Case Study 11 Notebook: Multiple Linear Regression

This Case Study contains analyses to answer the question:

Is there a linear association between height and weight?

Specifically, **for the sample**, we will address

- Is there a linear association between height and weight in a random sample of healthy adults?
- Is the linear relationship between height and weight different for males and females of different age groups in a random sample of healthy adults?

In this analysis we will examine the height and weight of a random sample of 487 healthy males and females of three different age groups:

- under 30
- 30-39
- 40 and over

The, **for the population**, we will explore whether these findings hold in a larger population of ALL healthy adults. So, IF we were to fit a multiple linear regression model, predicting the **weight** of a healthy adult (from the whole population) given **height**, **sex**, and **age group**, we would further like to answer the following questions.

- Is there sufficient evidence to suggest that the population slope for height is non-zero in this model?
- Is there sufficient evidence to suggest that the population slope for sex is non-zero in this model?
- Is there sufficient evidence to suggest that the population slope for age_group is non-zero in this model?

We will again use the statsmodels package that we first saw in Case Study 10.

Imports

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

Preparing the Data and Initial Summary Analyses

We will first read our bdims.csv dataset. There are no missing values represented in this dataset.

In [2]:

```
df_full = pd.read_csv('bdims.csv', sep=',')
df_full.head()
```

Out[2]:

biacromial_diameter	pelvic_breadth	bitrochanteric_diameter	chest_depth	chest_diameter	elbow_diamete

	biacromial_diameter	pelvic_breadth	bitrochanteric_diameter	chest_depth	chest_diameter	elbow_diamete
0	42.9	26.0	31.5	17.7	28.0	13
1	43.7	28.5	33.5	16.9	30.8	14.
2	40.1	28.2	33.3	20.9	31.7	13.
3	44.3	29.9	34.0	18.4	28.2	13.
4	42.5	29.9	34.0	21.5	29.4	15.

5 rows × 26 columns

In [3]:

```
df_full.shape
```

Out[3]:

```
(487, 26)
```

In [4]:

```
df_full.columns
```

Out[4]:

```
Index(['biacromial_diameter', 'pelvic_breadth', 'bitrochanteric_diameter',
       'chest_depth', 'chest_diameter', 'elbow_diameter', 'wrist_diameter',
       'knee_diameter', 'ankle_diameter', 'shoulder_girth', 'chest_girth',
       'waist_girth', 'navel_girth', 'hip_girth', 'thigh_girth', 'bicep_girth',
       'forearm_girth', 'knee_diameter.1', 'calf_girth', 'ankle_girth',
       'wrist_girth', 'age', 'weight', 'height', 'sex', 'age_group'],
      dtype='object')
```

We only plan to use a few variables from this dataframe, so let's create a smaller dataframe to enable faster/easier processing.

Specifically, we will only be examining the following attributes:

- the weights,
- the heights,
- the sex,
- the age group,
- the waist girth, and
- the elbow diameter.

In [5]:

```
df=df_full[['weight', 'height', 'sex', 'age_group', 'waist_girth', 'elbow_diameter']]
```

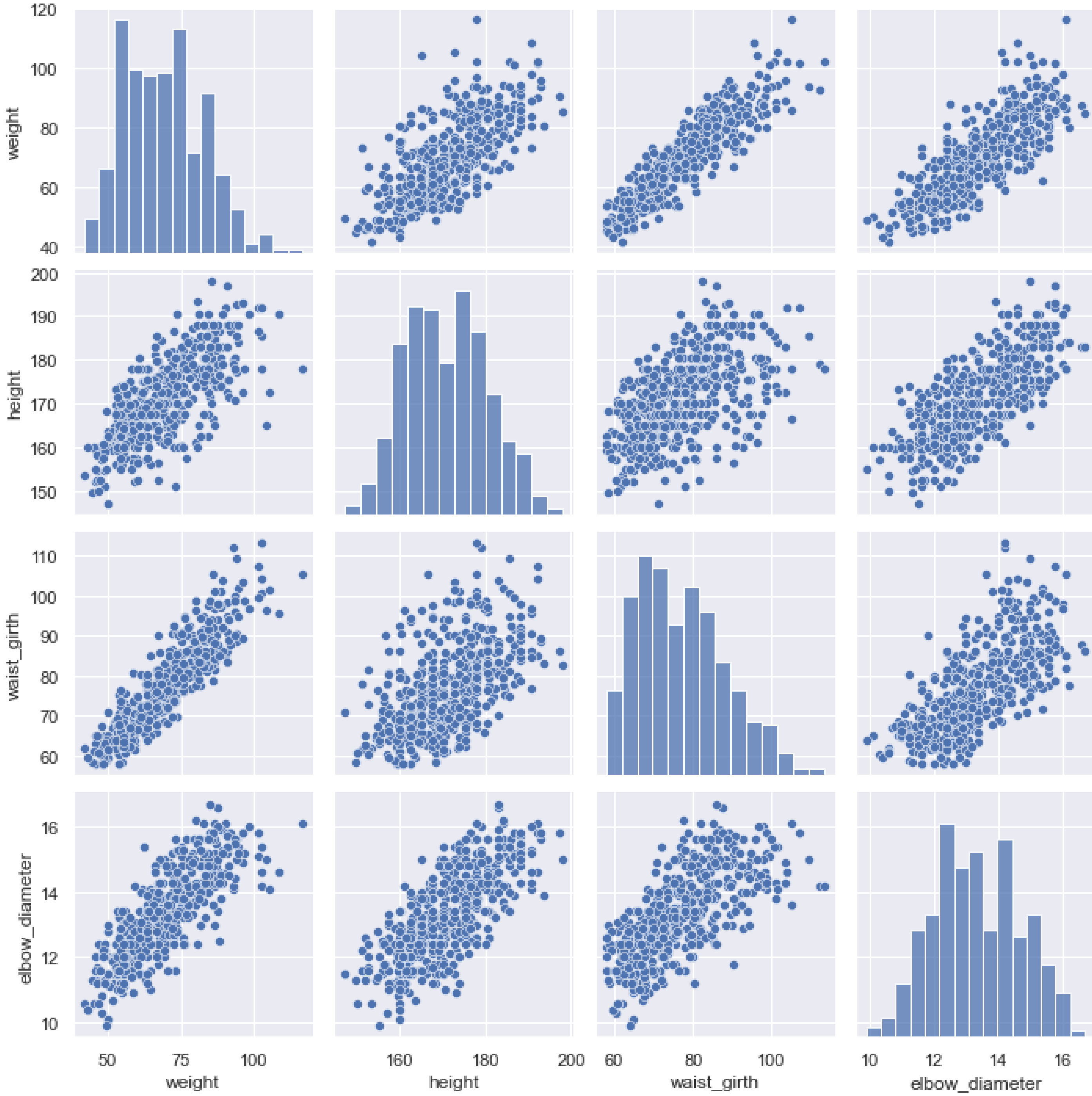
Initial Summary Visualizations

Now that our data is prepared, we can perform some preliminary visualizations and descriptions of the variables that we are interested in.

Let's first examine the pairwise relationships of each pair of numerical variables in the dataframe as well as the histogram of each numerical variable.

In [6]:

```
sns.pairplot(df)
plt.show()
```



Initial Summary Statistics

We can see that all pairs of numerical variables have linear relationships with each other. Therefore, we can use the correlation coefficient R to evaluate the strength and direction of each of these pairs of numerical variables.

In [7]:
```
df.corr()
```

Out[7]:

	weight	height	waist_girth	elbow_diameter
weight	1.000000	0.719449	0.905462	0.803470
height	0.719449	1.000000	0.558367	0.735508
waist_girth	0.905462	0.558367	1.000000	0.703589
elbow_diameter	0.803470	0.735508	0.703589	1.000000

Out of all pairs of numerical variables, it looks like **weight** and **waist girth** (correlation coefficient of 0.905) have the strongest association and **height** and **waist girth** have the weakest relationship (correlation coefficient of 0.558).

We will then begin examining the variables individually.

To start, we will generate a series of summary statistics about each of the quantitative variables individually.

In [8]:

```
df.describe()
```

Out[8]:

	weight	height	waist_girth	elbow_diameter
count	487.000000	487.000000	487.000000	487.000000
mean	68.947023	170.913963	76.896099	13.341273
std	13.455621	9.451632	11.151494	1.350339
min	42.000000	147.200000	57.900000	9.900000
25%	58.200000	163.350000	67.900000	12.400000
50%	67.900000	170.200000	75.700000	13.200000
75%	78.700000	177.800000	84.500000	14.300000
max	116.400000	198.100000	113.200000	16.700000

We can also examine each of the categorical variables individually and jointly.

