Kevin Ip
DBDA X427

Project Report

Introduction

As a newcomer to data science and a student taking an intro course to learn skills used in machine learning, I wanted to solve a problem that is well known, "Will a parent's height affect an adult child's maximum height?" Although someone without data science knowledge would already known the answer, we can prove it by analyzing and by applying statistics and machine learning techniques. Linear regression is an effective and efficient method to answer this question and it is used in many other prediction problems as well.

The dataset I will use was originally used by the statistician Francis Galton to analyze the same problem. Galton was famous for his work on heredity, eugenics, and genetics. He was also the half-cousin of Charles Darwin.
One can read more about him here: https://en.wikipedia.org/wiki/Francis_Galton
The data set is found here: http://www.randomservices.org/random/data

I have created two python notebooks. The first notebook uses the father and the mother's height separately to form two regression lines to predict a child's height. The second notebook uses both the father and mother's height. The second notebook uses the father and the mother's height together to form one regression line to predict a child's height, this should be the more accurate prediction.

Requirements

- Software: Anaconda, Jupyter Notebook
- Python Modules: numpy, pandas, sklearn, matplotlib, tkinter (second notebook)
- Data Set: http://www.randomservices.org/random/data (height data)

Description

First Notebook – single variable linear regression modeling

In the first notebook, we read the data in ‘family.txt’ using the ‘pandas’ python package in box [2]. The data contains 6 columns: Family Group, Father’s Height, Mother’s Height, Gender of the Adult Child, Adult Child’s Height, and the number of kids. The data contains 898 records. We want to perform single linear regression modeling on the dad’s height vs. child’s height and the mom’s height vs. child’s height separately, giving us two linear regression models.

Then we apply a few dataframe methods to get a sense of the data, these methods include: describe(), dtypes(), and head(). Then one would isolate the heights of the dad, mom, and the child. After we obtained the isolated data, we can fit the data to a Linear Regression model shown in box [6]. The model can be recognized as a linear equation taking the form of y = mx + b where y is the predicted value, m is the slope or the coefficient calculated from the model, x is the input value used to create the model, and b is the y-intercept of the axis when x is 0. The coefficient m and the intercept b are calculated by the linear regression model from the ‘sklearn’ python package.

$y = mx + b$

Dad’s Linear Regression Equation:
$y = 0.3994x_1 + 39.1104$

Mom’s Linear Regression Equation:
$y = 0.3132x_1 + 46.6908$

To test input values into the Linear Regression model we can use the linspace() method to generate input heights into our linear regression line and the predict() method to generate a predicted height of the adult child.

To visualize the data and calculations we can use matplotlib to generate graphs for us. In box [9], we generate three subplots. One subplot (histogram) for the distribution of the dad’s height, one subplot (histogram) for the distribution of the children’s heights, and one subplot (scatter) showing all dad’s heights, the regression line generated from the input heights from linspace() is also shown.

Similarly, in box [10], we generate three subplots. One subplot (histogram) for the distribution of the mom’s height, one subplot (histogram) for the distribution of the children’s heights, and one subplot (scatter) showing all dad’s heights, the regression line generated from the input heights from linspace() is also shown.

In box [11], we combine both parent’s heights in the same scatterplot to compare and visualize the data more easily. We see that the mother is shorter than the father on average.

Boxes [12 – 15] involve centering the data around the mean values of the father's and the mother's heights around 0. After doing this we can fit both models to normalization and create respective scatterplots for the mother and the father. The two scatterplots formed from the normalized data are centered around 0 on the x-axis for the parent's height and are also centered around 0 on the y-axis for the child's height. This graph allows us to look for any anomalies much easier.

<u>Second Notebook – multiple variable linear regression modeling</u>

In the second notebook, we read the same data in 'family.txt' using the 'pandas' python package in box [2]. The data contains 6 columns: Family Group, Father's Height, Mother's Height, Gender of the Adult Child, Adult Child's Height, and the number of kids. The data contains 898 records. We want to perform multiple linear regression modeling using both parent's heights as separate variables, giving us one linear regression model.

