Bungee Characterization Lab

1/22/20

PH 211

Katherine Skovborg

Lab partners: Walker Davis, Casey Heiskell

Overview:

In this lab, we explored the concept of the 'spring constant', or the stretchiness of a string, when subjected to a known force. To do this, we attached weights to surgical tubes hung from the ceiling to measure the change in stretch when attached to a certain weight. We used these measurements to determine the spring force as a function of the stretch. Once again, we entered the data we collected in python and used various tools to plot and analyze the data.

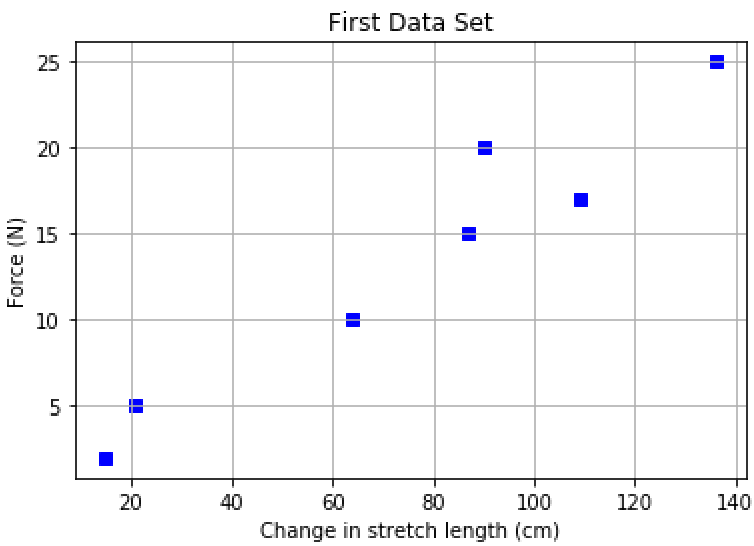
In [18]:
```
#import python tool libraries

import numpy as np
import matplotlib as mplot
import matplotlib.pyplot as plt
```

In [4]:

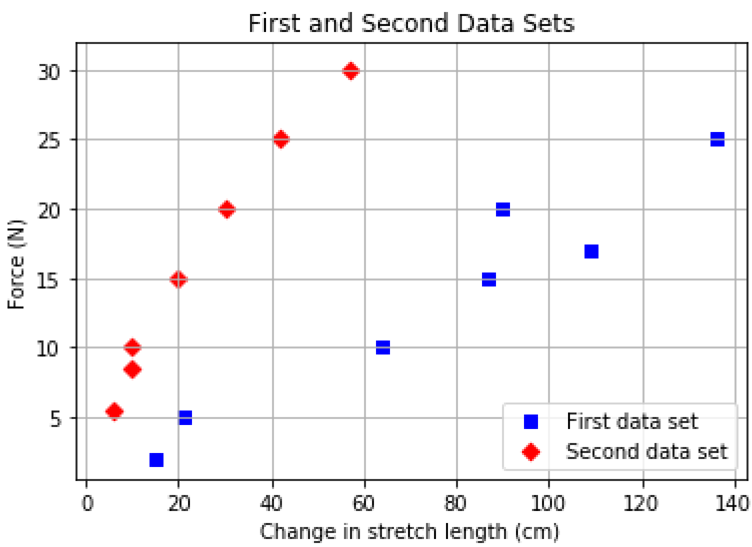
```
#Enter first and second data sets

xdata1 = [15,21,64,87,109,90,136,]

ydata1 = [2,5,10,15,17,20,25,]

xdata2 = [6,10,10,20,30,42,57]

ydata2 = [5.5,8.5,10,15,20,25,30]

print("First set of data:")
print("Stretch data: ", xdata1)
print("Force data: ", ydata1)

#Check length of first data set

lenxdata1 = len(xdata1)
lenydata1 = len(ydata1)

print("Number of data points (x): ", lenxdata1)
print("Number of data points (y): ", lenydata1)
print()

print("Second set of data:")
print("Stretch data: ", xdata2)
print("Force data: ", ydata2)

#Check length of second data set

lenxdata2 = len(xdata2)
lenydata2 = len(ydata2)

print("Number of data points (x): ", lenxdata2)
print("Number of data points(y): ", lenydata2)
```

```
First set of data:
Stretch data:  [15, 21, 64, 87, 109, 90, 136]
Force data:  [2, 5, 10, 15, 17, 20, 25]
Number of data points (x):  7
Number of data points (y):  7

Second set of data:
Stretch data:  [6, 10, 10, 20, 30, 42, 57]
Force data:  [5.5, 8.5, 10, 15, 20, 25, 30]
Number of data points (x):  7
Number of data points(y):  7
```

