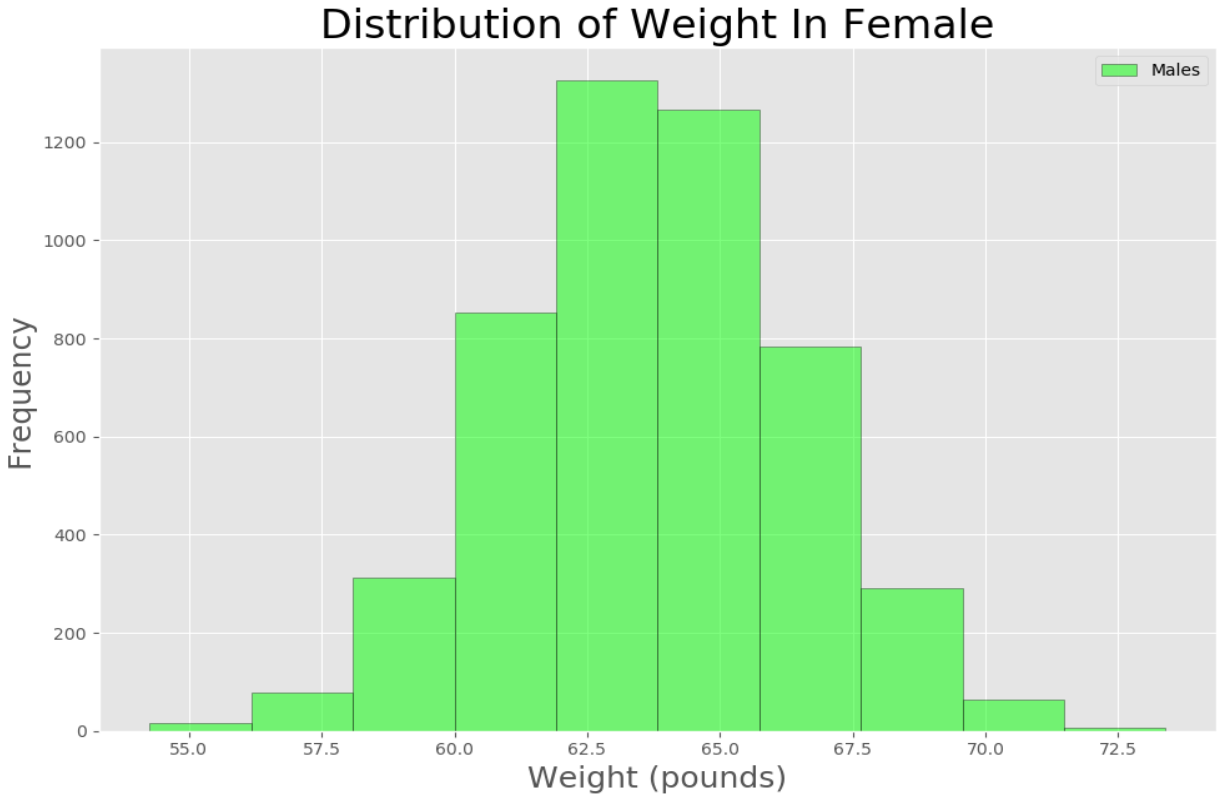
Both variables Height and Weight present a normal distribution. It can also be interesting as part of our exploratory analysis to plot the distribution of males and females in separated histograms**.**



