In [1]:

```
#1.Loading data.(10 points). Begin by downloading the file FakeData.csv and savir
# We will load the data in jupyter notebook using the read.csv() command.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('OneDrive\Desktop\Fakedata.csv')
df.head()

#df.shape
#df.dtypes
```

Out[1]:

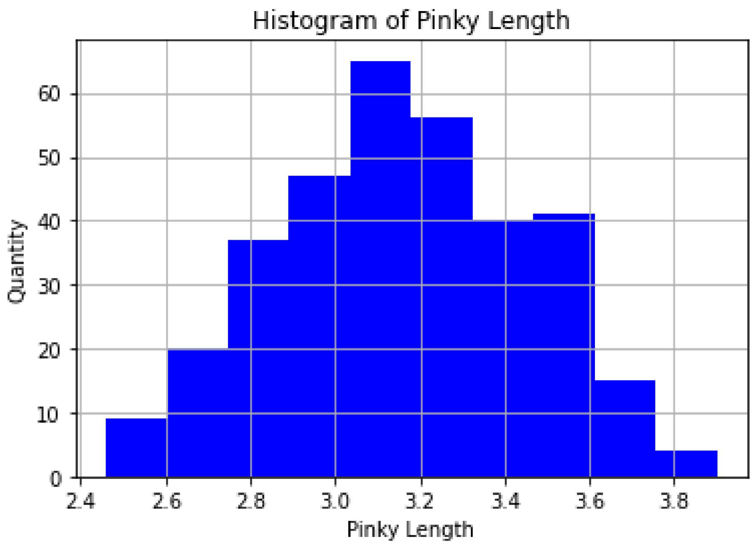
	ID	Year	FavColor	Region	Age	Breakfast	Height	Handspan	Pinkylen	Gender
0	858	2013	Grey	East	69	Multiple	58	8.4	2.81	F
1	791	2013	Other	West	68	Dairy	59	8.5	2.65	M
2	155	2013	Other	West	52	Multiple	65	9.3	2.90	M
3	546	2013	Brown	East	70	Meat	59	8.9	3.01	M
4	286	2013	Green	West	68	Multiple	63	9.5	3.19	M

In [2]:

```
#2.Histogram.(10 points). Create a histogram of the Pinkylen variable, adjust the
# set the main title to Histogram of Pinky Length and the x-axis label to an appr

hist = df.hist(column='Pinkylen', color='blue')
plt.xlabel('Pinky Length')
plt.ylabel('Quantity')
plt.title('Histogram of Pinky Length')
```

Out[2]:

```
Text(0.5, 1.0, 'Histogram of Pinky Length')
```

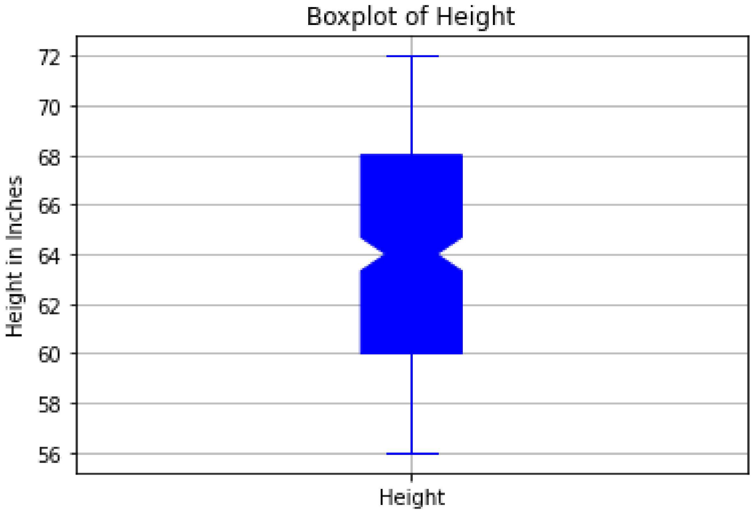


In [3]:

```
#3.Boxplots.(10 points) Construct a blue boxplot of the Height variable and label
#Provide the command and the output in your lab report.
box = df.boxplot(column = 'Height', color='Blue', notch=True, patch_artist=True)
plt.title('Boxplot of Height')
plt.ylabel('Height in Inches')
```