Then we apply a few dataframe methods to get a sense of the data, these methods include: describe(), dtypes(), and head(). Then one would isolate the heights of the dad, mom, and the child. After we obtained the isolated data, we can fit the data to a Linear Regression model shown in box [6]. The model can be recognized as a linear equation taking the form of $y = m_1x_1 + m_2x_2 + b$ where y is the predicted value, m_1 is the slope or the coefficient calculated from the dad model, m_2 is the slope or the coefficient calculated from the mom model, x_1 is the input value used to create the dad model, x_2 is the input value used to create the mom model, and b is the y-intercept of the axis when x is 0. The coefficient m and the intercept b are calculated by the linear regression model from the 'sklearn' python package.

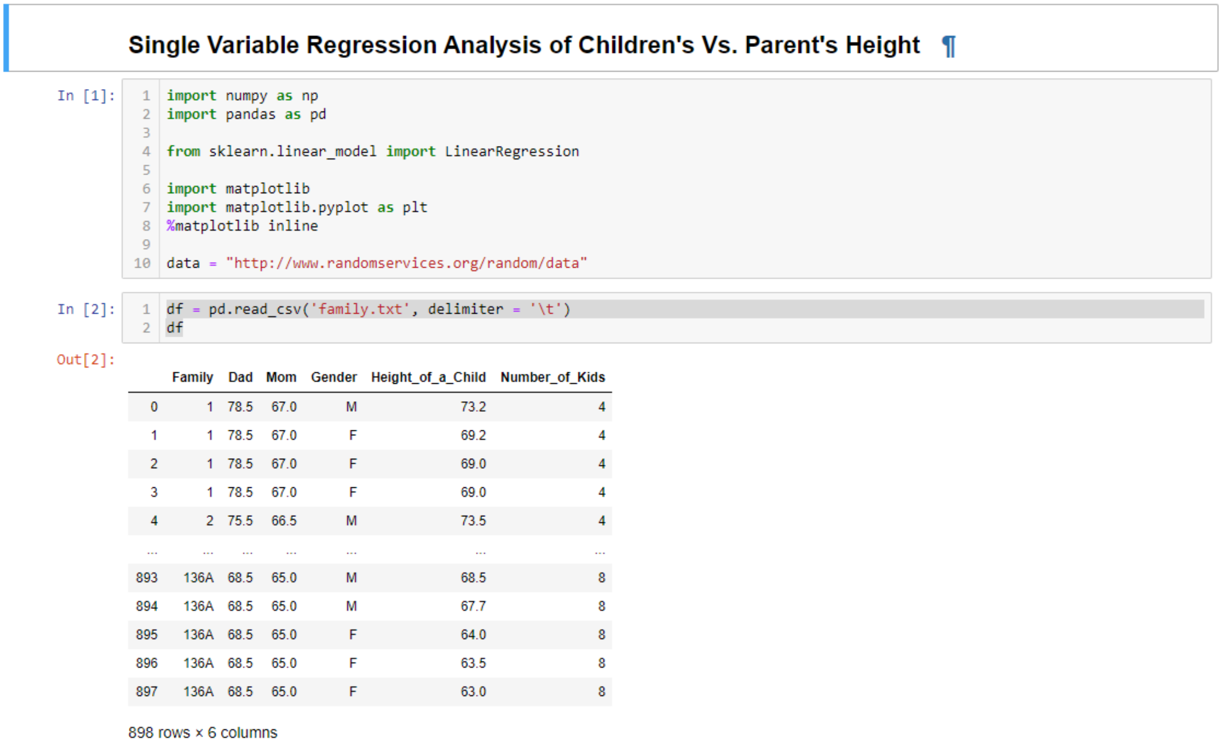
$y = m_1x_1 + m_2x_2 + b$

Both parent's height as variables, Linear Regression Equation:
$y = 0.3799x_1 + 0.2832x_2 + 22.3097$

To test input values into the Linear Regression model we can use the linspace() method to generate input heights into our linear regression line and the predict() method to generate a predicted height of the adult child.

In box [7], instead of using graphs to visualize the data and calculations, I wanted to try something different and created a GUI using the 'tkinter' python package. In the GUI, one can input the father's height and the mother's height and obtain the predicted adult child's height. The GUI also shows two scatterplot distributions of predicted values using just the father's model or just the mother's model. Regression lines are not shown here because they were calculated in the first notebook.