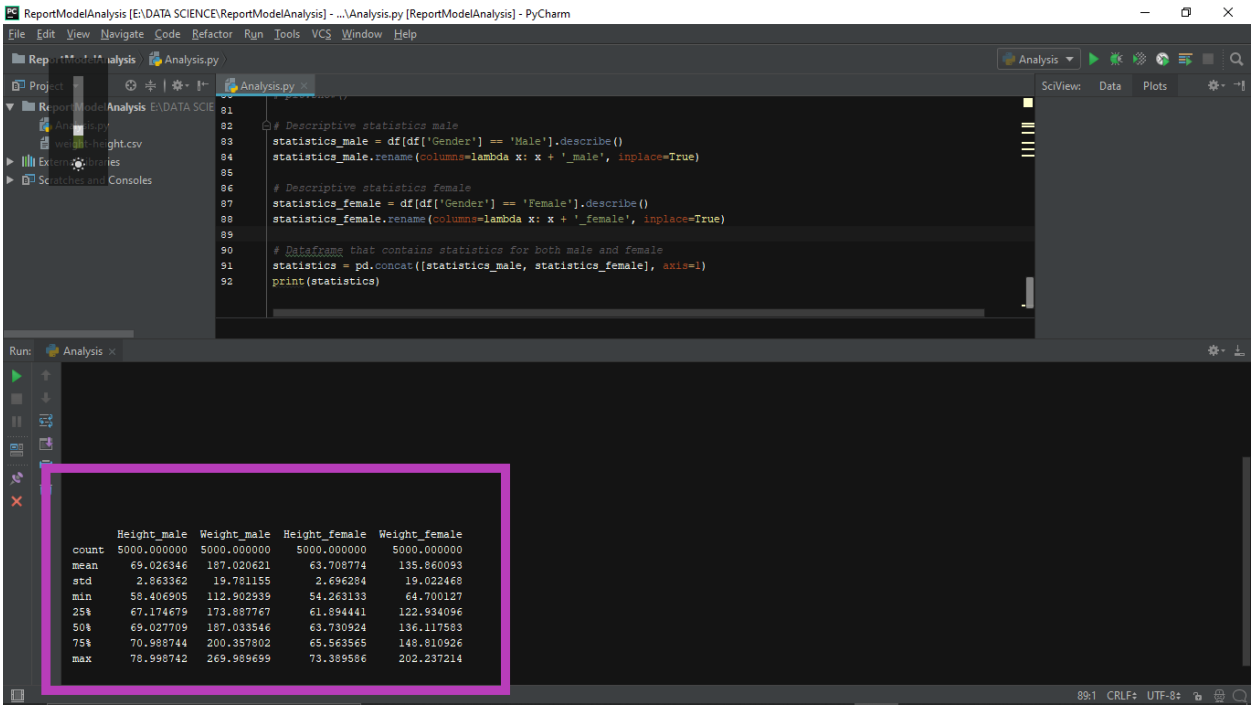


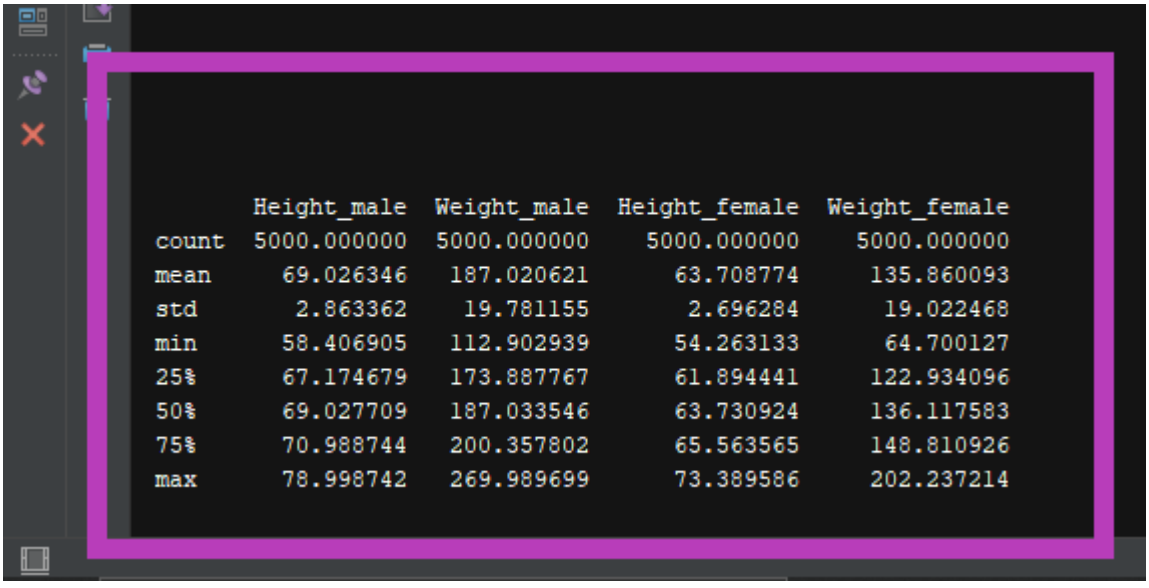
And for weight,


Distribution of Weight In Female
Males
Frequency
0
200
400
600
800
1000
1200
55.0
57.5
60.0
62.5
65.0
67.5
70.0
72.5
Weight (pounds)

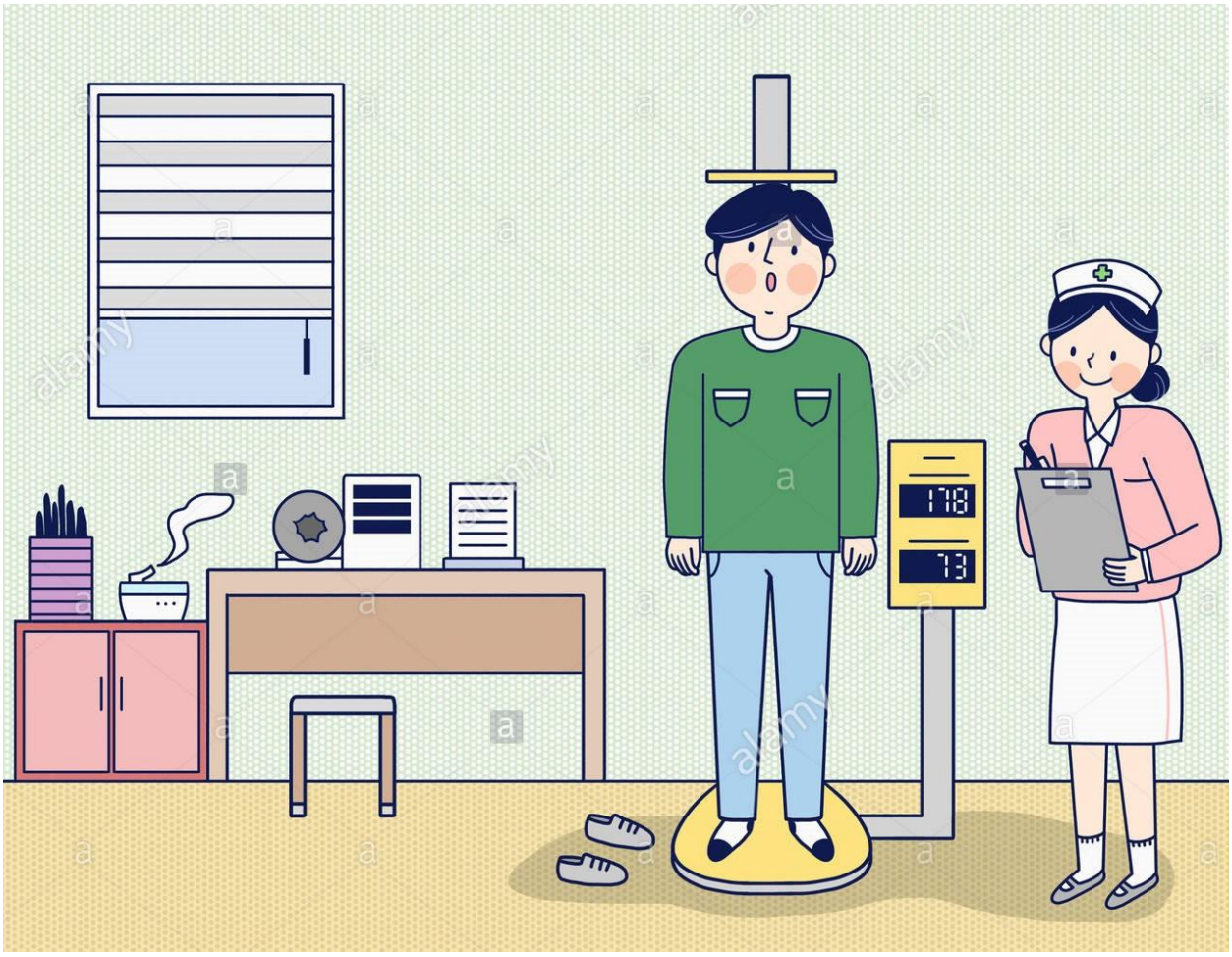

Distribution of Weight In Female
Males
Frequency
0
200
400
600
800
1000
1200
55.0
57.5
60.0
62.5
65.0
67.5
70.0
72.5
Weight (pounds)

Both height and weight present a normal distribution for males and females. Although the average of each distribution is larger for males, the spread of the distributions is similar for both genders. Pandas provides a method called describe that generates descriptive statistics of a dataset (central tendency, dispersion and shape.