In [9]:

```
df['sex'].value_counts()
```

Out[9]:

```
Female    260
Male      227
Name: sex, dtype: int64
```

In [10]:

```
df['age_group'].value_counts()
```

Out[10]:

```
under_30        277
30-39           118
40 and above     92
Name: age_group, dtype: int64
```

In [11]:

```
pd.crosstab(df['sex'], df['age_group'])
```

Out[11]:

age_group	30-39	40 and above	under_30
sex			
Female	61	37	162
Male	57	55	115

It looks the most represented age group is those that are under 30.

These basic analyses about the whole dataset are useful to get a sense as to the nature of the data that we are working with. However, in this specific analysis, we suspect that the **weight** of healthy adults is influenced by **height**. Thus we consider **weight** to be the response variable and **height** to be an explanatory varaible.

In addition, we would like to see how/if this relationship changes based on sex and age_group. Thus we will consider **sex** and **age_group** to also be explanatory variables.

Because we have a specific set of questions in mind that we would like to answer about the sample, we should be thoughtful about which visualizations and summary statistics will best help us answer these questions.

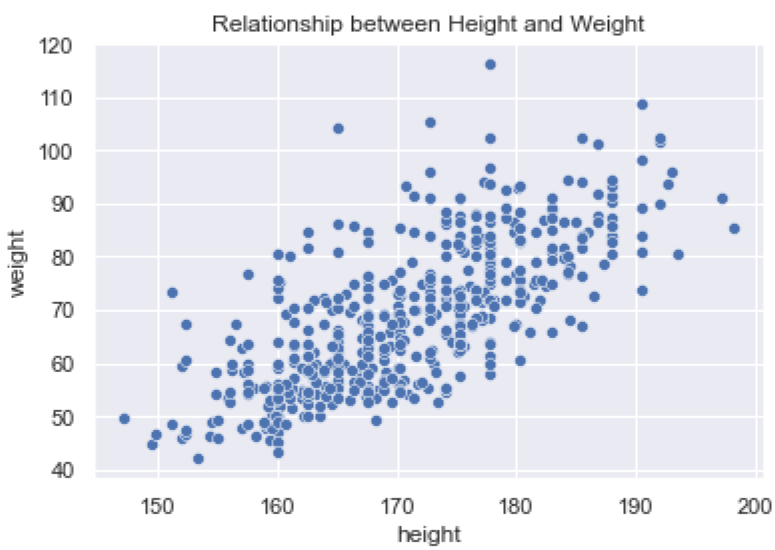
Relationships between the response variable and each explanatory variable *individually*.

Now, let's examine the relatioship between weight (the response variable) and each of our explanatory variables *individually*.

Weight and height.

In [12]:

```
sns.scatterplot(x="height",y='weight', data=df)
plt.title('Relationship between Height and Weight')
plt.show()
```



By looking at the scatterplot above, the association between height and weight in the sample is positive, linear, moderately strong, and there does not seem to be any obvious outliers.

Because the relationship is linear, we can use the correlation coefficient to quantify the strength and direction of the relationship.

In [13]:

```
df[['height','weight']].corr()
```

Out[13]:

	height	weight
height	1.000000	0.719449
weight	0.719449	1.000000

Thus, a relatively high correlation coefficient R=0.719, further validates that there is a moderately strong linear relationship between height and weight in the sample.

Weight and Sex

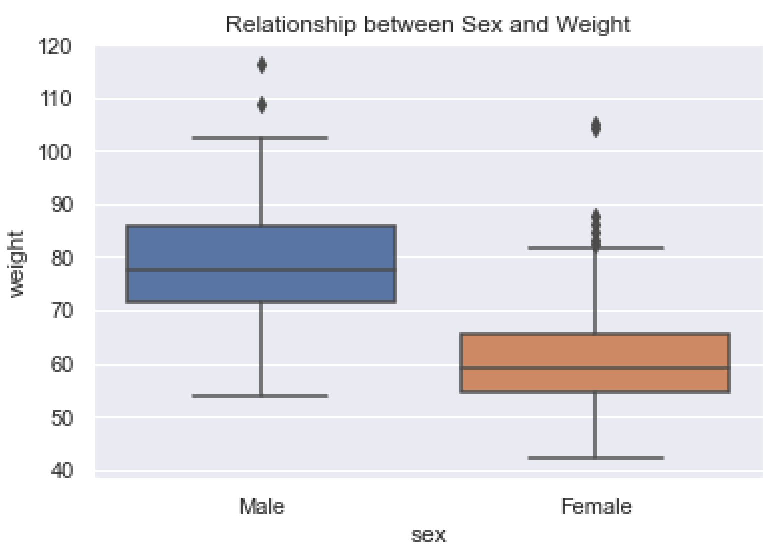
In [14]:

```
sns.violinplot(x="sex", y='weight', data=df)
```

```
plt.title('Relationship between Sex and Weight')
plt.show()
```

In [15]:

```
sns.boxplot(x="sex", y='weight', data=df)
plt.title('Relationship between Sex and Weight')
plt.show()
```



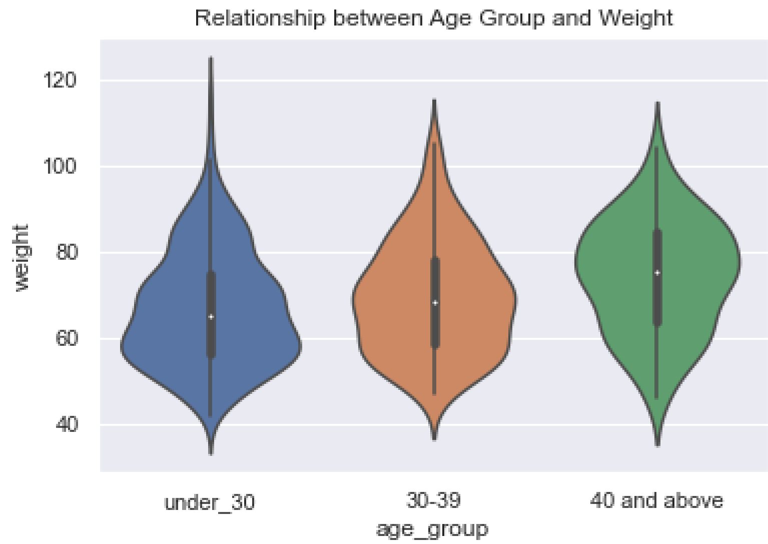
Remember, there are always four things that we should be ready to compare when discussing the association between a categorical variable (sex) and a numerical variable (weight).

1. Compare Measures of Center: We can see that the median male weight is higher than the median female weight.
2. Compare Measures of Spread: The IQR and range of male and females weight is about the same.
3. Compare Distribution Shapes: The male weight distribution looks slightly bimodal, and is slightly right skewed. The female weight distribution is unimodal, and more skewed to the right.
4. Compare Outliers: Both distributions have high outliers. The female distribution has a few more outliers than the male distribution.

Weight and Age Group

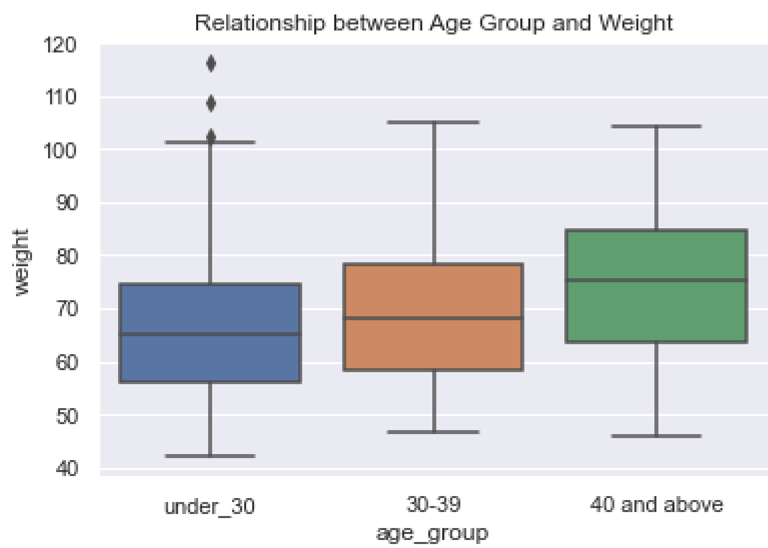
In [16]:

```
sns.violinplot(x="age_group", y='weight', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



In [17]:

```
sns.boxplot(x="age_group", y='weight', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



1. Compare Measures of Center: The median weight increase as the age group increases.
2. Compare Measures of Spread: The IQR and range of of the age groups is about the same.
3. Compare Distribution Shapes: Weight distributions look mostly unimodal. The under 30 and 30-39 age groups are slightly right skewed and the 40 and over age group is slightly left skewed.
4. Compare Outliers: Only the under 30 age group has outliers. These outliers are high.

