Data Meaning Lab

1/15/20

PH211

Katherine Skovborg

Lab Partners: Walker Davis, Casey Heiskell

Overview:

In this lab we were split into two groups. One at a time, each group had to estimate the length of two different hallways in a matter of seconds without using references. We then recorded each individual's estimations of each hallway length. We will continue to learn how to apply tools stored in python to enter, store, and plot the data we collected. This allows us to calculate the mean, median, standard deviation, minimum, and maximum values of each data set. In addition, we will plot several different histograms using these python tools in order to analyze the data and interpret what it means.

In [43]:
```
#Import python tool libraries

import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
```

In [15]:
```
#Enter data for each hallway from group 1

data_long = [130, 20, 80, 25, 60, 36, 22, 35, 30, 50, 80, 45, 30]
data_short = [100, 15, 40, 15, 40, 30, 20, 28, 20, 30, 60, 20, 20]

print("First data set (long hallway): ", data_long)
print("Second data set (short hallway): ", data_short)

#check length of data points for each data set

data_long_length = len(data_long)
data_short_length = len(data_short)

print("Length of first data set: ", data_long_length)
print("Length of second data set: ", data_long_length)
```

```
First data set (long hallway):  [130, 20, 80, 25, 60, 36, 22, 35, 30, 5
0, 80, 45, 30]
Second data set (short hallway):  [100, 15, 40, 15, 40, 30, 20, 28, 20,
30, 60, 20, 20]
Length of first data set:  13
Length of second data set:  13
```

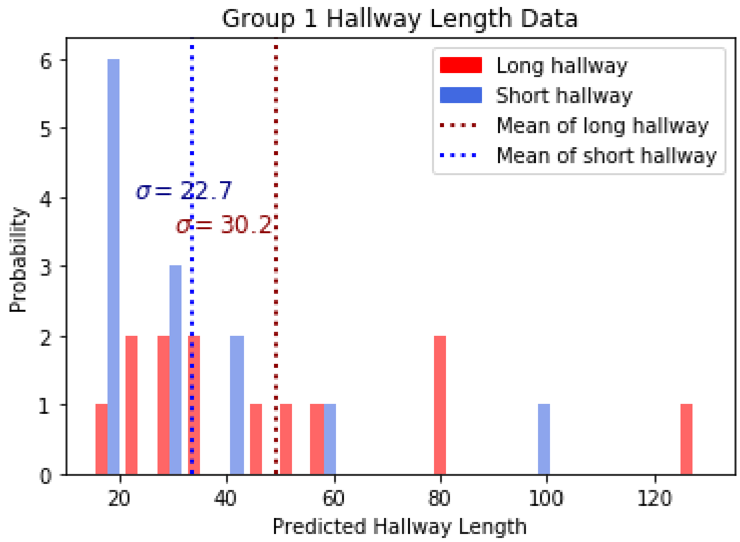
In [45]:
```
#Find the mean, median, standard deviation, minumum, and maximum for fir
st data set

print("Group 1 long hallway statistics:")

data_long_mean = np.mean(data_long)
data_long_median = np.median(data_long)
data_long_stddev = np.std(data_long)
data_long_min = np.min(data_long)
data_long_max = np.max(data_long)

print("Mean of first data set: ", data_long_mean)
print("Median of first data set: ", data_long_median)
print("Standard deviation of first data set: ", data_long_stddev)
print("Minimum of first data set: ", data_long_min)
print("Maximum of first data set: ", data_long_max)
```

```
Group 1 long hallway statistics:
Mean of first data set:  49.46153846153846
Median of first data set:  36.0
Standard deviation of first data set:  30.193263091204212
Minimum of first data set:  20
Maximum of first data set:  130
```