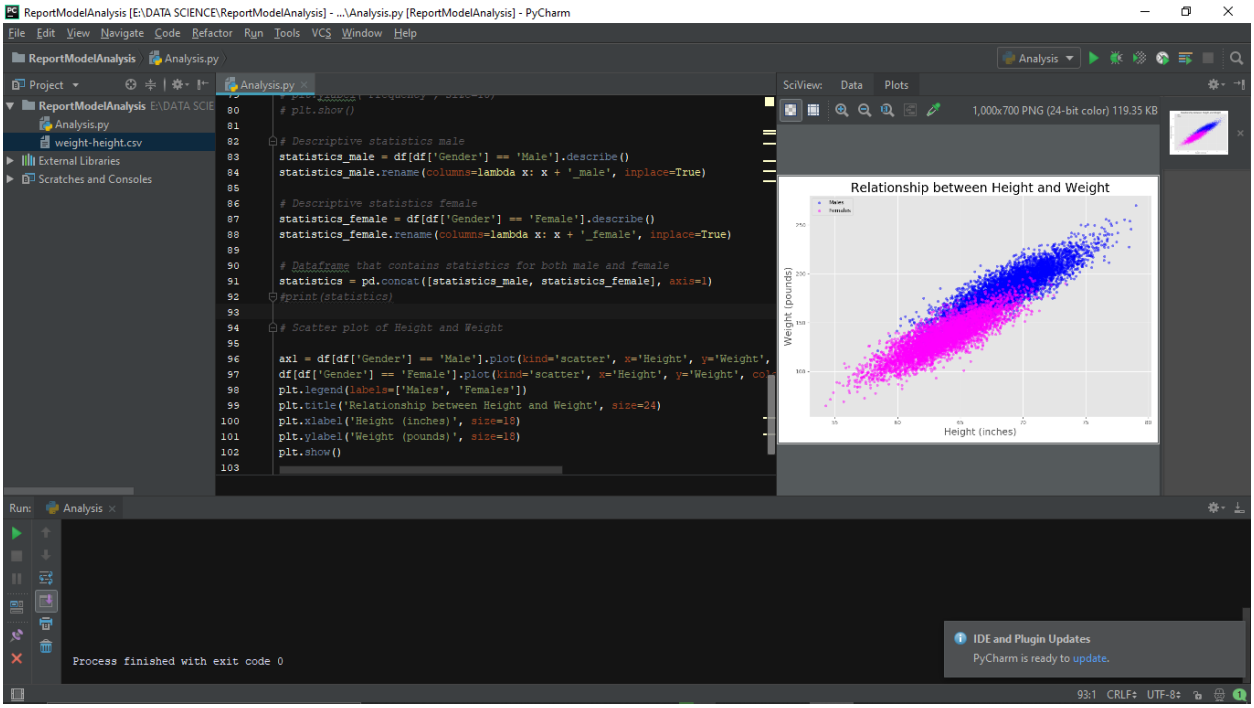


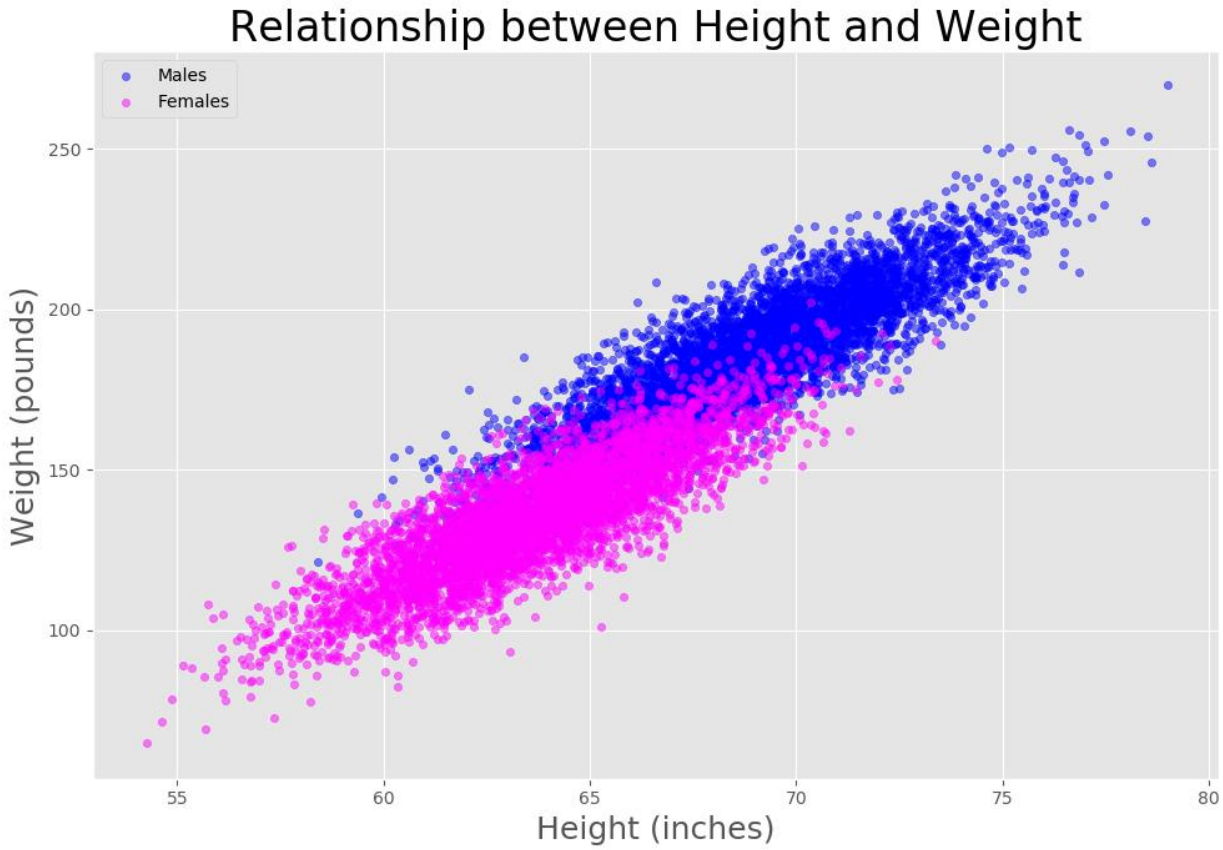


MODEL & ANALYSIS



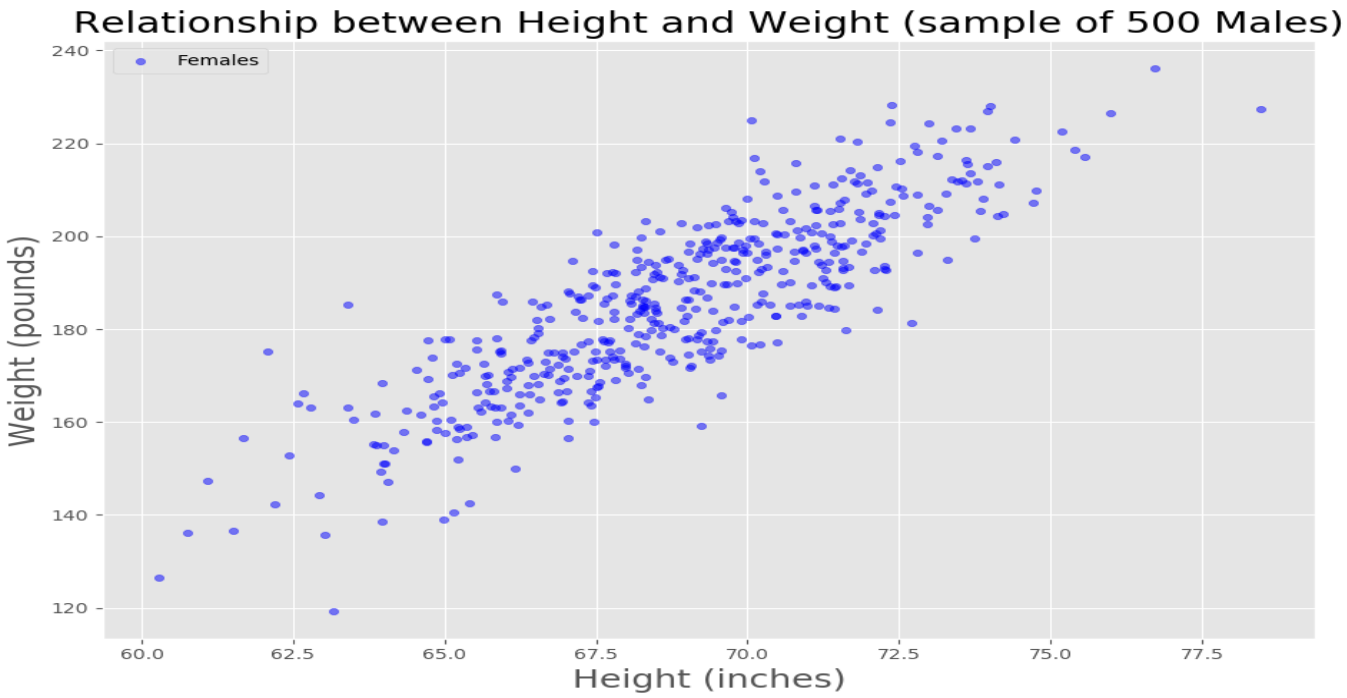
After all, now its time to decide which model we will use for prediction of weight by using height. As for now, we have two variables, one output and one input. But what if we want to include gender too. For a second, we focus on our situation, our problem is regression problem as it involves numbers.

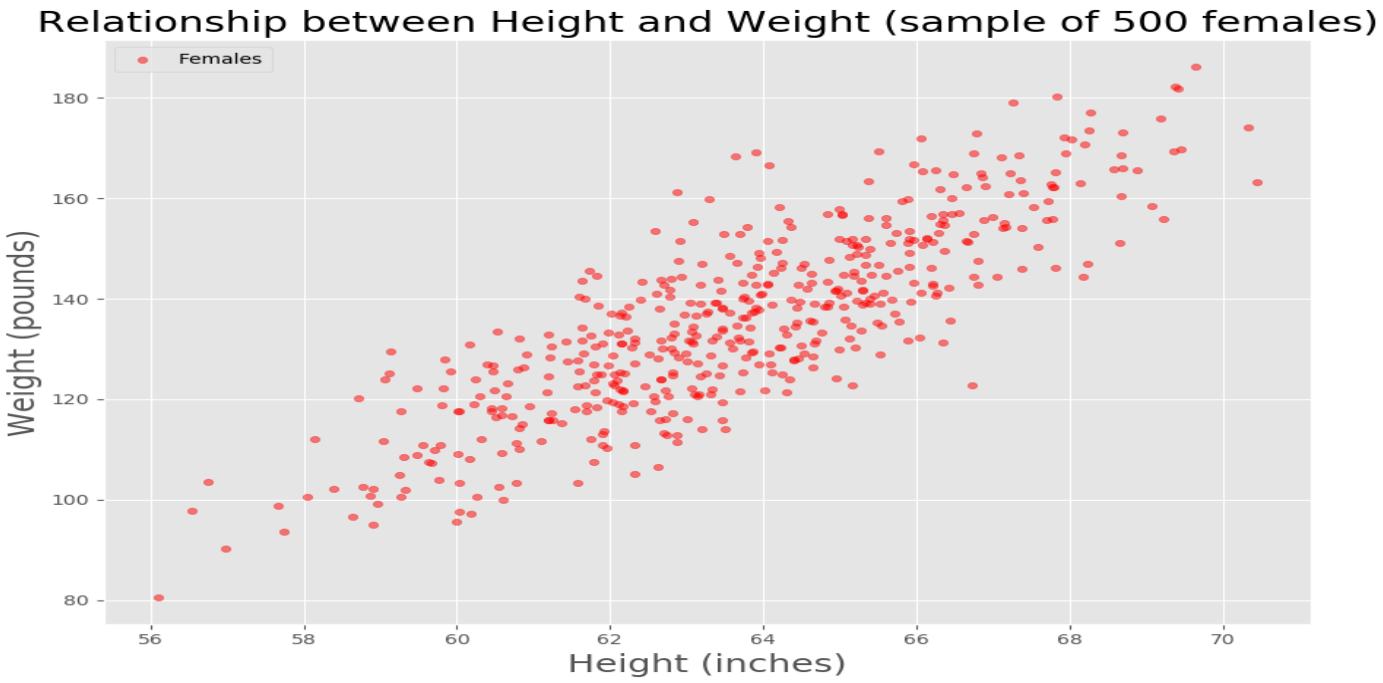

ReportModelAnalysis [E:\DATA SCIENCE\ReportModelAnalysis] - ...\Analysis.py [ReportModelAnalysis] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
ReportModelAnalysis
Analysis.py
weight-height.csv
External Libraries
Scratches and Consoles
# plt.show()
# Descriptive statistics male
statistics_male = df[df['Gender'] == 'Male'].describe()
statistics_male.rename(columns=lambda x: x + '_male', inplace=True)
# Descriptive statistics female
statistics_female = df[df['Gender'] == 'Female'].describe()
statistics_female.rename(columns=lambda x: x + '_female', inplace=True)
# Dataframe that contains statistics for both male and female
statistics = pd.concat([statistics_male, statistics_female], axis=1)
#print(statistics)
# Scatter plot of Height and Weight
ax1 = df[df['Gender'] == 'Male'].plot(kind='scatter', x='Height', y='Weight',
df[df['Gender'] == 'Female'].plot(kind='scatter', x='Height', y='Weight',
plt.legend(labels=['Males', 'Females'])
plt.title('Relationship between Height and Weight', size=24)
plt.xlabel('Height (inches)', size=18)
plt.ylabel('Weight (pounds)', size=18)
plt.show()
SciView: Data Plots
1,000x700 PNG (24-bit color) 119.35 KB
Relationship between Height and Weight
Weight (pounds)
Height (inches)
Run: Analysis
Process finished with exit code 0
IDE and Plugin Updates
PyCharm is ready to update.
93:1 CRLF: UTF-8:


Relationship between Height and Weight
Males
Females
Weight (pounds)
250
200
150
100
55
60
65
70
75
80
Height (inches)

Linear regression is a linear approach to model the relationship between a dependent variable (target variable) and one (simple regression) or more (multiple regression) independent variables. As we can see with increase in height, there is increase in weight as they are **directly proportional.**

We can study relationship of height and weight in male and female separately.