Relationships between the response variable and two explanatory variables.

It can be helpful to incorporate more than two variables into a graph. This is easier to do if at least one of the variables is categorical. Below, we consider adding variables into our original scatterplot to help us understand more about the relationship between different sets of variables.

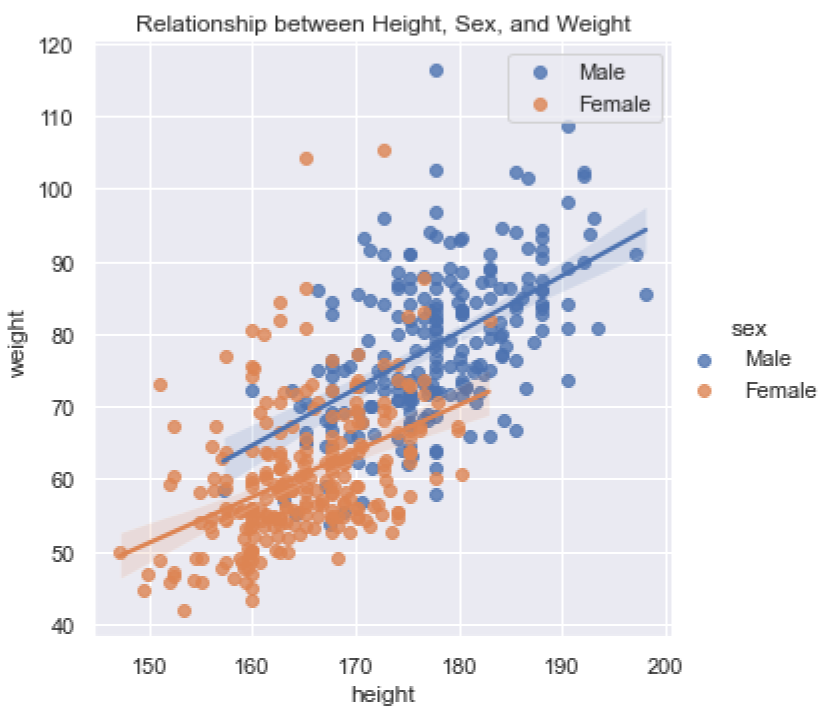
Relationship betweeen Height and Weight for Males and Females.

We can use the **sns.scatterplot()** function to plot the relationship between two numerical variables (using the **x and y parameters**), and then we can color-code the points based on another variable (using the **hue parameter**).

In [18]:
```
sns.scatterplot(x="height",y='weight', hue='sex', data=df)
plt.title('Relationship between Height, Sex, and Weight')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

If we want to draw a best fit line, for each of the levels represented by the **hue parameter** then we can use the **sns.lmplot()** function.

In [19]:
```
sns.lmplot(x="height",y='weight', hue='sex', data=df)
plt.title('Relationship between Height, Sex, and Weight')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```



- Intercept Comparison: It looks like the intercept of the male best fit line is higher than the intercept for the female best fit line.
- Slope Comparison: It looks like the slope of the male best fit line is slightly higher than the slope for the female best fit line.

If we would like to directly calculate a summary statistic on a subset of observations (where each subset corresponds to each level of a given categorical variable), then we can use the **.groupby()** function.

For instance, the code below calculates the correlation between weight and height for each level of the **sex** variable.

In [20]:
```
df[['sex', 'height', 'weight']].groupby(['sex']).corr()
```

Out[20]:

sex		height	weight
Female	**height**	1.000000	0.431059
	weight	0.431059	1.000000
Male	**height**	1.000000	0.536599
	weight	0.536599	1.000000

- Compare Correlations: Thus, the strength of the linear relationship of height and weight for males (0.537) is higher than the strength of the relationship of height and weight for females (0.431).

Relationship betweeen Height and Weight for the Age Groups

In [21]:
```
sns.scatterplot(x="height",y='weight', hue='age_group', data=df)
plt.title('Relationship between Height, Age Group, and Weight')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

In [22]:
```
sns.lmplot(x="height",y='weight', hue='age_group', data=df, ci=False)
plt.title('Relationship between Height, Weight, and Age Group')
plt.show()
```

- Slope Comparison: It looks like the slopes of the under 30 age group and 40 and above age groups are almost parallel, whereas the 30-39 age group slope is slightly smaller.

In [23]:
```
df.groupby(['age_group']).corr()
```

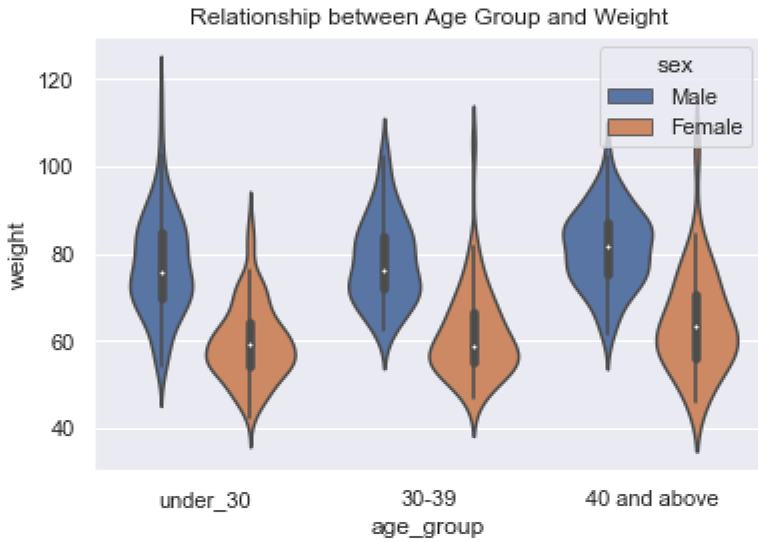
Out[23]:

age_group		weight	height	waist_girth	elbow_diameter
30-39	weight	1.000000	0.649175	0.916938	0.733133
	height	0.649175	1.000000	0.497285	0.719191
	waist_girth	0.916938	0.497285	1.000000	0.638171
	elbow_diameter	0.733133	0.719191	0.638171	1.000000
40 and above	weight	1.000000	0.699453	0.904493	0.776830
	height	0.699453	1.000000	0.581820	0.757793
	waist_girth	0.904493	0.581820	1.000000	0.671320
	elbow_diameter	0.776830	0.757793	0.671320	1.000000
under_30	weight	1.000000	0.768051	0.921540	0.825232
	height	0.768051	1.000000	0.632940	0.750642
	waist_girth	0.921540	0.632940	1.000000	0.739523
	elbow_diameter	0.825232	0.750642	0.739523	1.000000

- Compare Correlations: The strength of the linear relationship between height and weight is highest for those under 30 (0.768) and is lowest for those that are 30-39.

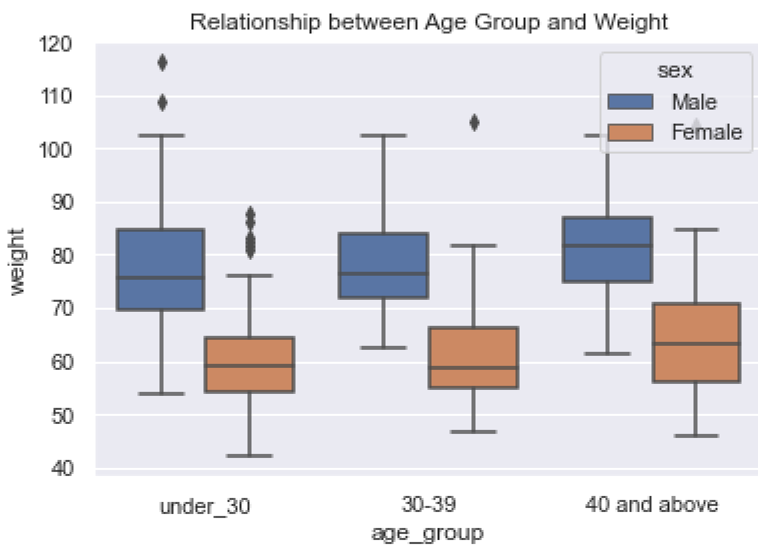
Relationship betweeen Weight and Sex by Age Group

In [24]:
```
sns.violinplot(x="age_group", y='weight', hue='sex', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