In [175]:
```
#Find the mean, median, standard deviation, minumum, and maximum for sec
ond data set

print("Group 1 short hallway statistics:")

data_short_mean = np.mean(data_short)
data_short_median = np.median(data_short)
data_short_stddev = np.std(data_short)
data_short_min = np.min(data_short)
data_short_max = np.max(data_short)

print("Mean of second data set: ", data_short_mean)
print("Median of second data set: ", data_short_median)
print("Standard deviation of second data set: ", data_short_stddev)
print("Minimum of second data set: ", data_short_min)
print("Maximum of second data set: ", data_short_max)
```

```
Group 1 short hallway statistics:
Mean of second data set:  33.69230769230769
Median of second data set:  28.0
Standard deviation of second data set:  22.662692539415847
Minimum of second data set:  15
Maximum of second data set:  100
```

In [181]:
```
#First histogram displaying each data set
#Example of 'too cold'... has too many bins

plt.title("Group 1 Hallway Length Data")
plt.xlabel("Predicted Hallway Length")
plt.ylabel("Probability")

num_bins = 20
fullrange = [15,130]
height, bins, patches = plt.hist([data_long,data_short], num_bins, fullr
ange,
                                    histtype = "bar", color=["red","roya
lblue"], alpha= .6)


#plot mean and standard deviation text box for each data set

plt.text(30.2, 3.5, '$\sigma=30.2$', color = "darkred", fontsize = "larg
e")
plt.text(22.7, 4, '$\sigma=22.7$', color = "navy", fontsize = "large")

redlinemean = plt.axvline(x = data_long_mean, ymin = 0, ymax = 6, color
= "darkred", linestyle = ":", linewidth = 2, label = "Mean of long hallw
ay")
bluelinemean = plt.axvline(x = data_short_mean, ymin = 0, ymax = 6, colo
r = "blue", linestyle = ":", linewidth = 2, label = "Mean of short hallw
ay")

#Create a legend

red_patch = mpatches.Patch(color='red', label='Long hallway')
royalblue_patch = mpatches.Patch(color='royalblue', label='Short hallwa
y')



plt.legend(handles=[red_patch, blue_patch, redlinemean, bluelinemean])


plt.show()
```



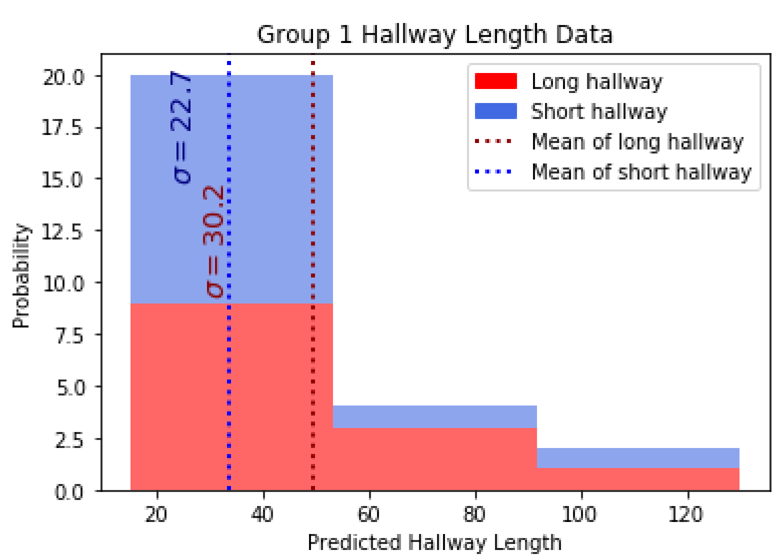
Strengths/Weaknesses:

Too cold

This is the most standard histogram. If only one set of data were being plotted, it could be a more ideal histogram. Another strength to this histogram is that it leaves a nice, clean space for extra labeling, etc. Although, when more than one data set is being plotted, it creates gaps between bars indicating that there is missing data, which is not true in this case. In addition, there are too many bins. This makes it more difficult to interpret the data.