Simple linear regression is a linear approach to modeling the relationship between a dependent variable and an independent variable**,** obtaining a line that best fits the data.

y = a + b x

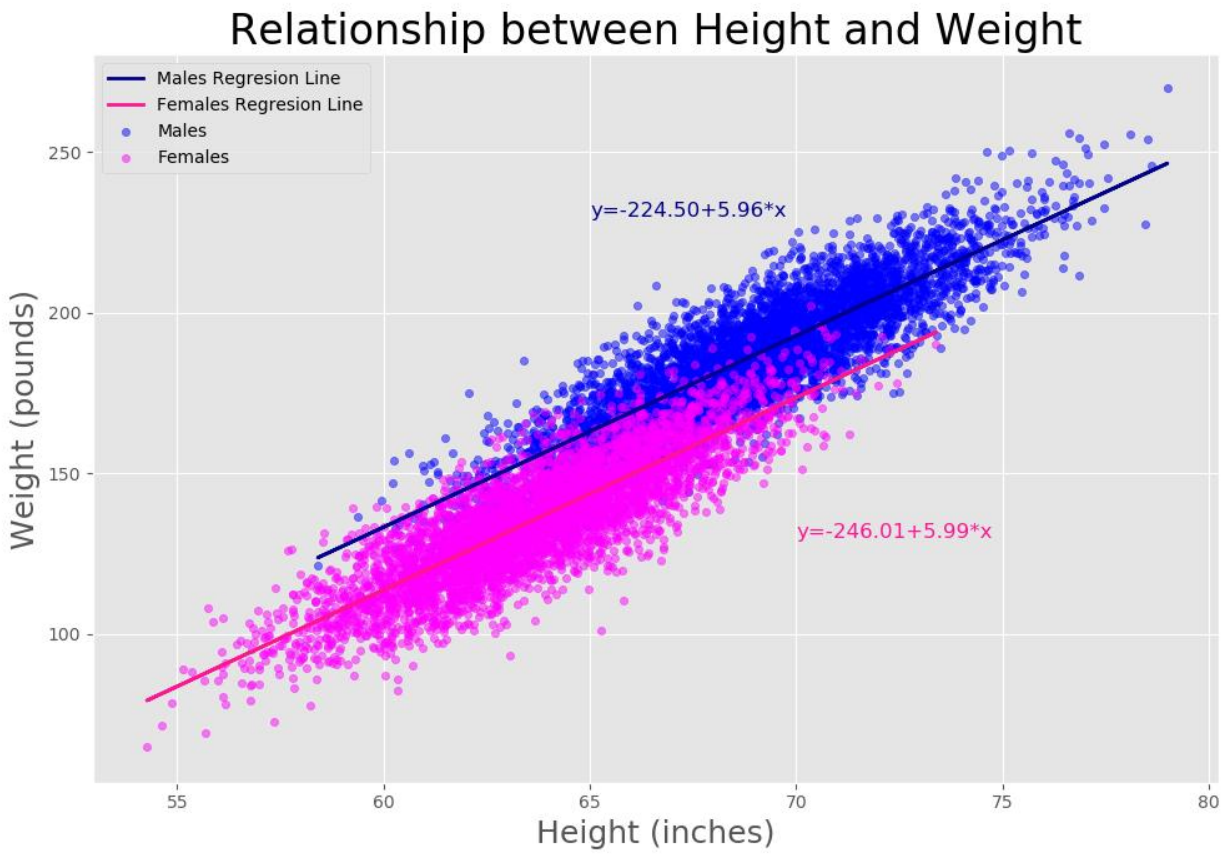
where **x** is the independent variable (height), **y** is the dependent variable (weight), **b** is the slope, and **a** is the intercept**.** The intercept represents the value of y when x is 0 and the slope indicates the steepness of the line. The objective is to obtain the line that best fits our data (the line that minimize the sum of square errors). The error is the difference between the real value **y** and the predicted value (y_hat), which is the value obtained using the calculated linear equation.

For checking error:

Error = y(real) – y(predicted)

= > y(real) – y(a+bx)

The numpy library function polyfit => numpy.polyfit(x,y,deg) fits a polynomial of degree *deg* to points *(x, y),* returning the polynomial coefficients that minimize the square error.