In [5]:
```
#Create a scatter plot of the first set of data

fig, ax = plt.subplots()
ax.scatter(xdata1, ydata1, color = "blue", marker = "s")

ax.set(xlabel='Change in stretch length (cm)', ylabel='Force (N)',
       title='First Data Set')
ax.grid()

plt.show()
```



In [6]:
```
#Create a scatter plot of the first and second data sets

fig, ax = plt.subplots()
ax.scatter(xdata1, ydata1, color = 'blue', marker = "s", label = "First
 data set")
ax.scatter(xdata2, ydata2, color = 'red', marker = "D", label = "Second
 data set")

ax.set(xlabel='Change in stretch length (cm)', ylabel='Force (N)',
       title='First and Second Data Sets')
ax.grid()


plt.legend(loc = 4)


plt.show()
```

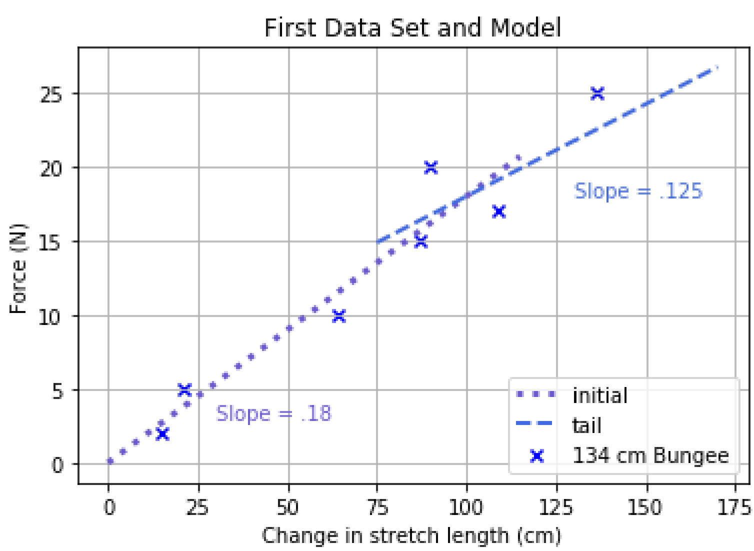


Discussion Deliverable 2:

Hooke's Law describes an ideal physics spring and states that the force needed to extend or compress a spring by some distance is proportional to that distance. To an extent, both data sets have features that are consistent with this law and have somewhat similar behaviors. The first data set is quite linear, disregarding one outlying data point, whereas the second data set resembles an exponential function. Overall, both data sets begin with steep slopes that become less steep as more force is applied.

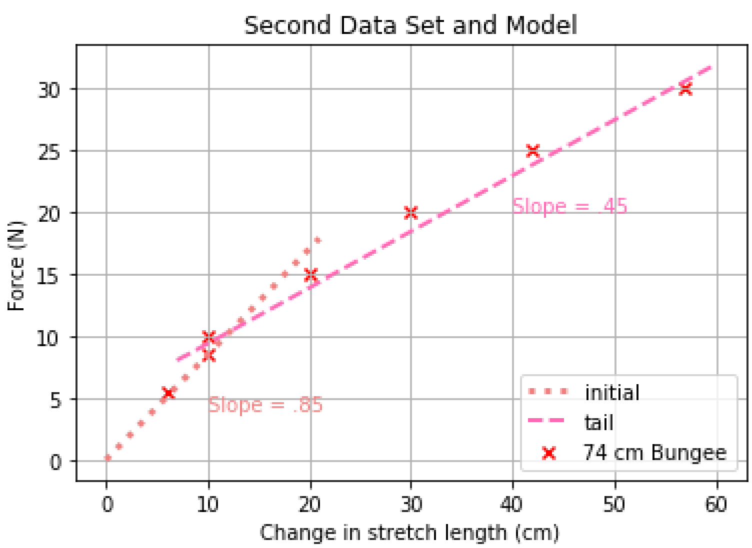
In [59]:
```
# actual model parameters - slope and intercept
model1slope = .18
model2slope = .125
model1int = 0.
model2int = 5.5

# range of x values -- choose lower and upper limits of range
model1x = np.linspace(0.,115.,20)
model2x = np.linspace(75.,170.,20)


# generate y values from model
model1y = model1slope*model1x + model1int
model2y = model2slope*model2x + model2int
```

In [60]:
```
#Create scatter plot of first data set with physics model

fig2, ax2 = plt.subplots()
ax2.scatter(xdata1, ydata1, marker = 'x', color = 'blue', label = "134 c
m Bungee")
ax2.plot(model1x, model1y, color = 'slateblue', linestyle = ':', linewid
th = 3., label = "initial")
ax2.plot(model2x, model2y, color = 'royalblue', linestyle = '--', linewi
dth = 2., label = "tail")
ax2.set(xlabel='Change in stretch length (cm)', ylabel='Force (N)',
       title='First Data Set and Model')
ax2.grid()

plt.text(30,3, "Slope = .18", color = "slateblue")
plt.text(130,18, "Slope = .125", color = "royalblue")

plt.legend(loc= 4)
plt.show()
```