Out[3]:

```
Text(0, 0.5, 'Height in Inches')
```

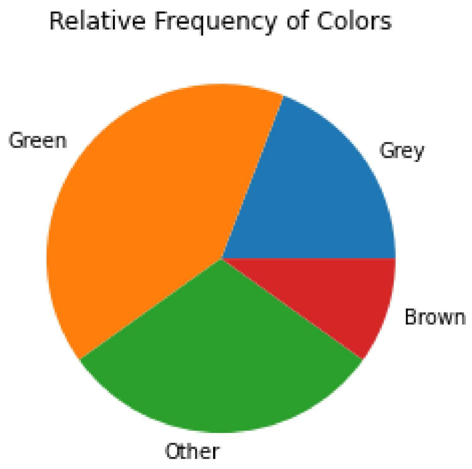


In [4]:

```
#4.Pie Chart.(10 points) Construct a pie chart of the FavColor variable based on
# Provide the command and the output in your lab report.
def rel_freq(x):
    freqs = [(value, x.count(value) / len(x)) for value in set(x)]
    return freqs

relFreqs = rel_freq(list(df['FavColor']))
#plot = plot.pie(y='relFreq', figsize=(5, 5))
relFreqs

freqs = [.1916, .4072, .3024, .0988]
marks = ["Grey", "Green", "Other", "Brown"]

plt.figure(figsize=(4, 4))
plt.pie(freqs, labels=marks)
plt.title("Relative Frequency of Colors")
```

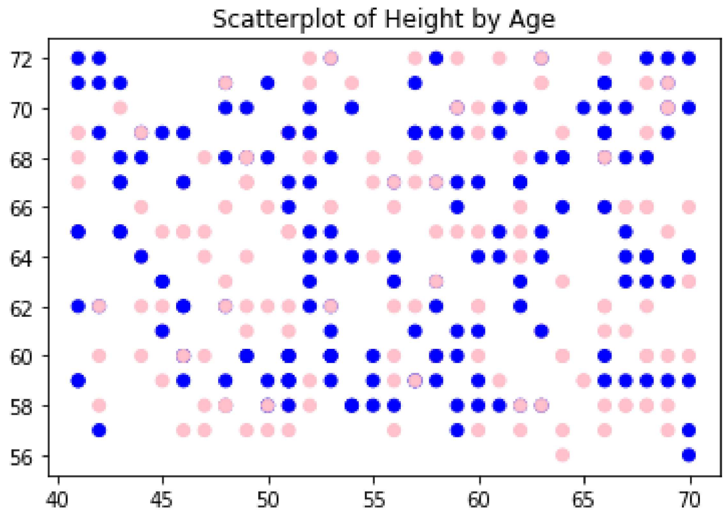
Out[4]:

```
Text(0.5, 1.0, 'Relative Frequency of Colors')
```



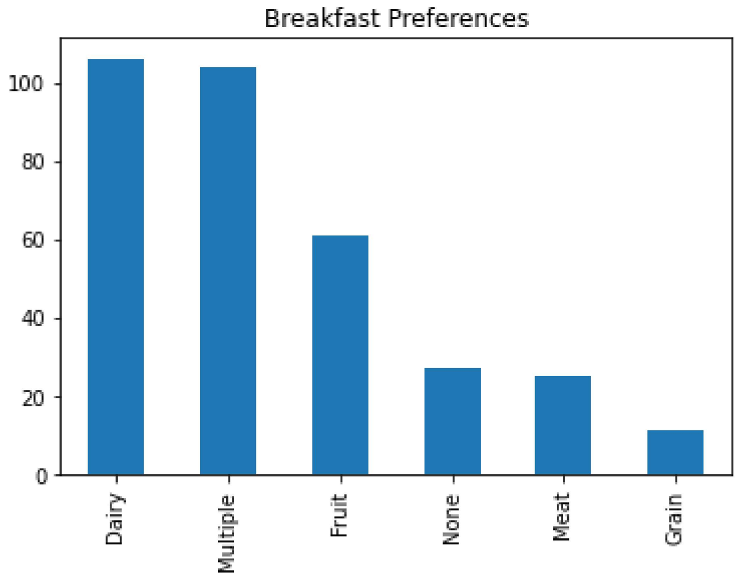
In [21]:
```
#5.Scatter Plot.(20 points)Construct a scatter plot of the Height variable for mo
#(Show in one graph)

colors = {'M':'blue', 'F':'pink'}

scatter = plt.scatter(x=df['Age'], y=df['Height'], c=df['Gender'].map(colors))
plt.title('Scatterplot of Height by Age')
```