Screenshots



Single Variable Regression Analysis of Children's Vs. Parent's Height ¶

In [1]:

```python
import numpy as np
import pandas as pd

from sklearn.linear_model import LinearRegression

import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

data = "http://www.randomservices.org/random/data"
```

In [2]:

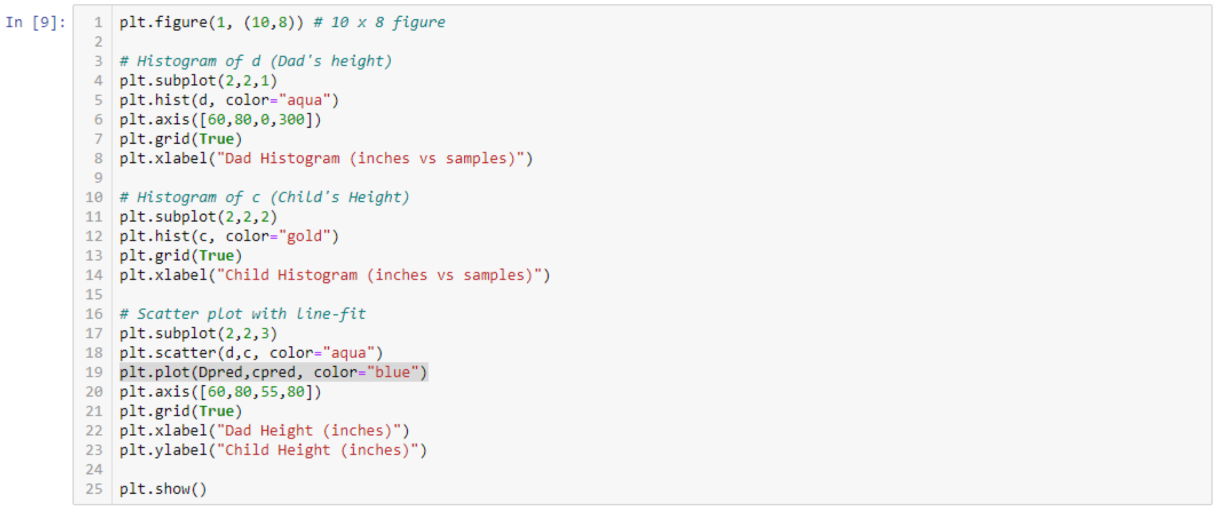
```python
df = pd.read_csv('family.txt', delimiter = '\t')
df
```

Out[2]:

	Family	Dad	Mom	Gender	Height_of_a_Child	Number_of_Kids
0	1	78.5	67.0	M	73.2	4
1	1	78.5	67.0	F	69.2	4
2	1	78.5	67.0	F	69.0	4
3	1	78.5	67.0	F	69.0	4
4	2	75.5	66.5	M	73.5	4
...	...	...	...	...	...	...
893	136A	68.5	65.0	M	68.5	8
894	136A	68.5	65.0	M	67.7	8
895	136A	68.5	65.0	F	64.0	8
896	136A	68.5	65.0	F	63.5	8
897	136A	68.5	65.0	F	63.0	8