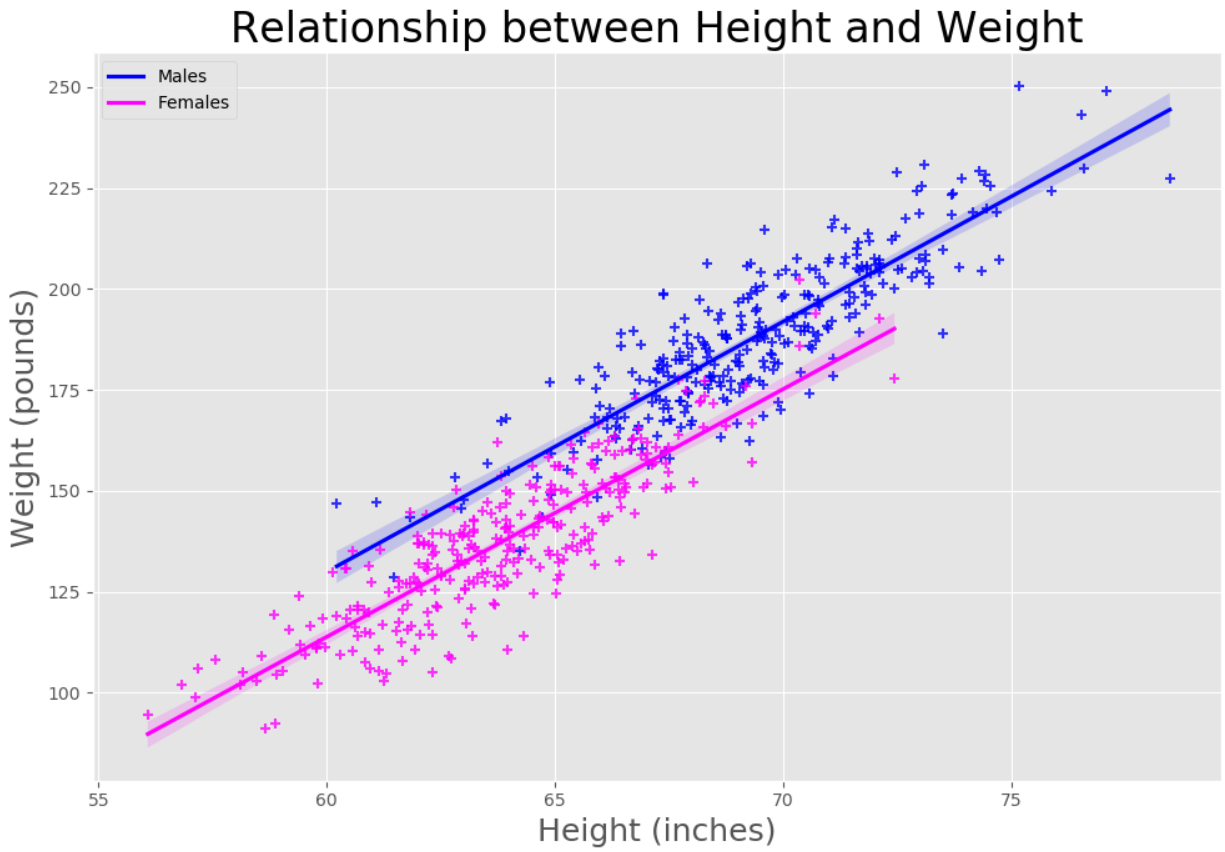


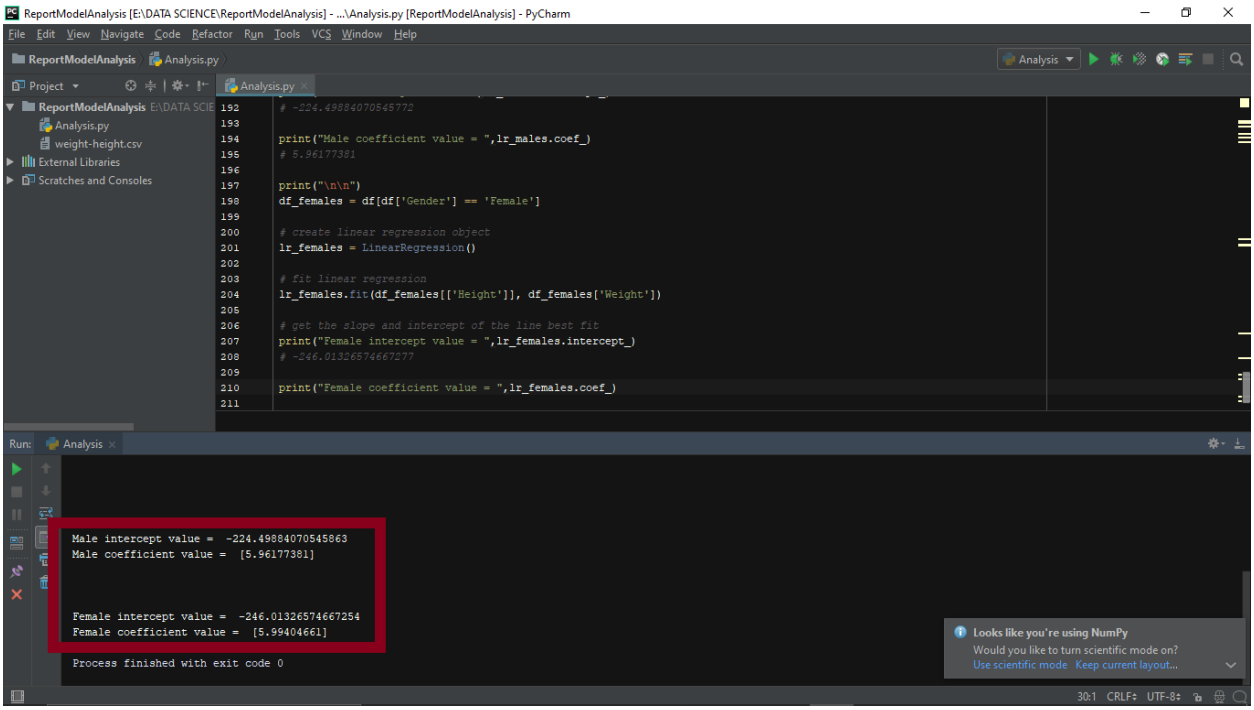
So here, we can see the clear picture of equations for both male and female.

Male = > y = -224.50 + (5.96 * x)

Female => y = -246.01 + (5.99 * x)

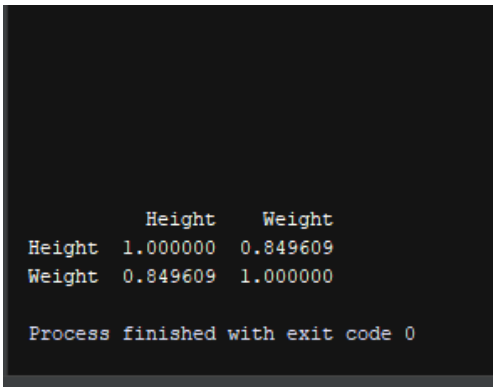
Furthermore, let study the relationship between height and weight. Are they linear too irrespective of gender?


Relationship between Height and Weight
Males
Females
Weight (pounds)
Height (inches)
250
225
200
175
150
125
100
55
60
65
70
75

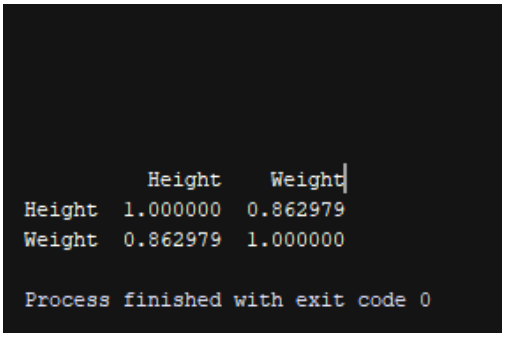

ReportModelAnalysis [E:\DATA SCIENCE\ReportModelAnalysis] - ...\Analysis.py [ReportModelAnalysis] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
ReportModelAnalysis
Analysis.py
Project
ReportModelAnalysis E:\DATA SCIE
Analysis.py
weight-height.csv
External Libraries
Scratches and Consoles
# -224.49884070545772
print("Male coefficient value = ",lr_males.coef_)
# 5.96177381
print("\n\n")
df_females = df[df['Gender'] == 'Female']
# create linear regression object
lr_females = LinearRegression()
# fit linear regression
lr_females.fit(df_females[['Height']], df_females['Weight'])
# get the slope and intercept of the line best fit
print("Female intercept value = ",lr_females.intercept_)
# -246.01326574667277
print("Female coefficient value = ",lr_females.coef_)
Run: Analysis
Male intercept value = -224.49884070545863
Male coefficient value = [5.96177381]
Female intercept value = -246.01326574667254
Female coefficient value = [5.99404661]
Process finished with exit code 0
Looks like you're using NumPy
Would you like to turn scientific mode on?
Use scientific mode Keep current layout...
30:1 CRLF UTF-8

Now we have got the intercept and coefficient values for both male and female model. We can also check the correlation between weight and height for male and female separately.