In [25]:

```
sns.boxplot(x="age_group", y='weight', hue='sex', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



Comparing Measure of Center Changes for Different Sexes:

- The difference in male and female median weight is about the same for all three age groups.

Comparing Measure of Spread Changes for Different Sexes:

- Under 30 males have a higher IQR than under 30 females.
- 40 and over males have a smaller IQR than 40 and over females.

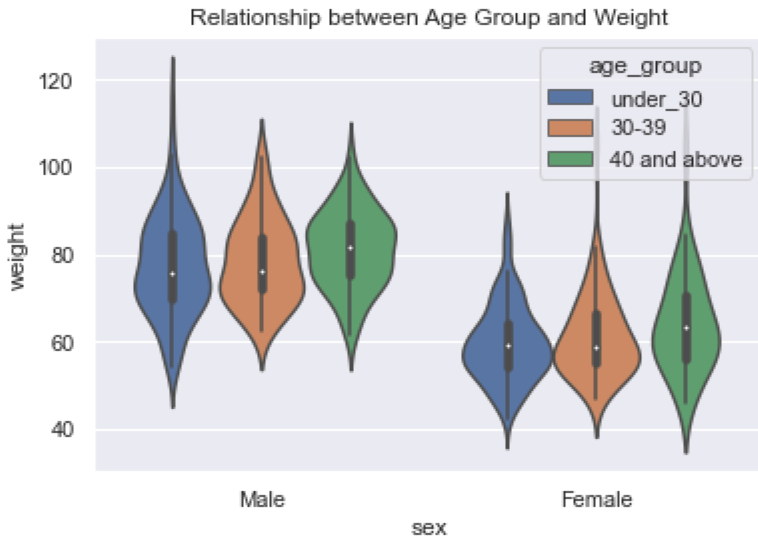
Comparing Shape Changes for Different Sexes:

- The skew for under 30 males is more right skewed than the skew for under 30 females. All distributions are unimodal.
- The skew for 30-39 and 40 and over males is less right skewed than skew for 30-39 and 40 and over females. All distributions are unimodal.

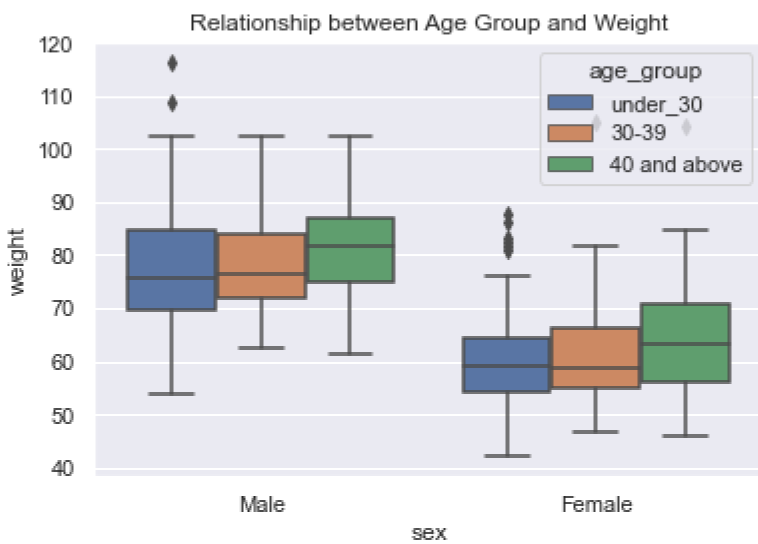
Comparing Outliers Changes for Different Sexes:

- For all three age groups, the females have more outliers.

In [26]:
```
sns.violinplot(x="sex", y='weight', hue='age_group', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



In [27]:
```
sns.boxplot(x="sex", y='weight', hue='age_group', data=df)
plt.title('Relationship between Age Group and Weight')
plt.show()
```



Comparing Measure of Center Changes as Age Increases:

- The weight median for males increase as the age group increases.
- The weight median for females increases as the age group increases.

Comparing Measure of Spread Changes as Age Increases:

- The weight spread for males decreases as the age group increases.
- The weight spread for females increases as the age group increases.

Comparing Shape Changes as Age Increases:

- The skew for males becomes less right skewed as the age increases. All distributions are unimodal.
- The skew for females becomes more right skewed as the age increases. All distributions are unimodal.

Comparing Outliers Changes as Age Increases:

- Only males under 30 have weight outliers.
- Only females under 30 have weight outliers.

Relationship between the response variable and three explanatory variables

We can also use the **sns.scatterplot()** function to plot the relationship between two numerical variables (using the **x and y parameters**), and then we can:

- color-code the points based on another variable (using the **hue parameter**), and
- code the marker size by another variable (using the **size parameter**).

In [28]:

```
sns.scatterplot(x="height",y='weight', hue='age_group', size='sex', data=df)
plt.title('Relationship between Height, Age Group, and Weight')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

We can also use the **sns.scatterplot()** function to plot the relationship between two numerical variables (using the **x and y parameters**), and then we can:

- color-code the points based on another variable (using the **hue parameter**), and
- code the marker style by another variable (using the **style parameter**).

In [29]:

```
sns.scatterplot(x="height",y='weight', hue='age_group', style='sex', data=df)
plt.title('Relationship between Height, Age Group, and Weight')
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

If we want to draw a best fit line (breaking the data down by two or more categorical variables), we can use the **sns.lmplot()** and specify variables names for the:

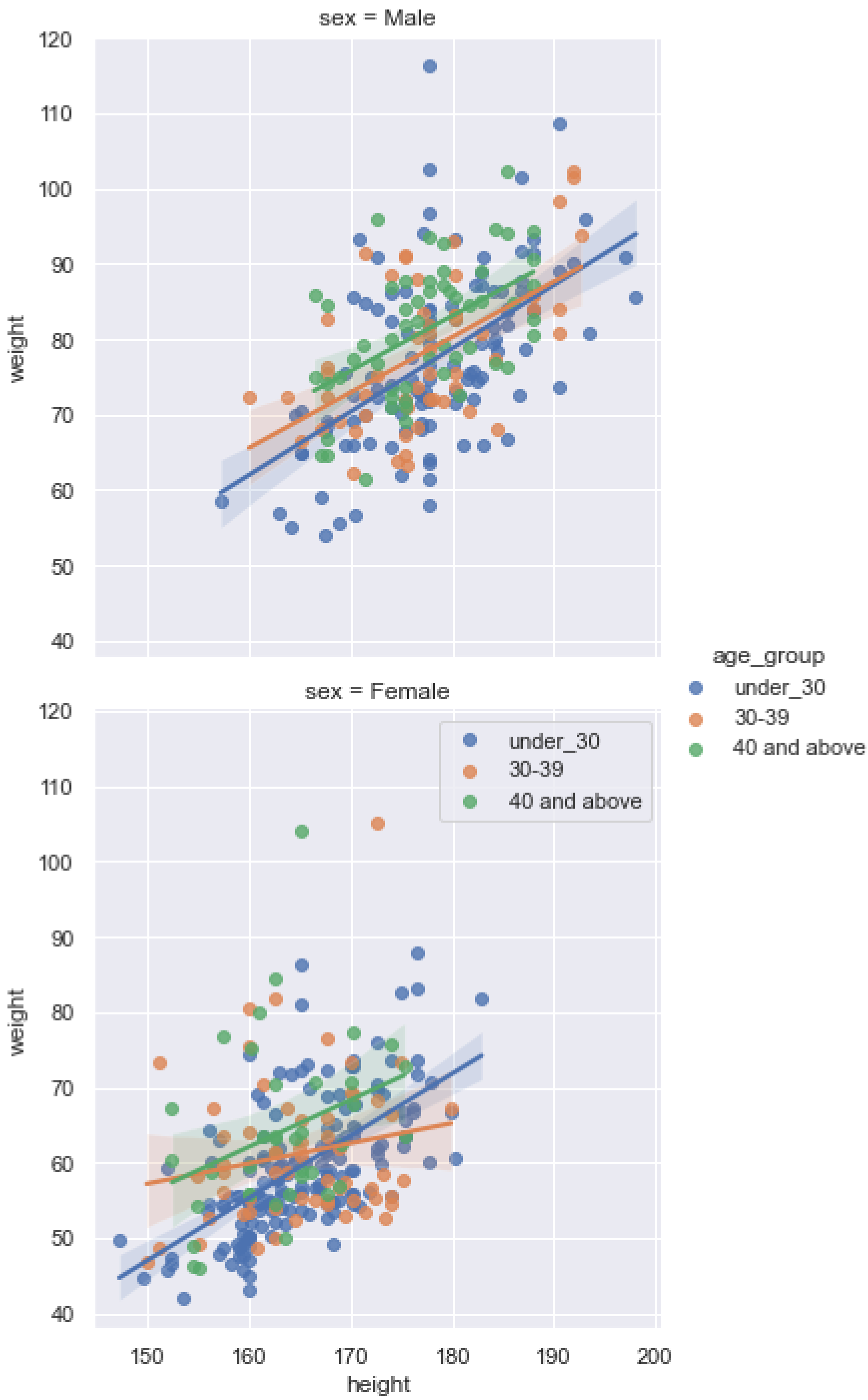
- col parameter and,
- row parameter.