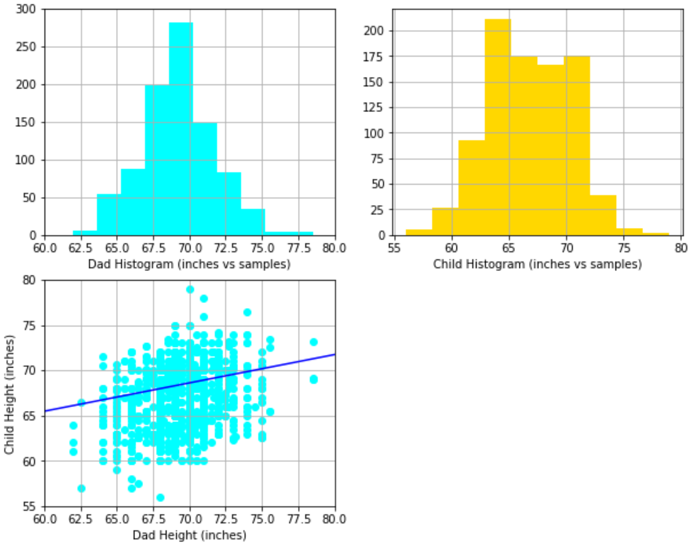
898 rows × 6 columns

Height Data Set

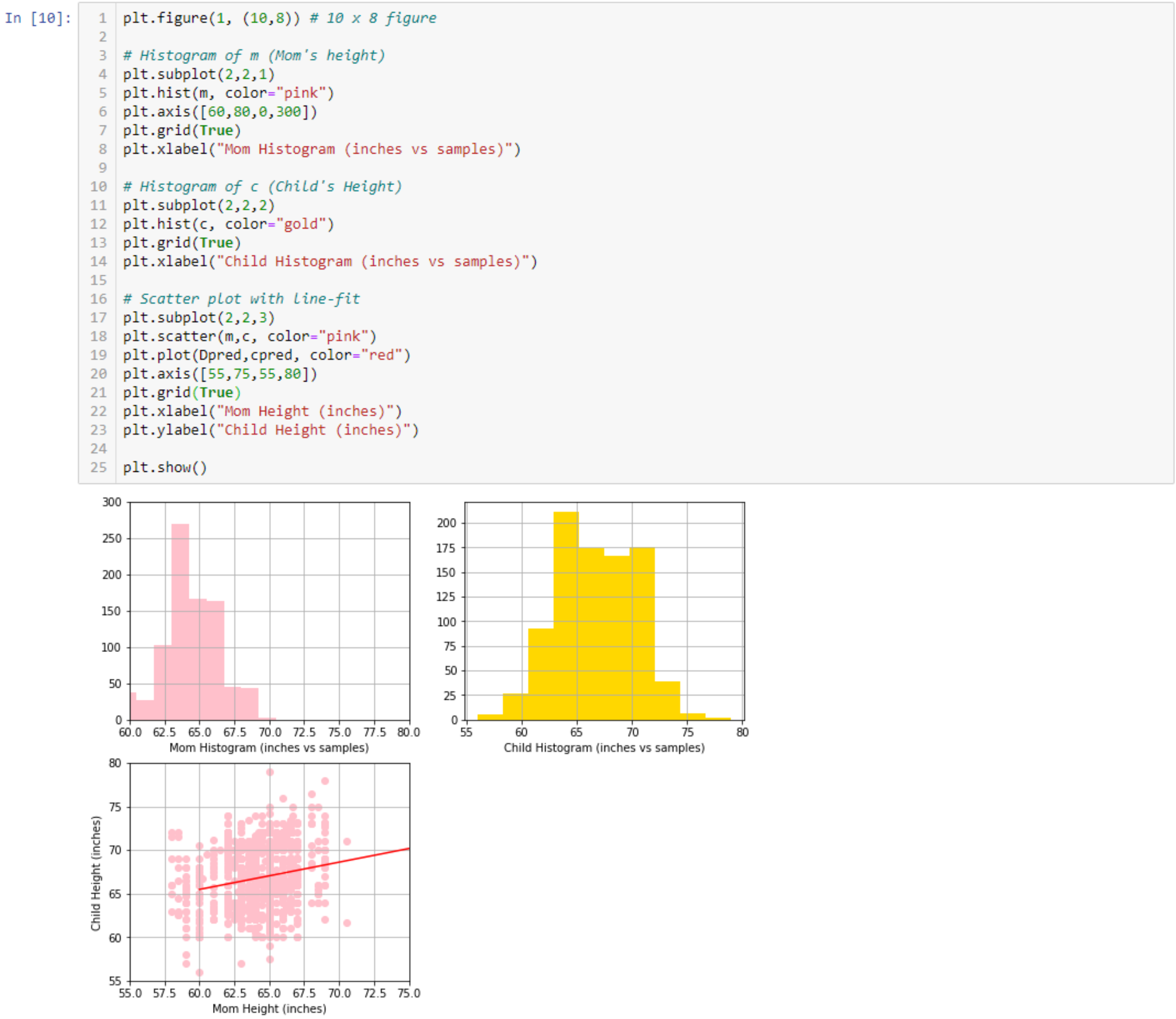


In [9]:

```python
plt.figure(1, (10,8)) # 10 x 8 figure

# Histogram of d (Dad's height)
plt.subplot(2,2,1)
plt.hist(d, color="aqua")
plt.axis([60,80,0,300])
plt.grid(True)
plt.xlabel("Dad Histogram (inches vs samples)")

# Histogram of c (Child's Height)
plt.subplot(2,2,2)
plt.hist(c, color="gold")
plt.grid(True)
plt.xlabel("Child Histogram (inches vs samples)")

# Scatter plot with line-fit
plt.subplot(2,2,3)
plt.scatter(d,c, color="aqua")
plt.plot(Dpred,cpred, color="blue")
plt.axis([60,80,55,80])
plt.grid(True)
plt.xlabel("Dad Height (inches)")
plt.ylabel("Child Height (inches)")

plt.show()
```