In [209]:

```
#Second histogram displaying both data sets
#Example of 'too hot'... too few bins

plt.title("Group 1 Hallway Length Data")
plt.xlabel("Predicted Hallway Length")
plt.ylabel("Probability")

num_bins = 3
fullrange = [15,130]
height, bins, patches = plt.hist([data_long,data_short], num_bins, fullr
ange,
                                     histtype = "barstacked", color=["re
d","royalblue"], alpha= .6)


#Plot mean and standard deviation for each data set

plt.text(28.5, 9.5, '$\sigma=30.2$', color = "darkred", fontsize = "x-la
rge", rotation = 90)
plt.text(22.7, 15, '$\sigma=22.7$', color = "navy", fontsize = "x-large"
, rotation = 90)

redlinemean = plt.axvline(x = data_long_mean, ymin = 0, ymax = 6, color
= "darkred", linestyle = ":", linewidth = 2, label = "Mean of long hallw
ay")
bluelinemean = plt.axvline(x = data_short_mean, ymin = 0, ymax = 6, colo
r = "blue", linestyle = ":", linewidth = 2, label = "Mean of short hallw
ay")


#Create a legend

red_patch = mpatches.Patch(color='red', label='Long hallway')
blue_patch = mpatches.Patch(color='royalblue', label='Short hallway')

plt.legend(handles=[red_patch, blue_patch, redlinemean, bluelinemean])


plt.show()
```



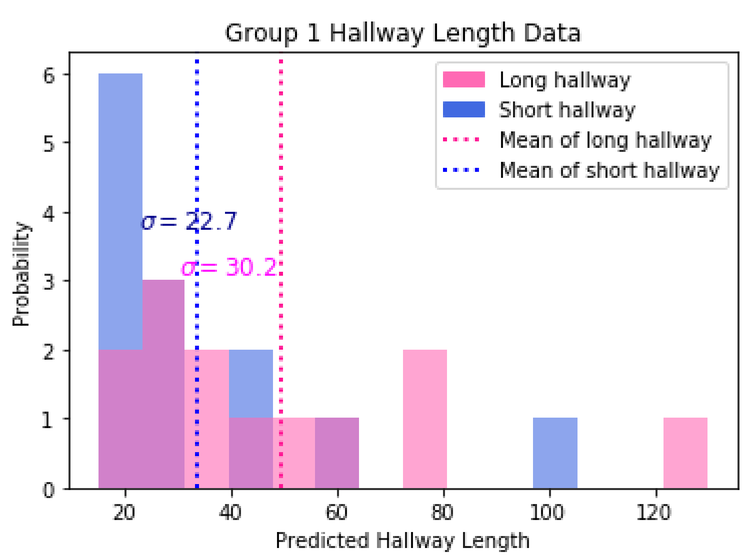
Strengths/Weaknesses:

Too hot

One thing that makes this histogram better than the previous one is that it fills the gaps between data. Although, this is partially due to the fact that all the data points are grouped into three bins. This also means that there isn't a clear display of the overlap between data sets. Like the graph above, there are too few bins to interpret the data with ease.

In [210]:

```
#Third histogram displaying both data sets
#Example of 'just right'... the perfect amount of bins

plt.title("Group 1 Hallway Length Data")
plt.xlabel("Predicted Hallway Length")
plt.ylabel("Probability")

num_bins = 14
fullrange = [15,130]
height, bins, patches = plt.hist([data_long,data_short], num_bins, fullr
ange,
                                 histtype = "stepfilled", color=["ho
tpink","royalblue"], alpha= .6)


#Plot mean and standard deviation for each data set

pinklinemean = plt.axvline(x = data_long_mean, ymin = 0, ymax = 6, color
= "deeppink", linestyle = ":", linewidth = 2, label = "Mean of long hall
way")
bluelinemean = plt.axvline(x = data_short_mean, ymin = 0, ymax = 6, colo
r = "blue", linestyle = ":", linewidth = 2, label = "Mean of short hallw
ay")

plt.text(30.2, 3.1, '$\sigma=30.2$', color = "magenta", fontsize = "larg
e")
plt.text(22.7, 3.75, '$\sigma=22.7$', color = "darkblue", fontsize = "la
rge")

#Create a legend

red_patch = mpatches.Patch(color='hotpink', label='Long hallway')
blue_patch = mpatches.Patch(color='royalblue', label='Short hallway')
plt.legend(handles=[red_patch, blue_patch, pinklinemean, bluelinemean])


plt.show()
```



Strengths/Weaknesses:

Just right

This histogram has the right amount of bins, fills gaps where there should be data, and clearly displays the overlap of data sets. It is clean, leaves room for extra labeling, and is the easiest to interpret and differ between the data points with this histogram style.