In [30]:

```
sns.lmplot(x="height",y='weight', hue='age_group', col='sex', data=df)
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```

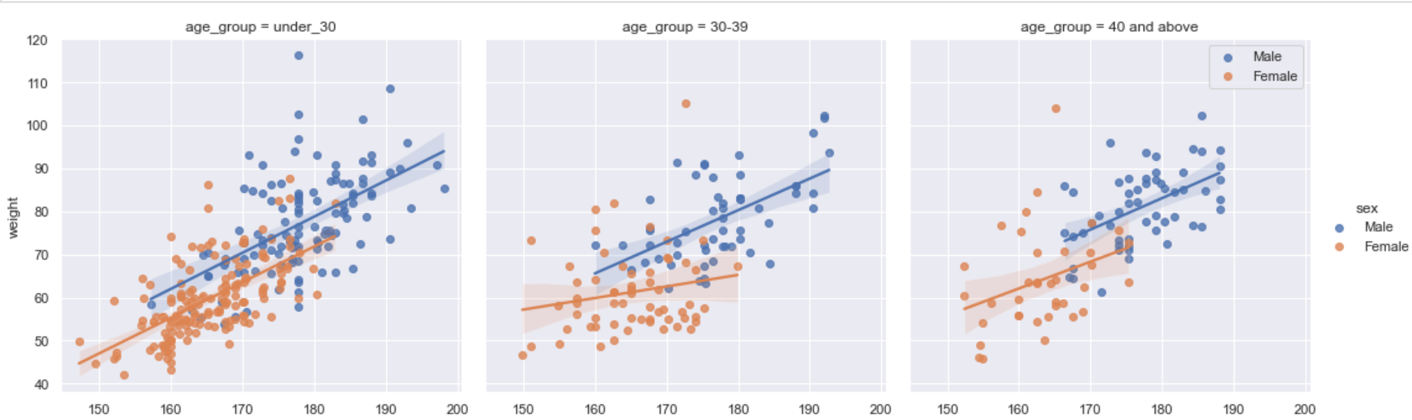
In [31]:

```
sns.lmplot(x="height",y='weight', hue='age_group', row='sex', data=df)
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```



In [32]:

```
sns.lmplot(x="height",y='weight', hue='sex', col='age_group', data=df)
plt.legend(bbox_to_anchor=(1,1))
plt.show()
```



In [33]:

```
df.groupby(['sex','age_group']).corr()
```

Out[33]:

sex	age_group		weight	height	waist_girth	elbow_diameter
Female	30-39	weight	1.000000	0.182259	0.884331	0.408199
		height	0.182259	1.000000	0.005927	0.350111
		waist_girth	0.884331	0.005927	1.000000	0.277085
		elbow_diameter	0.408199	0.350111	0.277085	1.000000
	40 and above	weight	1.000000	0.326263	0.860214	0.695378
		height	0.326263	1.000000	0.000972	0.389143
		waist_girth	0.860214	0.000972	1.000000	0.487145
		elbow_diameter	0.695378	0.389143	0.487145	1.000000
	under_30	weight	1.000000	0.613888	0.848198	0.680492
		height	0.613888	1.000000	0.322365	0.499629
		waist_girth	0.848198	0.322365	1.000000	0.468677
		elbow_diameter	0.680492	0.499629	0.468677	1.000000
Male	30-39	weight	1.000000	0.569800	0.814738	0.509941
		height	0.569800	1.000000	0.208744	0.507973
		waist_girth	0.814738	0.208744	1.000000	0.197176
		elbow_diameter	0.509941	0.507973	0.197176	1.000000
	40 and above	weight	1.000000	0.521314	0.798862	0.524962
		height	0.521314	1.000000	0.209410	0.536937
		waist_girth	0.798862	0.209410	1.000000	0.239805
		elbow_diameter	0.524962	0.536937	0.239805	1.000000
	under_30	weight	1.000000	0.553887	0.865876	0.648278
		height	0.553887	1.000000	0.296707	0.468378
		waist_girth	0.865876	0.296707	1.000000	0.421885
		elbow_diameter	0.648278	0.468378	0.421885	1.000000

- Comparing Correlations:
  - 30-39 females have the weakest relationship of height and weight (R=0.182)
  - 30-39 males have the strongest relationship of height and weight (R=0.570)
- Comparing Slopes:
  - 30-39 females have the smallest best fit line slope modeling the relationship of height and weight
  - the slopes of all other sex and age groups are more similar.

Fitting & Interpreting a Multiple Linear Regression Model

Let's create our multiple linear regression equation with weight as a response variable and height, sex, and age_group as explanatory variables.

We can use the same function and format as simple linear regression equations.

- The response variable goes on the left part of the "equation" in the **smf.ols()** function.
- The explanatory variables go on the right part of the "equation" in the **smf.ols()** function
  - Each explanatory variable is separated by a **+**.
  - The **smf.ols()** function automatically creates indicator variables when it detects a categorical explanatory variable.

This step fits the model and creates an object containing the results.

In [34]:
```
results = smf.ols('weight ~ height+sex+age_group', data=df).fit()
```

In [35]:
```
results.summary()
```

Out[35]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.594
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	176.1
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	7.71e-93
Time:	12:39:49	**Log-Likelihood:**	-1737.1
No. Observations:	487	**AIC:**	3484.
Df Residuals:	482	**BIC:**	3505.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-59.0102	9.409	-6.272	0.000	-77.498	-40.523
sex[T.Male]	7.9128	1.086	7.286	0.000	5.779	10.047
age_group[T.40 and above]	3.4030	1.202	2.830	0.005	1.041	5.765
age_group[T.under_30]	-1.7432	0.949	-1.838	0.067	-3.607	0.121
height	0.7291	0.057	12.821	0.000	0.617	0.841

Omnibus:	91.450	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	182.551
Skew:	1.032	**Prob(JB):**	2.29e-40
Kurtosis:	5.176	**Cond. No.**	4.14e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Formulating the multiple linear regression line

What regression line is fit by Python?

$$\hat{y} = -59.0102 + 7.9128 sex[T.Male] + 3.4030 age\_group[T.40andabove]$$
$$- 1.7432 age\_group[T.under\_30] + 0.7291 height$$

In other words,

$$\hat{y} = -59.0102 + 7.9128 Male + 3.4030 age[40andabove] - 1.7432 age[under30]$$
$$+ 0.7291 height$$

Interpreting the Intercept

- **We would <u>expect the average</u> weight of a female in her 30's that is 0 cm tall to be -59.0102 kg.**
  - This is a nonsensical answer though because:
    - You can't have a person that is 0 cm tall and
    - You can't have a person that has a negative weight.

Interpreting the Slopes

- <u>All else held equal</u>, if we were to increase the height of a healthy adult by 1 cm, then we would <u>expect</u> the weight to increase, <u>on average</u>, by 0.7291 kg.
- <u>All else held equal</u>, we would <u>expect the average</u> healthy adult male weight to be 7.9128 kg higher than healthy adult females.
- <u>All else held equal</u>, we would <u>expect the average</u> weight of healthy adults under 30 to be 1.7432 kg lower than healthy adults in their 30's.
- <u>All else held equal</u>, we would <u>expect the average</u> weight of healthy adults at least 40 to be 3.4030 kg higher than healthy adults in their 30's.

Inference Conditions for Multiple Linear Regression Intercept and Slopes

For this section, suppose now we also wanted to add ‘elbow diameter’ to our list of explanatory variables.

Fitting the Model

We will start by fitting a multiple linear regression model predicting weight with height, elbow diameter, sex, and age group.