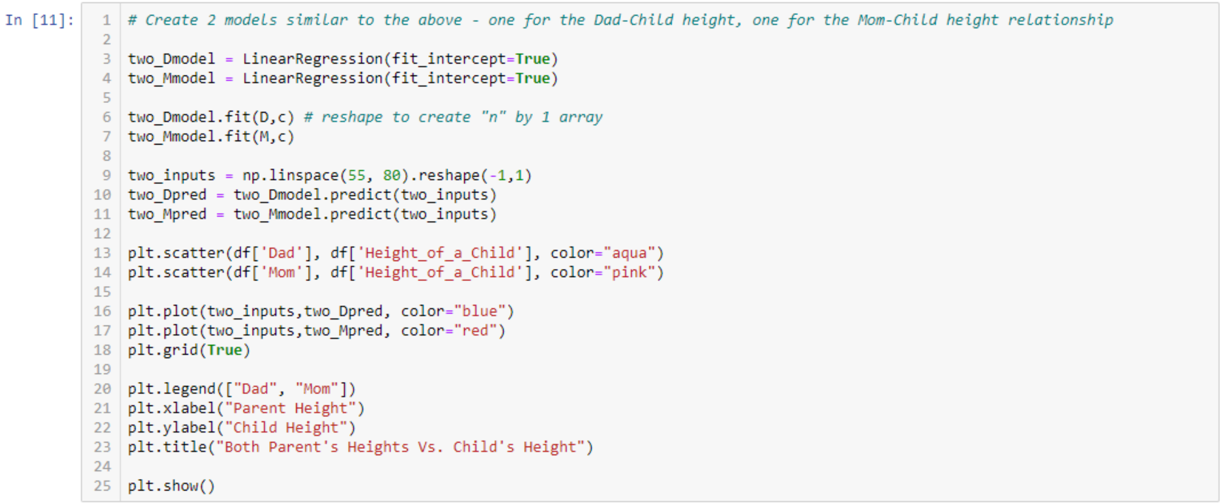
Father's Heights Subplots including height distributions (image 1 and image 2) and regression modeling (image 3)



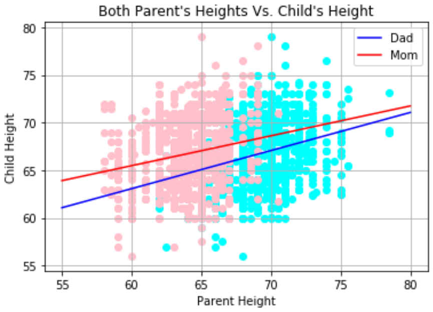
In [10]:

```
plt.figure(1, (10,8)) # 10 x 8 figure

# Histogram of m (Mom's height)
plt.subplot(2,2,1)
plt.hist(m, color="pink")
plt.axis([60,80,0,300])
plt.grid(True)
plt.xlabel("Mom Histogram (inches vs samples)")

# Histogram of c (Child's Height)
plt.subplot(2,2,2)
plt.hist(c, color="gold")
plt.grid(True)
plt.xlabel("Child Histogram (inches vs samples)")

# Scatter plot with Line-fit
plt.subplot(2,2,3)
plt.scatter(m,c, color="pink")
plt.plot(Dpred,cpred, color="red")
plt.axis([55,75,55,80])
plt.grid(True)
plt.xlabel("Mom Height (inches)")
plt.ylabel("Child Height (inches)")

plt.show()
```