For females:



For males:



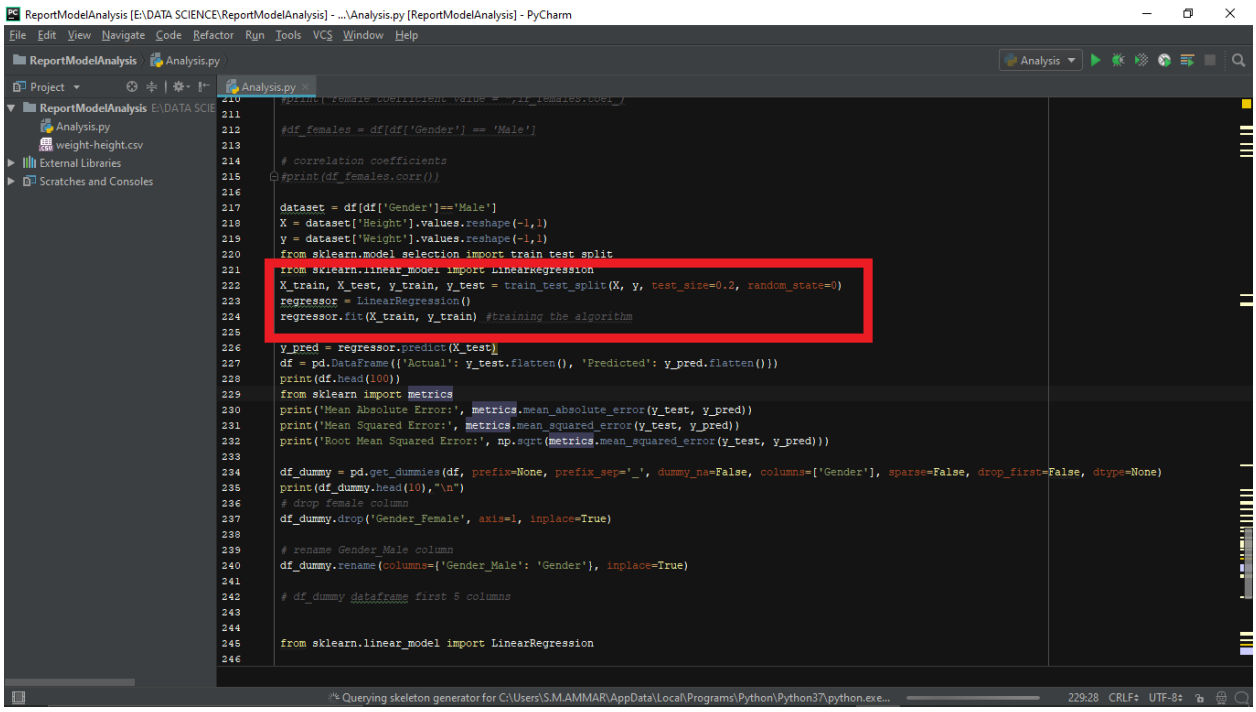
- Females correlation coefficient: **0.849608**

- Males correlation coefficient: **0.8629788**

So, our final models for calculating weight with respect to height are:

- **Males → Weight = -224.50+5.96*Height**
- **Females → Weight = -246.01+5.99*Height**

Splitting Data



We can perform prediction and our accuracy will also be checked in a same way. We can also check the accuracy of our model manually too.

Accuracy of Female Model:

```
        Actual   Predicted
0   131.130447  130.899225
1   124.467491  134.662850
2   158.212861  142.937964
3   134.092031  133.653062
4   152.939283  134.474242
..         ...         ...
95  139.114234  125.815964
96  131.970135  131.749151
97  133.066749  137.123620
98  140.328416  136.289544
99  142.433424  138.590688

[100 rows x 2 columns]
Mean Absolute Error: 7.883860820364109
Mean Squared Error: 98.24910085378063
Root Mean Squared Error: 9.912068444768762

Process finished with exit code 0
```

Accuracy of male model:

```
        Actual   Predicted
0   192.470770  176.574467
1   142.252546  146.465472
2   213.457233  199.793112
3   152.896965  180.213770
4   163.495203  178.241772
..         ...         ...
95  182.856213  195.514418
96  201.795504  186.069621
97  176.936187  185.631572
98  184.833520  193.132403
99  166.171185  173.936738

[100 rows x 2 columns]
Mean Absolute Error: 7.946830514126609
Mean Squared Error: 99.2717818984359
Root Mean Squared Error: 9.963522564757703

Process finished with exit code 0
```

So, you can see these are not perfect models but they are good enough to predict weight in real life (close to original ones).

Now we can also create a model with gender and height as input.

```
      Height      Weight  Gender_Female  Gender_Male
0  73.847017  241.893563              0            1
1  68.781904  162.310473              0            1
2  74.110105  212.740856              0            1
3  71.730978  220.042470              0            1
4  69.881796  206.349801              0            1
5  67.253016  152.212156              0            1
6  68.785081  183.927889              0            1
7  68.348516  167.971110              0            1
8  67.018950  175.929440              0            1
9  63.456494  156.399676              0            1

Intercept Value =  -244.92350252069937
Coefficient values =  [ 5.97694123 19.37771052]

Process finished with exit code 0
```