Analysis 1

Data From Group 1

The mean of the long hallway is 50 and the median is 36. The mean of the short hallway is 30 and the median is 28. These statistics are consistent with the graph, as shown above. The concentration of data points show that the majority of estimations for both hallways are between 15 and 60 meters. The standard deviation for the long hallway is 30 and the short is 22. The group 1 data overlaps between the $\pm 1\sigma$ range and therefore it cannot be stated with certainty that two different hallways were being measured. The data does not provide enough evidence.

In [19]:
```
#Enter both data sets from group 2

print("Data from group two:")

data_long2 = [60,25,80,75,50,50,40,25,20]
data_short2 = [20,15,30,20,25,20,35,15,15]

print("First data set (long hallway): ", data_long2)
print("Second data set (short hallway): ", data_short2)

#Check length of data points from each data set from group two

len_data_long2 = len(data_long2)
len_data_short2 = len(data_short2)

print("Length of first data set: ", len_data_long2)
print("Length of second data set: ", len_data_short2)
```

```
Data from group two:
First data set (long hallway):  [60, 25, 80, 75, 50, 50, 40, 25, 20]
Second data set (short hallway):  [20, 15, 30, 20, 25, 20, 35, 15, 15]
Length of first data set:  9
Length of second data set:  9
```

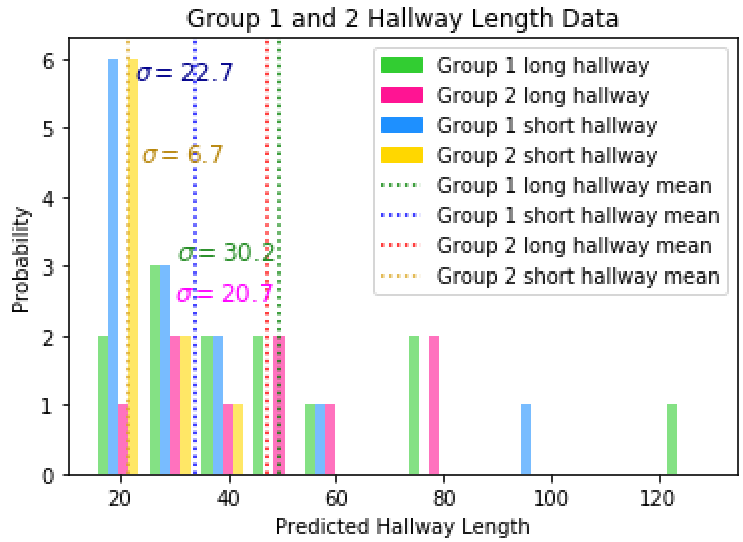
In [47]:
```
#Find the mean, median, standard deviation, minimum, and maximum of firs
t data set from group two

print("Group 2 long hallway statistics:")

data_long2_mean = np.mean(data_long2)
data_long2_median = np.median(data_long2)
data_long2_stddev = np.std(data_long2)
data_long2_max = np.max(data_long2)
data_long2_min = np.min(data_long2)

print("Mean of first data set: ", data_long2_mean)
print("Median of first data set: ", data_long2_median)
print("Standard deviation of first data set: ", data_long2_stddev)
print("Maximum of first data set: ", data_long2_max)
print("Minimum of first data set: ", data_long2_min)
```

```
Group 2 long hallway statistics:
Mean of first data set:  47.22222222222222
Median of first data set:  50.0
Standard deviation of first data set:  20.56306169257972
Maximum of first data set:  80
Minimum of first data set:  20
```