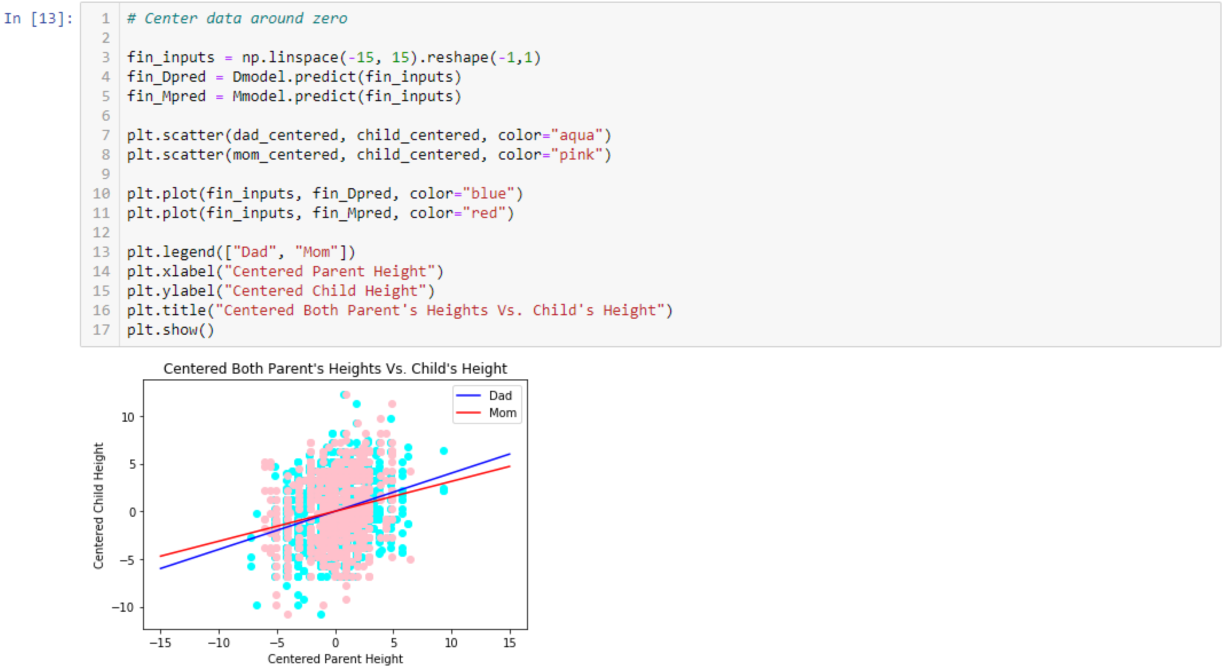
Mother's Heights Subplots including height distributions (image 1 and image 2) and regression modeling (image 3)



```python
# Create 2 models similar to the above - one for the Dad-Child height, one for the Mom-Child height relationship

two_Dmodel = LinearRegression(fit_intercept=True)
two_Mmodel = LinearRegression(fit_intercept=True)

two_Dmodel.fit(D,c) # reshape to create "n" by 1 array
two_Mmodel.fit(M,c)

two_inputs = np.linspace(55, 80).reshape(-1,1)
two_Dpred = two_Dmodel.predict(two_inputs)
two_Mpred = two_Mmodel.predict(two_inputs)

plt.scatter(df['Dad'], df['Height_of_a_Child'], color="aqua")
plt.scatter(df['Mom'], df['Height_of_a_Child'], color="pink")

plt.plot(two_inputs,two_Dpred, color="blue")
plt.plot(two_inputs,two_Mpred, color="red")
plt.grid(True)

plt.legend(["Dad", "Mom"])
plt.xlabel("Parent Height")
plt.ylabel("Child Height")
plt.title("Both Parent's Heights Vs. Child's Height")

plt.show()
```