In [66]:
```
# actual model parameters - slope and intercept
model1slope = .85
model2slope = .45
model1int = 0
model2int = 4.9

# range of x values -- choose lower and upper limits of range
model1x = np.linspace(0.,21.,20)
model2x = np.linspace(7.,60.,20)


# generate y values from model
model1y = model1slope*model1x + model1int
model2y = model2slope*model2x + model2int
```

In [67]:
```
#Create scatter plot of second data set with physics model

fig2, ax2 = plt.subplots()
ax2.scatter(xdata2, ydata2, marker = 'x', color = 'red', label = "74 cm
 Bungee")
ax2.plot(model1x, model1y, color = 'lightcoral', linestyle = ':', linewi
dth = 3., label = "initial")
ax2.plot(model2x, model2y, color = 'hotpink', linestyle = '--', linewidt
h = 2., label = "tail")
ax2.set(xlabel='Change in stretch length (cm)', ylabel='Force (N)',
       title='Second Data Set and Model')
ax2.grid()

plt.text(10,4, "Slope = .85", color = "lightcoral")
plt.text(40,20, "Slope = .45", color = "hotpink")

plt.legend(loc= 4)
plt.show()
```

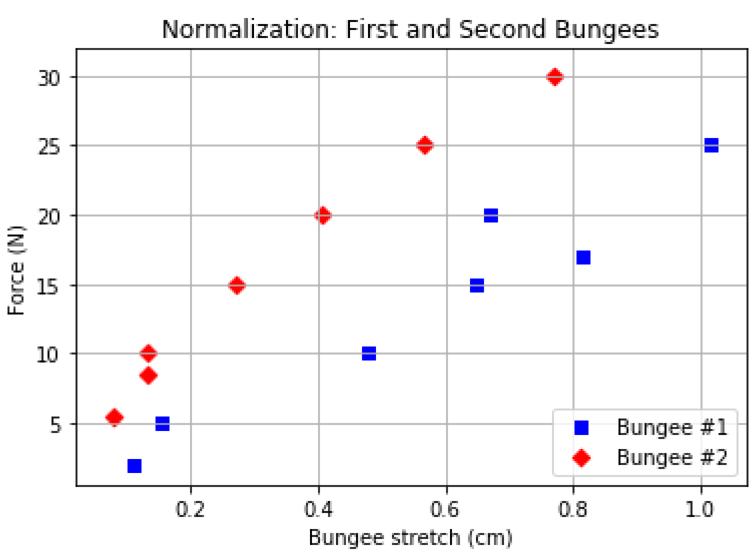


Discussion Deliverable 3:

Despite the fact that the first data set is much more linear than the second, applying the physics model/two line fit clearly illuminates a similar behavior in both bungees. In both graphs, the slope of the function is steepest initially and then becomes less steep as force increases. In other words, both bungees are the most stiff at the beginning. Neither of the data sets are completely linear or proportional, so they don't fully abide by Hooke's Law. This happens for two main reasons. First, the bungee becomes less stiff as it is used more. Second, the further you strech a bungee cord, the more it loses its elasticity. This explains why the slopes of the funtions decrease as force increases.

In [15]:
```
#Normalize the stretch of the bungee from each data set

length1 = 134.
xdata1norm = np.true_divide(xdata1,length1)

length2 = 74.
xdata2norm = np.true_divide(xdata2,length2)
```

In [16]:
```
#Create scatter plot representing normalization of both data sets (two s
eparate bungees)

fig, axn = plt.subplots()
axn.scatter(xdata1norm, ydata1, color = "blue", marker = "s", label = "B
ungee #1")
axn.scatter(xdata2norm, ydata2, color = "red", marker = "D", label = "Bu
ngee #2")

axn.set(xlabel='Bungee stretch (cm)', ylabel='Force (N)',
       title='Normalization: First and Second Bungees')
axn.grid()
plt.legend(loc = 4)

plt.show()
```



Discussion Deliverable 4:

We use normalization to determine whether or not a particular feature of the data has a simple dependence. In this lab, each bungee is a different length, so we normalized the data to see if they behave the same regardless. The graph above shows that the bungees do in fact behave the same. The data from bungee # 1 has an outlying data point, but if you look closely, both bungees produced a function with (basically) the same shape. This means that we could predict that the same type of bungee cords of different lengths would behave similarly.