In [36]:

```
results = smf.ols('weight ~ height+ elbow_diameter+sex+age_group', data=df).fit()
results.summary()
```

Out[36]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.688
Model:	OLS	**Adj. R-squared:**	0.685
Method:	Least Squares	**F-statistic:**	212.0
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	4.04e-119
Time:	12:39:49	**Log-Likelihood:**	-1672.9
No. Observations:	487	**AIC:**	3358.

Df Residuals:	481	**BIC:**	3383.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-75.1198	8.363	-8.982	0.000	-91.553	-58.686
sex[T.Male]	0.7195	1.125	0.640	0.523	-1.490	2.929
age_group[T.40 and above]	1.2766	1.070	1.194	0.233	-0.825	3.378
age_group[T.under_30]	-1.4391	0.833	-1.728	0.085	-3.075	0.197
height	0.4139	0.056	7.345	0.000	0.303	0.525
elbow_diameter	5.5146	0.458	12.043	0.000	4.615	6.414

Omnibus:	61.496	**Durbin-Watson:**	1.925
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	93.358
Skew:	0.833	**Prob(JB):**	5.34e-21
Kurtosis:	4.352	**Cond. No.**	4.21e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.21e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Summarizing the Strength of the Model

What is the R^2 of this model?

$R^2 = 0.688$

In [37]:

```
results.rsquared
```

Out[37]:

```
0.6878417369562507
```

Checking Conditions

<u>Condition 1</u>: Linearity Condition

Because the distribution of points in the plot below are roughly evenly distributed above and below the line as we move from left to right, we can say the linearity condition is met.

In [38]:

```
sns.regplot(x=results.fittedvalues, y=results.resid, ci=None)
plt.ylabel('Residual')
plt.xlabel('Fitted Value')
plt.show()
```

<u>Condition 2</u>: Constant Variability of Residuals Condition

Because the y-axis spread of points in the plot below slightly change as we move from left to right, we can say that this condition is slightly not met.

In [39]:

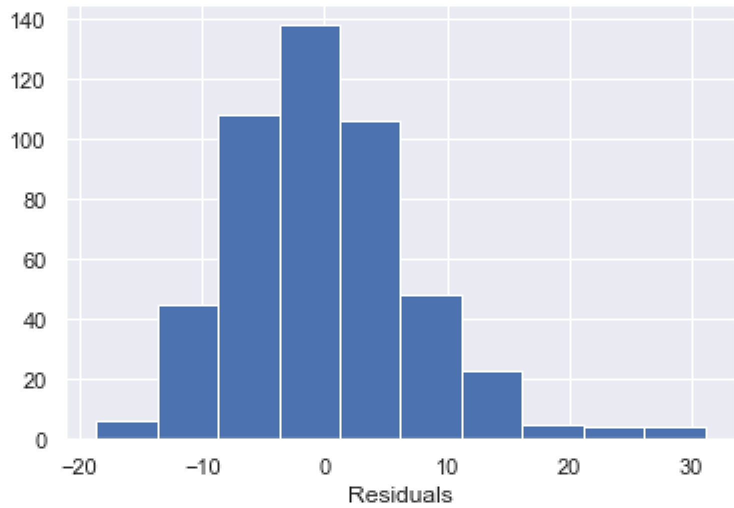
```
sns.regplot(x=results.fittedvalues, y=results.resid, ci=None)
plt.ylabel('Residual')
plt.xlabel('Fitted Value')
plt.show()
```

<u>Condition 3</u>: Normality of Residuals (with Mean of 0) Condition

Because the histogram of residuals is slightly skewed to the right, the assumption that the residuals are normally distributed is slightly not met. (However, it does look like the mean is about 0).

In [40]:

```
plt.hist(results.resid)
plt.xlabel('Residuals')
plt.show()
```



Condition 4 Independence of Residuals Condition

At the very least, we verify that:

- the data is randomly sampled and
- the sample size n=487<10% of all healthy adults

Thus the condition for independence of residuals may not be violated in this particular way.

However, it may still be the case that these residuals are not independent (for other reasons that you will discuss in later statistics classes).

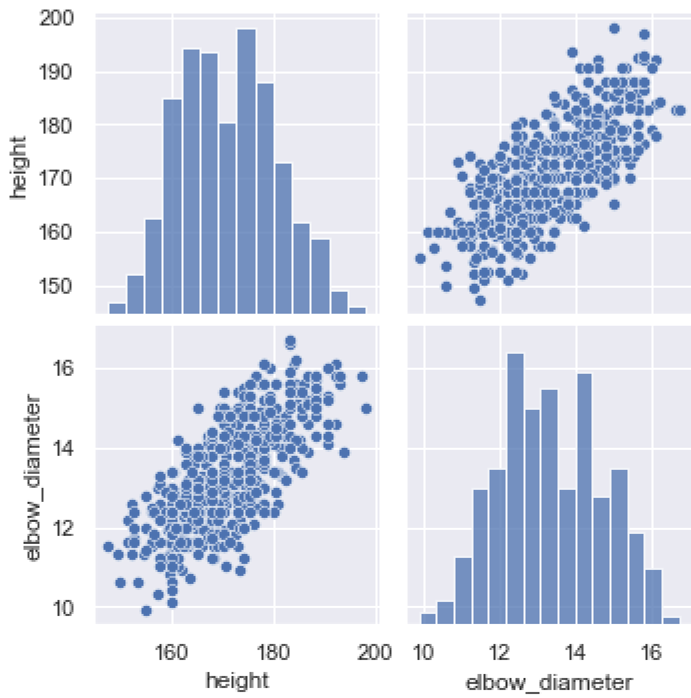
Condition 5: No Multicollinearity Condition

Let's take a look at the relationships between each pair of numerical explanatory variables. (We only have two in this example, but the **sns.pairplot()** can be useful when you have lots of numerical explanatory variables).

We see that there is a pretty strong linear relationship (R=0.736) between the explanatory variables height and elbow_diameter. **Thus the multicollinearity condition is violated**.

Leaving in both of these explanatory variables may lead to biased estimates for the slopes.

In [41]:
```
sns.pairplot(df[['height','elbow_diameter']])
plt.show()
```



In [42]:
```
df[['height','elbow_diameter']].corr()
```

Out[42]:

	height	elbow_diameter
height	1.000000	0.735508
elbow_diameter	0.735508	1.000000

If we were to delete one of these numerical explanatory variables (because of the multicollinearity condition being violated), which one would you choose.

Let's try deleting both variables (one at a time) and see which resulting model has a higher R^2 (ie. more explanatory power).

In [43]:
```
results = smf.ols('weight ~ height+sex+age_group', data=df).fit()
print('R^2 for the model without elbow_diameter:',results.rsquared)
```

```
R^2 for the model without elbow_diameter: 0.5937143871724586
```

In [44]:
```
results = smf.ols('weight ~ elbow_diameter+sex+age_group', data=df).fit()
print('R^2 for the model without height:',results.rsquared)
```

```
R^2 for the model without height: 0.6528321666723597
```

If you were particularly interested in exploring the relationship between weight, height, sex, and age, then you would want to delete the height variable. This is because the R^2 of the model without height is higher, and thus can explain more of the variability of weight.

However, we also want to consider the context of our model. Because we knew ahead of time that we were interested in exploring the relationship between weight, height, sex, and age, we will delete elbow_diameter and keep height.

Re-fit the model and re-check the conditions

Now, we will fit a multiple linear regression model predicting weight with the following explanatory variables and check the conditions:

a. Height

b. ~~Elbow diameter~~

c. Sex

d. Age group.