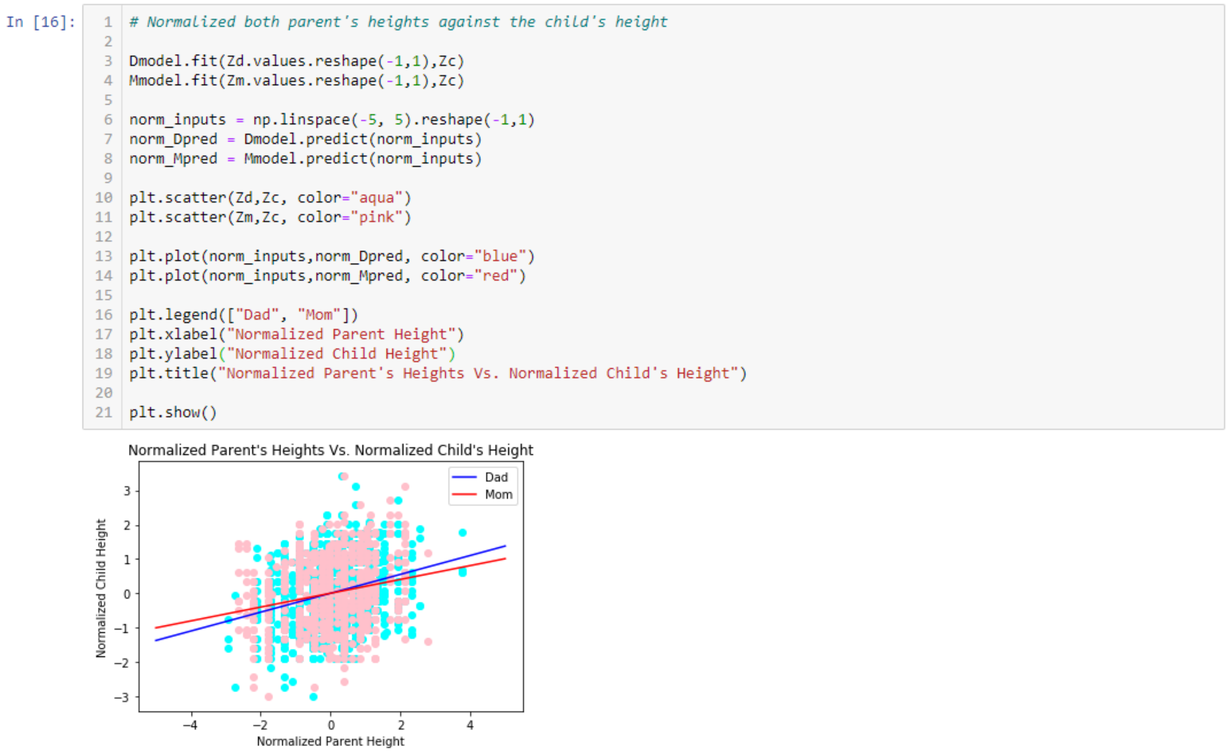
Both Father’s and Mother’s Scatterplots combined



In [13]:

```
# Center data around zero

fin_inputs = np.linspace(-15, 15).reshape(-1,1)
fin_Dpred = Dmodel.predict(fin_inputs)
fin_Mpred = Mmodel.predict(fin_inputs)

plt.scatter(dad_centered, child_centered, color="aqua")
plt.scatter(mom_centered, child_centered, color="pink")

plt.plot(fin_inputs, fin_Dpred, color="blue")
plt.plot(fin_inputs, fin_Mpred, color="red")

plt.legend(["Dad", "Mom"])
plt.xlabel("Centered Parent Height")
plt.ylabel("Centered Child Height")
plt.title("Centered Both Parent's Heights Vs. Child's Height")
plt.show()
```

