

Overview of Creating Graphs

Introduction:

To generate my graphs, I downloaded a data source from Kaggle. The spread sheet contained 33 columns of different power lifting variables and 8,000 rows of which represented different powerlifting competitors. To analyse this data set I used python through Jupyter on Anaconda, and accessed functions from NumPy, Matplotlib and Pandas libraries.

Code for Body Weight vs Lift Totals

```
#libraries to assist data analysis
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

```
#checking what directory is being used
import os
os.getcwd()
```

```
'/Users/willsaliba/Downloads/grand challenges'
```

```
#importing file as variable df 10001rx33c Including name row
df = pd.read_excel("./data.xlsx")
```

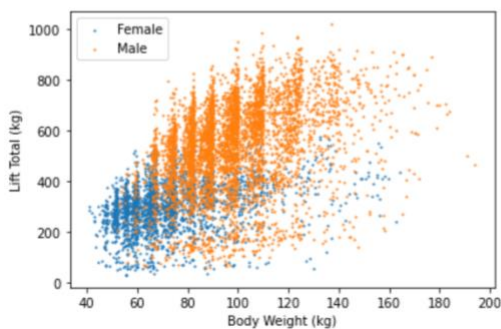
```
#display start of df
df.head(3)
```

	Name	Sex	Event	Equipment	Age	AgeClass	Division	BodyweightKg	WeightClassKg	Squat1Kg	...	Best3DeadliftKg	TotalKg	Place	Wilks	McCulloch
0	Abbie Murphy	F	SBD	Wraps	29.0	24-34	F-OR	59.8	60	80.0	...	130.0	290.0	4	324.16	324.16
1	Abbie Tuong	F	SBD	Wraps	29.0	24-34	F-OR	58.5	60	100.0	...	145.0	332.5	2	378.07	378.07
2	Ainslee Hooper	F	B	Raw	40.0	40-44	F-OR	55.4	56	NaN	...	NaN	32.5	1	38.56	38.56

3 rows x 33 columns

```
#seperating male and female
female=df[df.Sex == 'F']
male=df[df.Sex == 'M']
```

```
#graphing
plt.scatter(female.BodyweightKg, female.TotalKg,1)
plt.scatter(male.BodyweightKg, male.TotalKg,1)
plt.xlabel("Body Weight (kg)")
plt.ylabel("Lift Total (kg)")
plt.legend(["Female", "Male"])
plt.show()
```



Code for Number of Competitors vs Age

```
#using counter function
from collections import Counter

#creating freq array of ages for male

c=Counter(male.Age)
mAgeFreq=np.zeros(71, dtype=int)
age=12
for i in range(71):
    mAgeFreq[i]=c[age]
    age=age+1

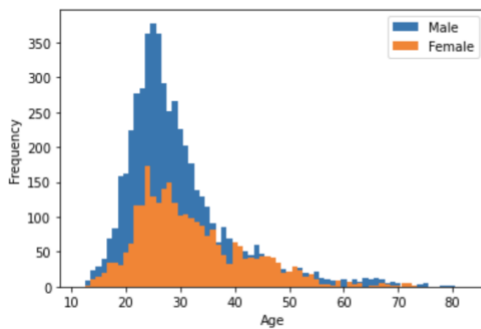
#returns array of number of competitors for each age
#creating array which can be graphed
#only inputting ages from 12+ because no competitors younger than 12
#for loop to fill array which can be graphed

#repeat same as above for female

c=Counter(female.Age)
fAgeFreq=np.zeros(71, dtype=int)
age=12
for i in range(71):
    fAgeFreq[i]=c[age]
    age=age+1
```

```
#creating array of ages to use as x axis units
ages=np.zeros(71, dtype=int)
index=12
for i in range(71):
    ages[i]=index
    index=index+1
```

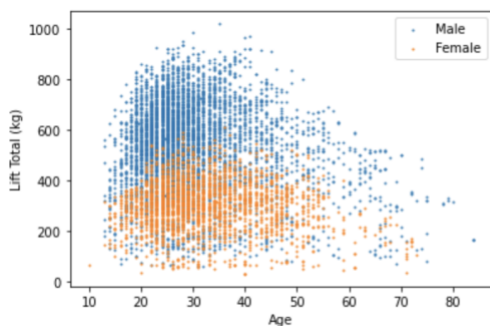
```
#graphing
width=1
plt.bar(ages,mAgeFreq,width=width)
plt.bar(ages,fAgeFreq,width=width)
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.legend(["Male", "Female"])
plt.show()
```



Code for Age VS Lift Total