In [45]:

```
results = smf.ols('weight ~ height+sex+age_group', data=df).fit()
results.summary()
```

Out[45]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.594
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	176.1
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	7.71e-93
Time:	12:39:49	**Log-Likelihood:**	-1737.1
No. Observations:	487	**AIC:**	3484.
Df Residuals:	482	**BIC:**	3505.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-59.0102	9.409	-6.272	0.000	-77.498	-40.523
sex[T.Male]	7.9128	1.086	7.286	0.000	5.779	10.047
age_group[T.40 and above]	3.4030	1.202	2.830	0.005	1.041	5.765
age_group[T.under_30]	-1.7432	0.949	-1.838	0.067	-3.607	0.121
height	0.7291	0.057	12.821	0.000	0.617	0.841

Omnibus:	91.450	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	182.551
Skew:	1.032	**Prob(JB):**	2.29e-40
Kurtosis:	5.176	**Cond. No.**	4.14e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

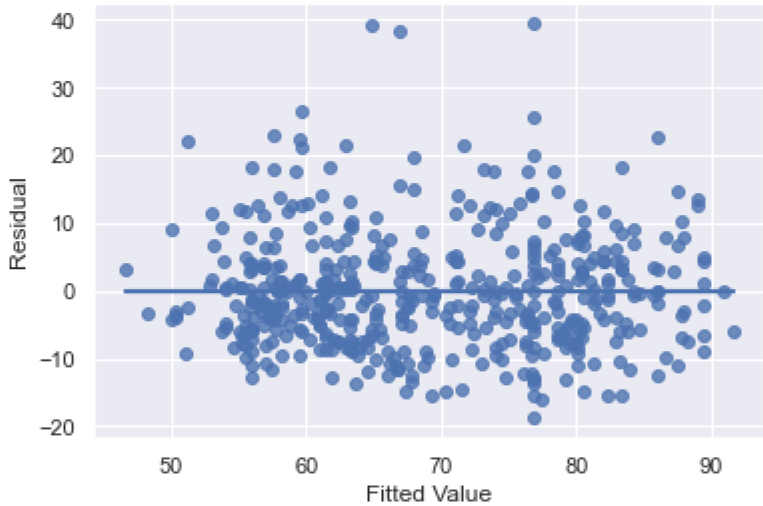
Condition 1: Linearity Condition

Because the distribution of points in the plot below are roughly evenly distributed above and below the line as we move from left to right, we can say the linearity condition is met.

In [46]:

```
sns.regplot(x=results.fittedvalues, y=results.resid, ci=None)
plt.ylabel('Residual')
```

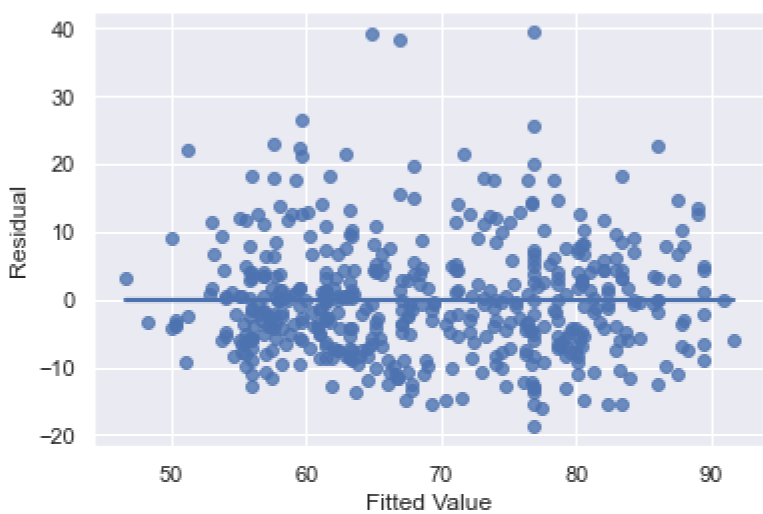
```
plt.xlabel('Fitted Value')
plt.show()
```



<u>Condition 2</u>: Constant Variability of Residuals Condition

Because the y-axis spread of points in the plot below doesn't really change as we move from left to right, we can say that this condition is met.

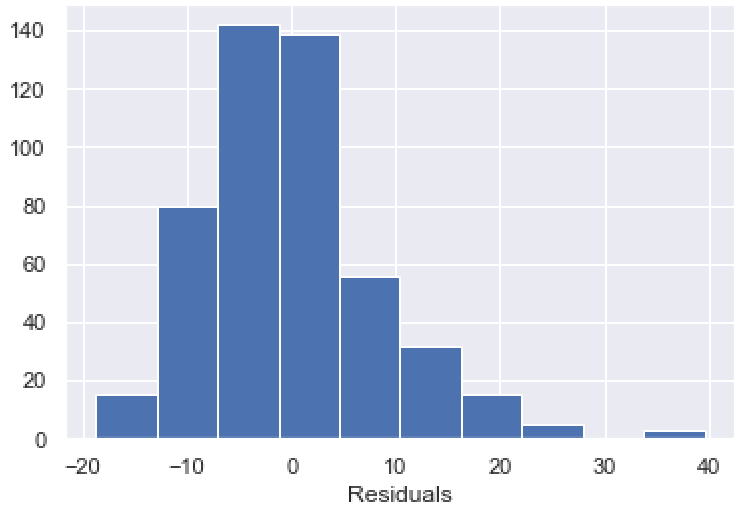
In [47]:
```
sns.regplot(x=results.fittedvalues, y=results.resid, ci=None)
plt.ylabel('Residual')
plt.xlabel('Fitted Value')
plt.show()
```



<u>Condition 3</u>: Normality of Residuals (with Mean of 0) Condition

Because the histogram of residuals is slightly skewed to the right, the assumption that the residuals are normally distributed is slightly not met. (However, it does look like the mean is about 0).

In [48]:
```
plt.hist(results.resid)
plt.xlabel('Residuals')
plt.show()
```



Condition 4 Independence of Residuals Condition

At the very least, we verify that:

- the data is randomly sampled and
- the sample size n=487<10% of all healthy adults

Thus the condition for independence of residuals may not be violated in this particular way.

However, it may still be the case that these residuals are not independent (for other reasons that you will discuss in later statistics classes).

Condition 5: No Multicollinearity Condition

This new model only has one numerical explanatory variable, height. So height will not be collinear with another numerical variable. Thus this condition is met.

Inference for a Single Multiple Linear Regression Slope

Confidence Interval for a Single Multiple Linear Regression Slope

Create a 90% confidence interval for the height slope in the model above and interpret it.

In [49]:

```
results = smf.ols('weight ~ height+sex+age_group', data=df).fit()
results.summary()
```

Out[49]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.594
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	176.1
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	7.71e-93
Time:	12:39:49	**Log-Likelihood:**	-1737.1
No. Observations:	487	**AIC:**	3484.
Df Residuals:	482	**BIC:**	3505.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-59.0102	9.409	-6.272	0.000	-77.498	-40.523
sex[T.Male]	7.9128	1.086	7.286	0.000	5.779	10.047
age_group[T.40 and above]	3.4030	1.202	2.830	0.005	1.041	5.765
age_group[T.under_30]	-1.7432	0.949	-1.838	0.067	-3.607	0.121
height	0.7291	0.057	12.821	0.000	0.617	0.841

Omnibus:	91.450	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	182.551
Skew:	1.032	**Prob(JB):**	2.29e-40
Kurtosis:	5.176	**Cond. No.**	4.14e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

1. First, because our multiple linear regression inference conditions hold (checked above), our confidence interval will not be making invalid interpretations and conclusions.

2. Creating the confidence interval.

In [50]:

```
point_estimate=0.7291
print('Point Estimate = Sample Slope = ', point_estimate)
```

```
Point Estimate = Sample Slope =  0.7291
```

In [51]:

```
standard_error=0.057
print('Standard Error', standard_error)
```

```
Standard Error 0.057
```

The critical value for this 90% confidence interval is the postive t-score t^* (from the t-distribution with $df = n - p - 1 = 487 - 4 - 1 = 482$) that encapsulates an area of 0.90 between $-t^*$ and t^*.

(Remember, p=4=the number of slopes in the model.)

In [52]:

```
from scipy.stats import t

critical_value=t.ppf(0.95, df=482)
print('Critical Value:', critical_value)
```

```
Critical Value: 1.6480210955444845
```

In [53]:

```
lower_bound=point_estimate-critical_value*standard_error
upper_bound=point_estimate+critical_value*standard_error

print('90% Confidence Interval for the Height Population Slope in the Current Model:', lov
```

```
90% Confidence Interval for the Height Population Slope in the Current Model: 0.6351627975
539643 0.8230372024460356
```

3. Interpret the confidence interval.

We are 90% confident that the height slope in the multiple linear regression population model (that predicts the weight of all healthy adults with height, age group, and sex) is between 0.635 and 0.823.

Conducting a hypothesis test for a single population slope, testing the claim $H_a : \beta_i \neq 0$.

We are interested in testing the claim that the height slope in the multiple linear regression population model (that predicts the weight of all healthy adults with height, age group, and sex) is non-zero.

1. Set up your hypotheses.

$H_0 : \beta_4 = 0$

$H_A : \beta_4 \neq 0$

(β_4 is the population slope that corresponds to height)

2. Check your multiple linear regression inference conditions

We already checked them for this model above, and found that they all hold. Thus the conclusions that we make with our p-value will not be invalid.