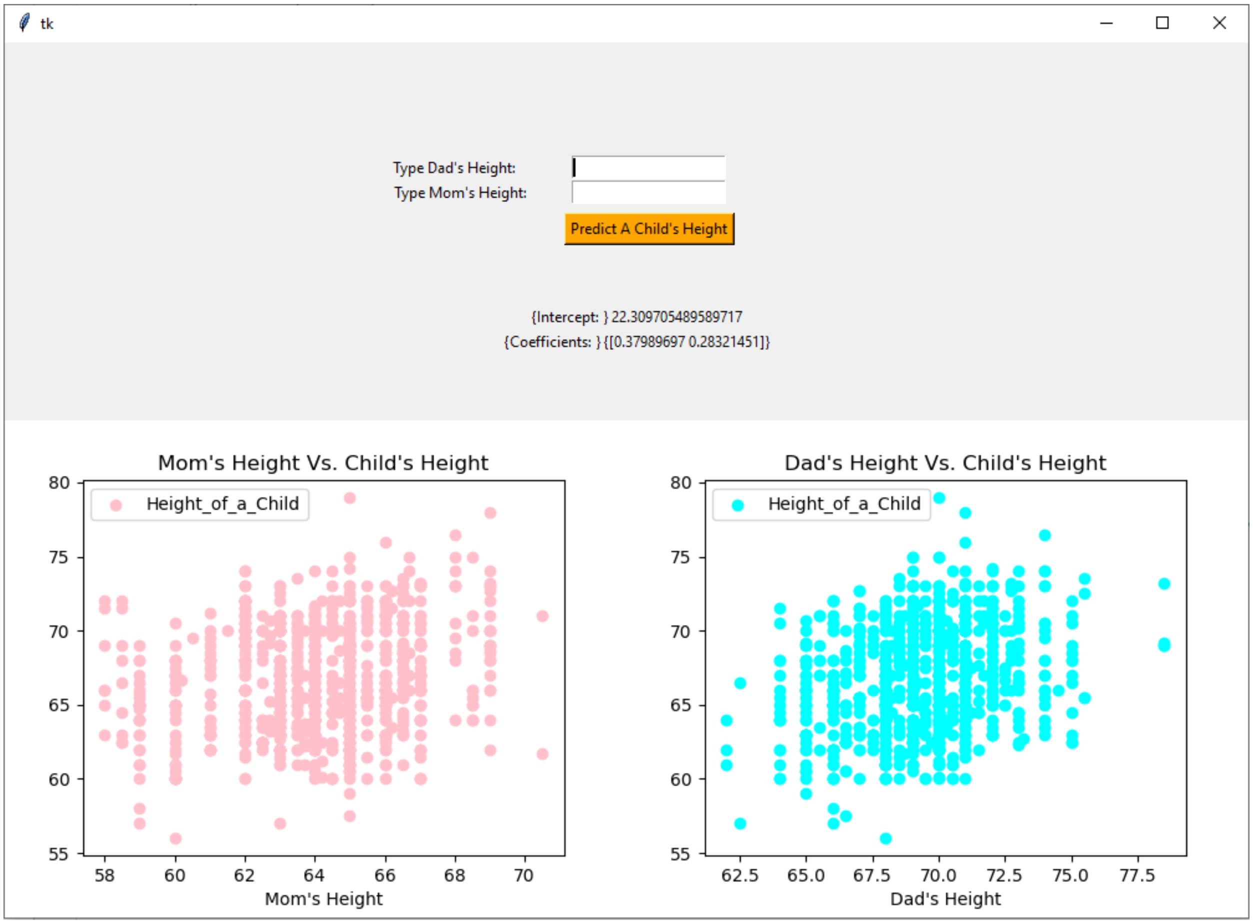
Centered Scatterplots



In [16]:

```
# Normalized both parent's heights against the child's height

Dmodel.fit(Zd.values.reshape(-1,1),Zc)
Mmodel.fit(Zm.values.reshape(-1,1),Zc)

norm_inputs = np.linspace(-5, 5).reshape(-1,1)
norm_Dpred = Dmodel.predict(norm_inputs)
norm_Mpred = Mmodel.predict(norm_inputs)

plt.scatter(Zd,Zc, color="aqua")
plt.scatter(Zm,Zc, color="pink")

plt.plot(norm_inputs,norm_Dpred, color="blue")
plt.plot(norm_inputs,norm_Mpred, color="red")

plt.legend(["Dad", "Mom"])
plt.xlabel("Normalized Parent Height")
plt.ylabel("Normalized Child Height")
plt.title("Normalized Parent's Heights Vs. Normalized Child's Height")

plt.show()
```

Normalized Scatterplots



```
{Intercept: } 22.309705489589717
{Coefficients: } {[0.37989697 0.28321451]}
```

GUI for multiple variable (2 variables) regression modeling

Conclusion

Overall, our model that we generated is successful in predicting children's heights from their parent's heights. There is a positive relationship between a child's height and the parent's height, and we can observe the relationship through the linear regression equations below.

Dad's Linear Regression Equation:
$y = 0.3994x_1 + 39.1104$

Mom's Linear Regression Equation:
$y = 0.3132x_1 + 46.6908$

Both parent's height as variables, Linear Regression Equation:
$y = 0.3799x_1 + 0.2832x_2 + 22.3097$

The results from the model and the graphs indicate that children of shorter parents are usually smaller than average and that children of taller parents are usually taller than average. We can predict this from our linear regression models. We also observe this from the graphs that are generated by plotting a parent's height against the child's height without doing any statistics at all.

Some improvements to this analysis that can be made include having more data records from families in general. The data is taken from families in Great Britain, we should also include having more data from different populations across the world to have a more comprehensive view. Another factor that we should think about is that this data is old and that new generations of humans will be taller in general.

Python Program

Two related programs in separate files are in a compressed file. The first program is for single variable linear regression analysis and the second program is for multiple variable linear regression analysis.