```
mAge=male.Age
mTotal=male.TotalKg
```

```
plt.scatter(male.Age,male.TotalKg,1)
plt.scatter(female.Age,female.TotalKg,1)
plt.xlabel("Age")
plt.ylabel("Lift Total (kg)")
plt.legend(["Male", "Female"])
plt.show()
```



Code for Equipment Type VS Lift Total

```
#filling nan values with 0
df.replace(np.nan, 0);
```

```
#creating array to hold frequency of each type for males [raw, single-ply, multi-ply, wraps]
mFreqType=np.zeros(4, dtype=int)
#creating array to cumulative sum of totals for each type, for males [raw, single-ply, multi-ply, wraps]
mTypeSumTotal=np.zeros(4, dtype=int)

#using for loop to fill each frequency of equip type array and sum of totals for each type
for i in range(len(male)):
    if (male.TotalKg.iloc[i]>0):
        if (male.Equipment.iloc[i]=='Raw'):
            #ensuring valid entry
            #ensuring correct type
            mFreqType[0] = mFreqType[0] + 1
            mTypeSumTotal[0] = mTypeSumTotal[0] + male.TotalKg.iloc[i]
        elif (male.Equipment.iloc[i]=='Single-ply'):
            mFreqType[1] = mFreqType[1]+1
            mTypeSumTotal[1] = mTypeSumTotal[1] + male.TotalKg.iloc[i]
        elif (male.Equipment.iloc[i]=='Multi-ply'):
            mFreqType[2] = mFreqType[2]+1
            mTypeSumTotal[2] = mTypeSumTotal[2] + male.TotalKg.iloc[i]
        elif (male.Equipment.iloc[i]=='Wraps'):
            mFreqType[3] = mFreqType[3]+1
            mTypeSumTotal[3] = mTypeSumTotal[3] + male.TotalKg.iloc[i]
```

```
#using frequency and cumulative totals can now calculate average for each type
mAveType=np.zeros(4, dtype=int) #ave array
for i in range(4):
    mAveType[i] = mTypeSumTotal[i] / mFreqType[i]
```

```
#repeating for woman
fFreqType=np.zeros(4, dtype=int)
fTypeSumTotal=np.zeros(4, dtype=int)
for i in range(len(female)):
    if (female.TotalKg.iloc[i]>0):
        if (female.Equipment.iloc[i]=='Raw'):
            fFreqType[0] = fFreqType[0] + 1
            fTypeSumTotal[0] = fTypeSumTotal[0] + female.TotalKg.iloc[i]
        elif (female.Equipment.iloc[i]=='Single-ply'):
            fFreqType[1] = fFreqType[1]+1
            fTypeSumTotal[1] = fTypeSumTotal[1] + female.TotalKg.iloc[i]
        elif (female.Equipment.iloc[i]=='Multi-ply'):
            fFreqType[2] = fFreqType[2]+1
            fTypeSumTotal[2] = fTypeSumTotal[2] + female.TotalKg.iloc[i]
        elif (female.Equipment.iloc[i]=='Wraps'):
            fFreqType[3] = fFreqType[3]+1
            fTypeSumTotal[3] = fTypeSumTotal[3] + female.TotalKg.iloc[i]

print(fFreqType)
print(fTypeSumTotal)
```

```
[ 461    0   36 2125]
[ 84984    0 10708 708348]
```

```
#because 0 single-ply data for female cannot use for loop to divide fTypeSumTotal / fFreqType because cannot do 0 / 0
#so have to do manually for each
fAveType=np.zeros(4, dtype=int) #create female ave array
fAveType[0] = fTypeSumTotal[0] / fFreqType[0]
fAveType[1] = 0
fAveType[2] = fTypeSumTotal[2] / fFreqType[2]
fAveType[3] = fTypeSumTotal[3] / fFreqType[3]
```

```
#graphing
width=.4
x=np.arange(4)
plt.xticks(x,["Raw", "Single-ply", "Multi-ply", "Wraps"])
plt.bar(x-(width/2),mAveType,width=width)
plt.bar(x+(width/2),fAveType,width=width)
plt.xlabel("Equipment Type")
plt.ylabel("Average Lift Total (Kg)")
plt.legend(["Male", "Female"])
plt.show()
```