Out[21]:
```
Text(0.5, 1.0, 'Scatterplot of Height by Age')
```



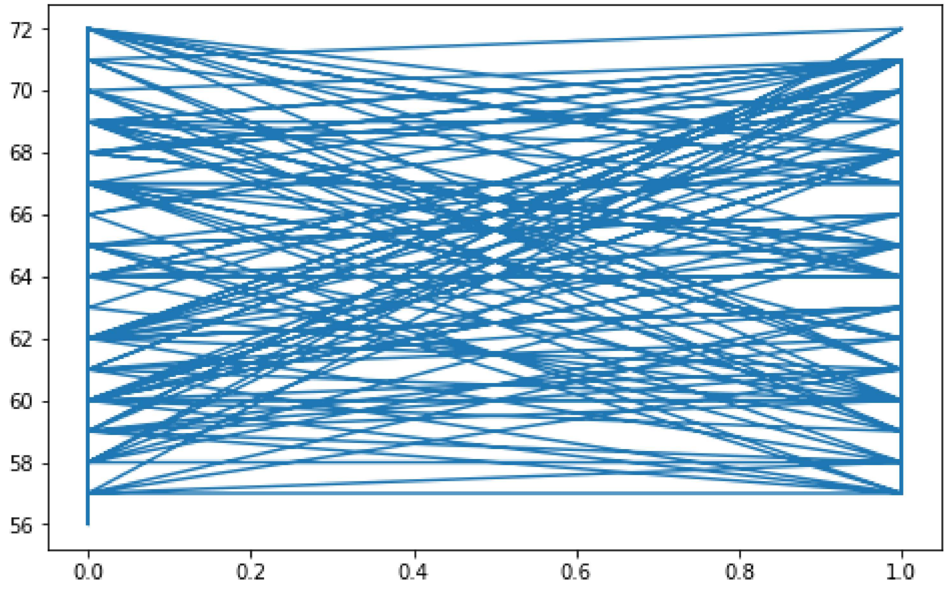
In [25]:
```
#6.Bar Graph.(10 points)Construct bar graph for the Breakfast variable and label
# Provide the command and the output in your lab report

df['Breakfast'].value_counts().plot(kind='bar')
plt.title("Breakfast Preferences")
```

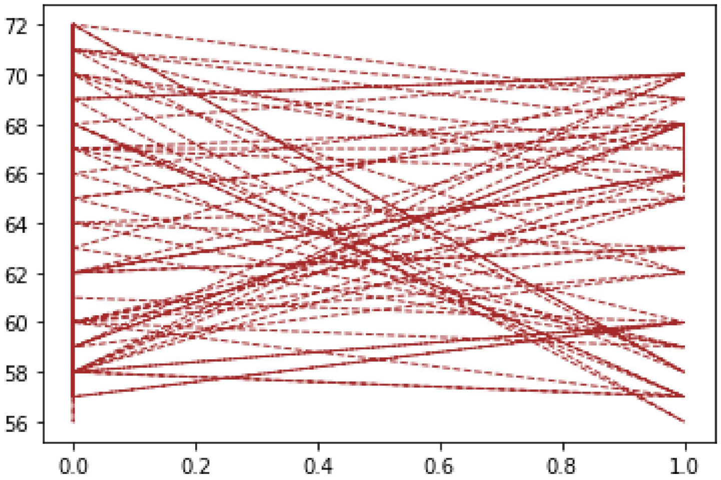
Out[25]:
```
Text(0.5, 1.0, 'Breakfast Preferences')
```