In [48]:
```
#Find the mean, median, standard deviation, maximum, and minimum of firs
t data set from group two

print("Group 2 short hallway statistics:")

data_short2_mean = np.mean(data_short2)
data_short2_median = np.median(data_short2)
data_short2_stddev = np.std(data_short2)
data_short2_max = np.max(data_short2)
data_short2_min = np.min(data_short2)

print("Mean of second data set: ", data_short2_mean)
print("Median of second data set: ", data_short2_median)
print("Standard deviation of second data set: ", data_short2_stddev)
print("Maximum of second data set: ", data_short2_max)
print("Minimum of second data set: ", data_short2_min)
```

```
Group 2 short hallway statistics:
Mean of second data set:  21.666666666666668
Median of second data set:  20.0
Standard deviation of second data set:  6.666666666666667
Maximum of second data set:  35
Minimum of second data set:  15
```

In [221]:
```
#Plot a histogram that displays both data sets from both groups


plt.title("Group 1 and 2 Hallway Length Data")
plt.xlabel("Predicted Hallway Length")
plt.ylabel("Probability")

num_bins = 12
fullrange = [15,130]
height, bins, patches = plt.hist([data_long,data_short,data_long2,data_s
hort2], num_bins,
                                     fullrange, histtype = "bar",
                                         color=["limegreen","dodgerblue"
,"deeppink","gold"], alpha= .6)


#Plot mean and standard deviation for each data set

plt.text(30.4, 3.1, '$\sigma=30.2$', color = "forestgreen", fontsize =
"large")
plt.text(22.9, 5.7, '$\sigma=22.7$', color = "darkblue", fontsize = "lar
ge")
plt.text(30, 2.5, '$\sigma=20.7$', color = "magenta", fontsize = "large"
)
plt.text(24, 4.5, '$\sigma=6.7$', color = "darkgoldenrod", fontsize = "l
arge")

greenlinemean = plt.axvline(x = data_long_mean, ymin = 0, ymax = 6, colo
r = "green", linestyle = ":", label = "Group 1 long hallway mean")
bluelinemean = plt.axvline(x = data_short_mean, ymin = 0, ymax = 6, colo
r = "blue", linestyle = ":", label = "Group 1 short hallway mean")
redlinemean = plt.axvline(x = data_long2_mean, ymin = 0, ymax = 6, color
= "red", linestyle = ":", label = "Group 2 long hallway mean")
goldlinemean = plt.axvline(x = data_short2_mean, ymin = 0, ymax = 6, col
or = "goldenrod", linestyle = ":", label = "Group 2 short hallway mean")


#Create a legend

redpatch = mpatches.Patch(color = "limegreen", label = "Group 1 long hal
lway")
bluepatch = mpatches.Patch(color = "dodgerblue", label = "Group 1 short
 hallway")
greenpatch = mpatches.Patch(color = "deeppink", label = "Group 2 long ha
llway")
orangepatch = mpatches.Patch(color = "gold", label = "Group 2 short hall
way")

plt.legend(handles = [redpatch,greenpatch,bluepatch,orangepatch, greenli
nemean, bluelinemean, redlinemean, goldlinemean])


plt.show()
```



Analysis 2

Group 1 and 2 data

This situation is very similar to my first analysis. According to the lab procedure, both groups measured the same two hallways under the same guidlines. Of course, the data says differently.

Group 1 Long hallway: Mean = 49.5, Standard deviation = 30.2, Short hallway: Mean = 33.7, Standard deviation = 22.7

Group 2 Long hallway: Mean = 47.2, Standard deviation = 20.7, Short hallway: Mean = 21.7, Standard deviation = 6.7

Once again, the mean of each data set from each group is within $\pm 1\sigma$, so there is no way of saying that each hallway is different. In addition, you cannot determine whether or not each group measured the same hallways.