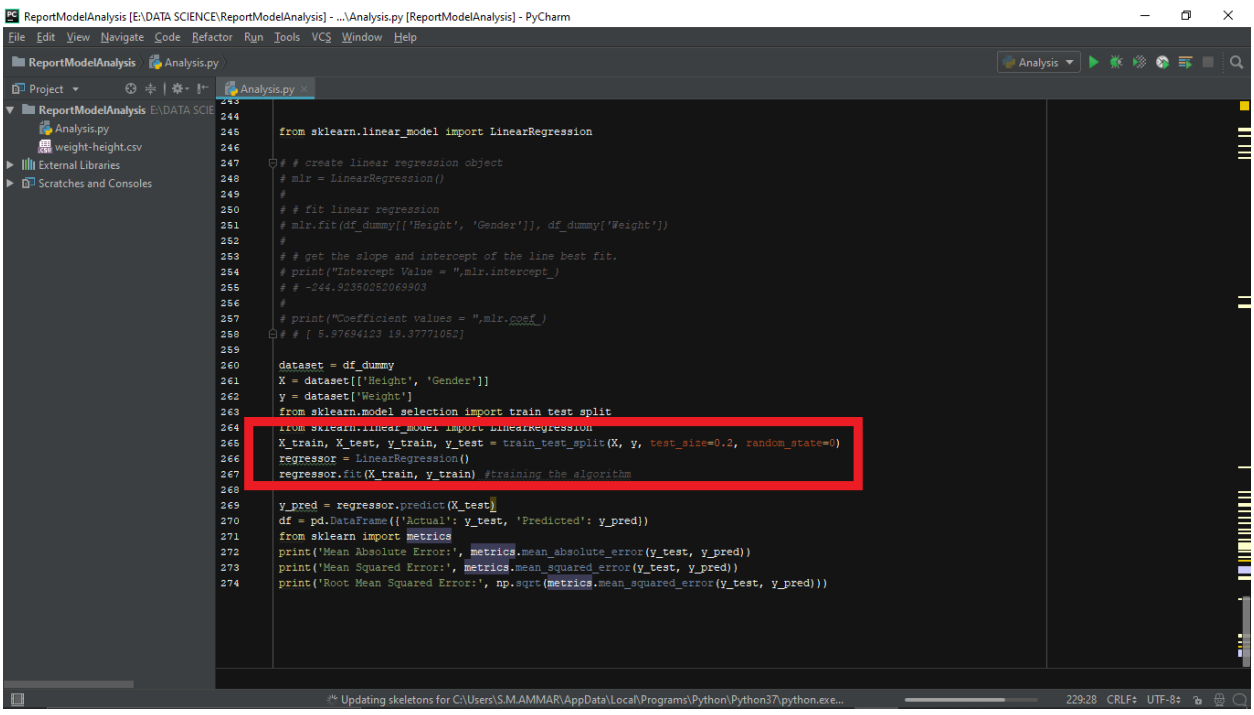
New Model:

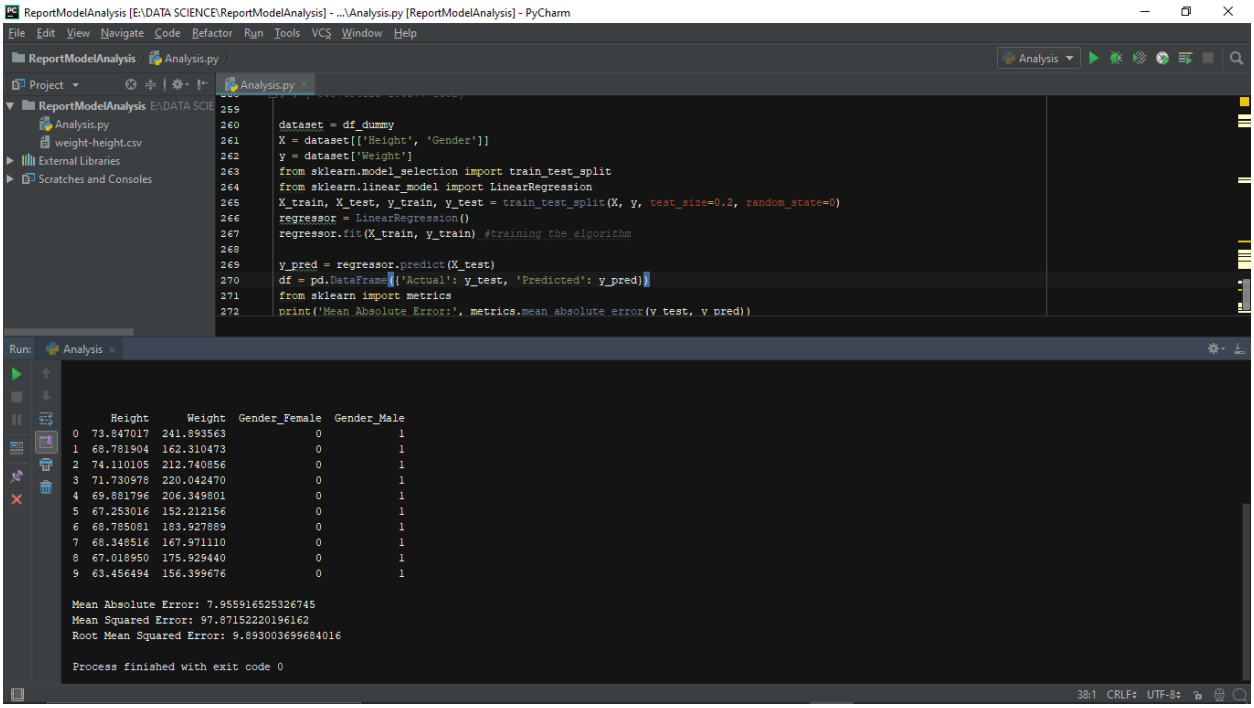
Weight = -244.9235+5.9769*Height+19.3777*Gender

Male → Weight = -244.9235+5.9769*Height+19.3777*1= -225.5458+5.9769*Height

Female → Weight = -244.9235+5.9769*Height+19.3777*0 =-244.9235+5.9769*Height

Data splitting and testing:




ReportModelAnalysis [E:\DATA SCIENCE\ReportModelAnalysis] - ...\Analysis.py [ReportModelAnalysis] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
ReportModelAnalysis
Analysis.py
Project
ReportModelAnalysis E:\DATA SCIE
Analysis.py
weight-height.csv
External Libraries
Scratches and Consoles
dataset = df_dummy
X = dataset[['Height', 'Gender']]
y = dataset['Weight']
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
regressor = LinearRegression()
regressor.fit(X_train, y_train) #training the algorithm
y_pred = regressor.predict(X_test)
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
Run: Analysis
Height Weight Gender_Female Gender_Male
0 73.847017 241.893563 0 1
1 68.781904 162.310473 0 1
2 74.110105 212.740856 0 1
3 71.730978 220.042470 0 1
4 69.881796 206.349801 0 1
5 67.253016 152.212156 0 1
6 68.785081 183.927889 0 1
7 68.348516 167.971110 0 1
8 67.018950 175.929440 0 1
9 63.456494 156.399676 0 1
Mean Absolute Error: 7.955916525326745
Mean Squared Error: 97.87152220196162
Root Mean Squared Error: 9.893003699684016
Process finished with exit code 0
38:1 CRLF: UTF-8:

The End