3. Find the p-value for this test (in your summary output table).

The summary output table tell us that this p-value<0.0001

4. Calculate the p-value for this test (using the point estimate, the standard error, and the t-distribution).

In [54]:

```
point_estimate=0.7291
print('Point Estimate = Sample Slope = ', point_estimate)

standard_error=0.057
print('Standard Error', standard_error)

null_value=0
print('Null Value:', null_value)

test_stat=(point_estimate-null_value)/standard_error
print('Test Statistic:', test_stat)

pvalue=2*(1-t.cdf(np.abs(test_stat), df=482))
print('p-value', pvalue)
```

```
Point Estimate = Sample Slope =  0.7291
Standard Error 0.057
Null Value: 0
Test Statistic: 12.791228070175437
p-value 0.0
```

5. Make a conclusion with this p-value and a signficance level $\alpha = 0.10$.

Because $p - value < 0.0001 < \alpha = 0.10$, we reject the null hypothesis. Thus we have sufficient evidence to suggest that the height slope β_4 in the multiple linear regression population model (that predicts the weight of all healthy adults with height, age group, and sex) is non-zero.

6. Make a conclusion using the 90% confidence interval that you calculated above.

Because the null value (0) is not in the 90% confidence interval (0.635, 0.823) we reject the null hypothesis. Thus we have sufficient evidence to suggest that the height slope β_4 in the multiple linear regression population model (that predicts the weight of all healthy adults with height, age group, and sex) is non-zero.

Inference for ALL Multiple Linear Regression Slopes

What if instead of focusing on a single slope, we wanted to make a statement about ALL slopes simultaneously? Could we perform this inference?

The answer is **yes**, as long as we have a certain form of hypotheses.

For this case, we do need to introduce a new reference distribution, the F distribution.

F distribution

The F distribution only allows non-negative (0 or positive) values. The distribution is typically described as skewed right. The distribution also has two different degrees of freedom.

For example, calculate the probability that an F-score is less than or equal to 4, (using df1=3 and df2=9).

$P(F_{3,9} \leq 4) = 0.954$

In [55]:

```
from scipy.stats import f
f.cdf(4,  dfn=3,dfd=9)
```

Out[55]:

```
0.954016001798486
```

Conducting the Test $H_0 : \beta_1 - \beta_2 = ... = \beta_p = 0$

For example, when using sex, height, and age_group to predict weight in a linear regression equation, is there significant evidence to suggest that at least one of the slopes in the population linear regression model is non-zero?

In [56]:

```
results.summary()
```

Out[56]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.594
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	176.1
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	7.71e-93
Time:	12:39:50	**Log-Likelihood:**	-1737.1
No. Observations:	487	**AIC:**	3484.
Df Residuals:	482	**BIC:**	3505.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-59.0102	9.409	-6.272	0.000	-77.498	-40.523
sex[T.Male]	7.9128	1.086	7.286	0.000	5.779	10.047

age_group[T.40 and above]	3.4030	1.202	2.830	0.005	1.041	5.765
age_group[T.under_30]	-1.7432	0.949	-1.838	0.067	-3.607	0.121
height	0.7291	0.057	12.821	0.000	0.617	0.841

Omnibus:	91.450	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	182.551
Skew:	1.032	**Prob(JB):**	2.29e-40
Kurtosis:	5.176	**Cond. No.**	4.14e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

1. First set up hypotheses for this test.

$H_0 : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$

H_A : at least one $\beta_i \neq 0$ (for $i = 1, 2, 3, 4$)

2. Next, check conditions for this test.

We have already checked the multiple linear regression conditions for inference and they all hold. Thus the inference that we make with this test will be valid.

3. Use the summary output table to find the test statistic for this test.

The output table tells us that $test\_stat = 176.1$.

4. Use this test statistic and the F-distribution to calculate the p-value for this test

The two sets of degrees of freedom for this test are:

- $df1 = p = 4$ (i.e. p=number of slopes)
- $df2 = n - p - 1 = 487 - 4 - 1 = 482$

$p - value = P(F_{4,482} \geq teststat) = P(F_{4,482} \geq 176.1) = 1.11 \times 10^{-16}$

In [57]:

```
pvalue=1-f.cdf(176.1, dfn=4,dfd=482)
print('p-value: ',pvalue)
```

```
p-value:  1.1102230246251565e-16
```

5. Verify this p-value using the summary output table.

The summary output tables says that $p - value = 7.71 \times 10^{-93}$. Because the p-value is very small, these p-value may be off due to some precision errors.

6. Make a conclusion with this p-value, using $\alpha = 0.05$

Because the $p - value = 7.71 \times 10^{-93} \alpha = 0.05$, we reject the null hypothesis. Thus there is sufficient evidence to suggest that at least one of the four population slopes in the model predicting weight with height, sex, and age group is non-zero.

Linear Regression Models with Interaction Variables

To allow for more flexibility in the fit of our linear regression models, we may want to include interaction terms. Interaction terms allow for different slopes for certain variables, depending on the value of a second variable.

For example, set up a multiple linear regression model predicting weight with height, sex, and the interaction between height and sex.

In [58]:
```
mod_int = smf.ols('weight~height+sex+height*sex',
                  data=df).fit()
```

In [59]:
```
mod_int.summary()
```

Out[59]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.575
Model:	OLS	**Adj. R-squared:**	0.572
Method:	Least Squares	**F-statistic:**	217.5
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	3.00e-89
Time:	12:39:50	**Log-Likelihood:**	-1748.3
No. Observations:	487	**AIC:**	3505.
Df Residuals:	483	**BIC:**	3521.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-43.8193	13.791	-3.177	0.002	-70.917	-16.721
sex[T.Male]	-16.1357	19.885	-0.811	0.418	-55.208	22.936
height	0.6333	0.084	7.577	0.000	0.469	0.798
height:sex[T.Male]	0.1453	0.116	1.252	0.211	-0.083	0.373

Omnibus:	92.090	**Durbin-Watson:**	2.002
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	187.527
Skew:	1.030	**Prob(JB):**	1.90e-41
Kurtosis:	5.235	**Cond. No.**	1.11e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.11e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

For example, is there sufficient evidence to suggest that the slope of the interaction variable (of height and sex) in the population model (that predicts weight with height, sex, and the interaction of height and sex) is non-zero?

In other words, can I have one joint slope for height regardless of sex, or should I separate males and females so that I have a height slope specific to each sex?

$H_0 : \beta_3 = 0$

$H_A : \beta_3 \neq 0$

(Where β_3 is the slope that corresponds to the interaction of height and sex).

Because the p-value for this test $= 0.211 \geq \alpha = 0.05$, we fail to reject the null hypothesis. Thus there is not sufficient evidence to suggest that there is a linear interaction effect in this population model.

Making a Prediction with Multiple Linear Regression

For example, suppose that I want to predict weight based on height, sex, and age.

First, set up the multiple linear regression model that predicts weight with height, sex, and age group.

Then predict the weight of a 20 year old woman that is 170cm tall.

In [60]:

```
final_mod = smf.ols('weight~height+sex+age_group',
                    data=df).fit()
final_mod.summary()
```

Out[60]:

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.594
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	176.1
Date:	Wed, 22 Mar 2023	**Prob (F-statistic):**	7.71e-93
Time:	12:39:50	**Log-Likelihood:**	-1737.1
No. Observations:	487	**AIC:**	3484.
Df Residuals:	482	**BIC:**	3505.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-59.0102	9.409	-6.272	0.000	-77.498	-40.523
sex[T.Male]	7.9128	1.086	7.286	0.000	5.779	10.047
age_group[T.40 and above]	3.4030	1.202	2.830	0.005	1.041	5.765
age_group[T.under_30]	-1.7432	0.949	-1.838	0.067	-3.607	0.121
height	0.7291	0.057	12.821	0.000	0.617	0.841

Omnibus:	91.450	**Durbin-Watson:**	1.991
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	182.551
Skew:	1.032	**Prob(JB):**	2.29e-40
Kurtosis:	5.176	**Cond. No.**	4.14e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.14e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Model:

$$\hat{weight} = -59.0102 + 7.9128 sex[T.Male] + 3.4030 age_group[T.40andabove]$$
$$- 1.7432 age_group[T.under_30] + 0.7291 height$$

Prediction by Hand:

$$\hat{weight} = -59.0102 + 7.9128(0) + 3.4030 age_group(0) - 1.7432(1) + 0.7291(170) = 63.1978$$

In [61]:
```
final_mod.predict(exog=dict(height=170, sex='Female', age_group='under_30'))
```

Out[61]:
```
0    63.197806
dtype: float64
```