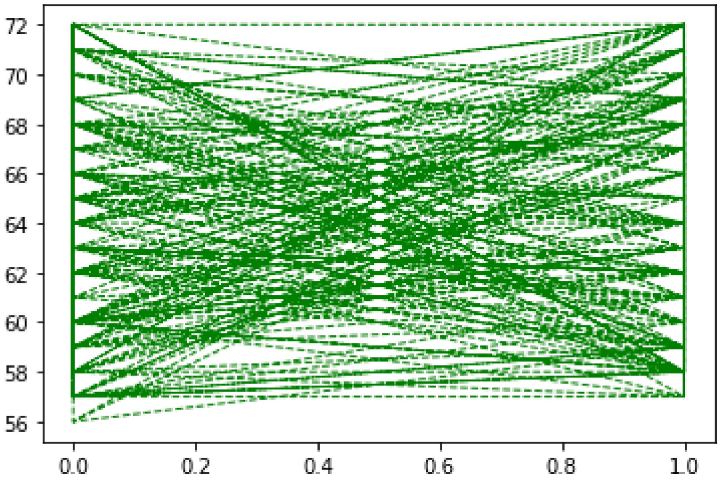


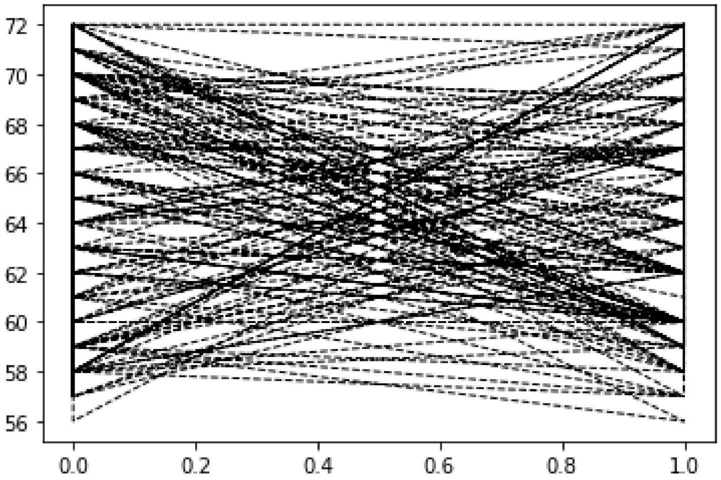
In [39]:

```
#7.Line Chart.(30 points)Construct a line chart of the Height variable based on c
# (There will be 3 subplots in one graph.)

x = df['Height']
y1 = df['FavColor'] == 'Grey'
y2 = df['FavColor'] == 'Brown'
y3 = df['FavColor'] == 'Green'
y4 = df['FavColor'] == 'Other'

plt.figure(num = 3, figsize=(8, 5))

plt.plot(y1, x)
plt.show()

plt.plot(y2, x,
         color='brown',
         linewidth=1.0,
         linestyle='--'
        )
plt.show()

plt.plot(y3, x,
         color='green',
         linewidth=1.0,
         linestyle='--'
        )
plt.show()

plt.plot(y4, x,
         color='black',
         linewidth=1.0,
         linestyle='--'
        )
plt.show()
```




72
70
68
66
64
62
60
58
56
0.0
0.2
0.4
0.6
0.8
1.0


72
70
68
66
64
62
60
58
56
0.0
0.2
0.4
0.6
0.8
1.0


72
70
68
66
64
62
60
58
56
0.0
0.2
0.4
0.6
0.8
1.0

In [ ]:

In [ ]: