Weightlifting

March 23, 2023

In this jupyter notebook, I'll analyze weightlifting data from the summer olympics from 2000 to 2020 and see what interesting items I can come up with. Specifically, I'll take a look at the ratios of weights lifted to bodyweight for different weightclasses. It generally holds that heavier competitors can lift heavier weights, so investigating the ratios of weights lifted to bodyweight may be more interesting.

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

[2]:
```
import os
os.getcwd()
```

[2]:
```
'C:\\Users\\jflieder'
```

[3]:
```
os.chdir(r"C:\Users\jflieder\Desktop\Code\Data Science Portfolio\Data Science␣
↪projects") #make sure I'm working in the same directory as the target dataset
os.getcwd()
```

[3]:
```
'C:\\Users\\jflieder\\Desktop\\Code\\Data Science Portfolio\\Data Science
projects'
```

[4]:
```
data = pd.read_csv("weight.csv")
data.head()
```

[4]:

	Unnamed: 0	Athlete	Bodyweight (kg)	Snatch (kg)	Clean & Jerk (kg)	Total (kg)	Ranking
0	0	Halil Mutlu (TUR)	55.62	137.5	167.5	305.0	1
1	1	Wu Wenxiong (CHN)	55.48	125.0	162.5	287.5	2
2	2	Zhang Xiangxiang (CHN)	55.94	125.0	162.5	287.5	3
3	3	Wang Shin-yuan (TPE)	55.38	125.0	160.0	285.0	4
4	4	Sergio Álvarez (CUB)	55.66	120.0	155.0	275.0	5

```
                                                  Url  \
0  https://en.wikipedia.org//wiki/Weightlifting_a…
1  https://en.wikipedia.org//wiki/Weightlifting_a…
2  https://en.wikipedia.org//wiki/Weightlifting_a…
3  https://en.wikipedia.org//wiki/Weightlifting_a…
4  https://en.wikipedia.org//wiki/Weightlifting_a…

                                                Title  Year Gender
0  Weightlifting at the 2000 Summer Olympics - Me…  2000    Men
1  Weightlifting at the 2000 Summer Olympics - Me…  2000    Men
2  Weightlifting at the 2000 Summer Olympics - Me…  2000    Men
3  Weightlifting at the 2000 Summer Olympics - Me…  2000    Men
4  Weightlifting at the 2000 Summer Olympics - Me…  2000    Men
```

[5]:
```
data.isnull().sum() #check for bad data entries
```

[5]:
```
Unnamed: 0           0
Athlete              0
Bodyweight (kg)      0
Snatch (kg)          0
Clean & Jerk (kg)    0
Total (kg)           0
Ranking              0
Url                  0
Title                0
Year                 0
Gender               0
dtype: int64
```

[6]:
```
data = data[['Athlete', 'Bodyweight (kg)', 'Snatch (kg)', 'Clean & Jerk (kg)',␣
 ↪'Total (kg)', 'Ranking', 'Year', 'Gender']]
data.rename(columns = {'Bodyweight (kg)':'BW', 'Snatch (kg)': 'SN', 'Clean &␣
 ↪Jerk (kg)': 'CJ', 'Total (kg)': 'Total'}, inplace = True)
data.head()
```

[6]:

	Athlete	BW	SN	CJ	Total	Ranking	Year	Gender
0	Halil Mutlu (TUR)	55.62	137.5	167.5	305.0	1	2000	Men
1	Wu Wenxiong (CHN)	55.48	125.0	162.5	287.5	2	2000	Men
2	Zhang Xiangxiang (CHN)	55.94	125.0	162.5	287.5	3	2000	Men
3	Wang Shin-yuan (TPE)	55.38	125.0	160.0	285.0	4	2000	Men
4	Sergio Álvarez (CUB)	55.66	120.0	155.0	275.0	5	2000	Men

[7]:
```
data.dtypes
```

[7]:
```
Athlete     object
BW         float64
```

```
SN          float64
CJ          float64
Total       float64
Ranking       int64
Year          int64
Gender       object
dtype: object
```

I would like to investigate proportions of bodyweight to the weights lifted, so I'll add some columns.

[8]:
```
data['SN_to_BW'] = data.apply(lambda row: row.SN / row.BW, axis = 1)
data['CJ_to_BW'] = data.apply(lambda row: row.CJ / row.BW, axis = 1)
data['Total_to_BW'] = data.apply(lambda row: row.Total / row.BW, axis = 1)
data.head()
```

[8]:

	Athlete	BW	SN	CJ	Total	Ranking	Year	Gender	\
0	Halil Mutlu (TUR)	55.62	137.5	167.5	305.0	1	2000	Men	
1	Wu Wenxiong (CHN)	55.48	125.0	162.5	287.5	2	2000	Men	
2	Zhang Xiangxiang (CHN)	55.94	125.0	162.5	287.5	3	2000	Men	
3	Wang Shin-yuan (TPE)	55.38	125.0	160.0	285.0	4	2000	Men	
4	Sergio Álvarez (CUB)	55.66	120.0	155.0	275.0	5	2000	Men	

	SN_to_BW	CJ_to_BW	Total_to_BW
0	2.472132	3.011507	5.483639
1	2.253064	2.928983	5.182048
2	2.234537	2.904898	5.139435
3	2.257133	2.889130	5.146262
4	2.155947	2.784765	4.940711

[9]:
```
data['SN'].value_counts()
```

[9]:
```
-1.0      88
 175.0    36
 155.0    34
 105.0    33
 140.0    32
          ..
 203.0     1
 169.0     1
 63.0      1
 89.0      1
 5.0       1
Name: SN, Length: 181, dtype: int64
```

It appears that an entry of -1 signifies a lack of a good lift for that event. Let's get a gender breakdown for that.

[10]:
```
data_men = data[data['Gender'] == 'Men']
data_women = data[data['Gender'] == 'Women']
men_SN_fail_count = data_men['SN'].value_counts()[-1]
men_CJ_fail_count = data_men['CJ'].value_counts()[-1]
women_SN_fail_count = data_women['SN'].value_counts()[-1]
women_CJ_fail_count = data_women['CJ'].value_counts()[-1]
```

[11]:
```
print('Of the male competitors,', men_SN_fail_count, 'failed to record a good␣
 ↪snatch lift and', men_CJ_fail_count, 'failed to record a good clean and jerk␣
 ↪lift.')
print('Of the female competitors,', women_SN_fail_count, 'failed to record a␣
 ↪good snatch lift and', women_CJ_fail_count, 'failed to record a good clean␣
 ↪and jerk lift.')
```

```
Of the male competitors, 64 failed to record a good snatch lift and 145 failed
to record a good clean and jerk lift.
Of the female competitors, 24 failed to record a good snatch lift and 52 failed
to record a good clean and jerk lift.
```

[12]:
```
men_any_fail_count = len(data_men[(data_men['SN'] == -1) | (data_men['CJ'] ==␣
 ↪-1)])
women_any_fail_count = len(data_women[(data_women['SN'] == -1) |␣
 ↪(data_women['CJ'] == -1)])
print('Of all male competitors,', men_any_fail_count, 'competitors failed to␣
 ↪record a good lift for either snatch or clean and jerk. This is', round(␣
 ↪((men_any_fail_count / len(data_men)) * 100), 2), 'percent of all male␣
 ↪competitors.')
print()
print('Of all female competitors,', women_any_fail_count, 'competitors failed␣
 ↪to record a good lift for either snatch or clean and jerk. This is', round(␣
 ↪((women_any_fail_count / len(data_women)) * 100), 2), 'percent of all female␣
 ↪competitors.')
```

```
Of all male competitors, 145 competitors failed to record a good lift for either
snatch or clean and jerk. This is 16.51 percent of all male competitors.

Of all female competitors, 52 competitors failed to record a good lift for
either snatch or clean and jerk. This is 9.29 percent of all female competitors.
```

I'm inferring from these numbers that if no snatch is completed, a competitor may not bother or be allowed to attempt a clean and jerk.

I'll now remove the rows with entries of -1 in the weight lifted columns. I could choose to leave them in, but I believe I'll get a better comparison of the weightclasses without those rows.

[13]:
```
print(len(data_men))
print(len(data_women))
```

```
878
560
```

[14]:
```
data_men = data_men[(data_men['SN'] != -1) & (data_men['CJ'] != -1)]
data_women = data_women[(data_women['SN'] != -1) & (data_women['CJ'] != -1)]
```

[15]:
```
data_men.head()
```

[15]:

	Athlete	BW	SN	CJ	Total	Ranking	Year	Gender	SN_to_BW	CJ_to_BW	Total_to_BW
0	Halil Mutlu (TUR)	55.62	137.5	167.5	305.0	1	2000	Men	2.472132	3.011507	5.483639
1	Wu Wenxiong (CHN)	55.48	125.0	162.5	287.5	2	2000	Men	2.253064	2.928983	5.182048
2	Zhang Xiangxiang (CHN)	55.94	125.0	162.5	287.5	3	2000	Men	2.234537	2.904898	5.139435
3	Wang Shin-yuan (TPE)	55.38	125.0	160.0	285.0	4	2000	Men	2.257133	2.889130	5.146262
4	Sergio Álvarez (CUB)	55.66	120.0	155.0	275.0	5	2000	Men	2.155947	2.784765	4.940711

[16]:
```
data_women.head()
```

[16]:

	Athlete	BW	SN	CJ	Total	Ranking	Year	Gender	SN_to_BW	CJ_to_BW	Total_to_BW
22	Tara Nott (USA)	47.48	82.5	102.5	185.0	1	2000	Women	1.737574	2.158804	3.896377
23	Raema Lisa Rumbewas (INA)	47.98	80.0	105.0	185.0	2	2000	Women	1.667361	2.188412	3.855773
24	Sri Indriyani (INA)	47.28	82.5	100.0	182.5	3	2000	Women	1.744924	2.115059	3.859983
25	Kay Thi Win (MYA)	47.48	80.0	100.0	180.0	4	2000	Women	1.684920	2.106150	3.791070
26	Robin Goad (USA)	47.66	77.5	100.0	177.5	5	2000	Women	1.626102	2.098196	3.724297

[17]:
```
print(len(data_men))
print(len(data_women))
```

```
733
508
```

I also need to remove rows where the bodyweight is entered at -1 for some reason. Otherwise, ratios of weights to bodyweight will throw off the data.

[18]:
```
data_men = data_men[(data_men['BW'] != -1)]
data_women = data_women[(data_women['BW'] != -1)]
```

[19]:
```
print(len(data_men))
print(len(data_women))
```

```
729
505
```

Now I'll make a boxplot of the ratio of average total weight lifted to bodyweight for each weight-class for both men and women. To do this, I'll just add in a column each to the data_men and data_women dataframes establishing the weightclass of the lifter of each row. This will be imperfect because the weightclass limits are not held consistent from 2000 through 2020.

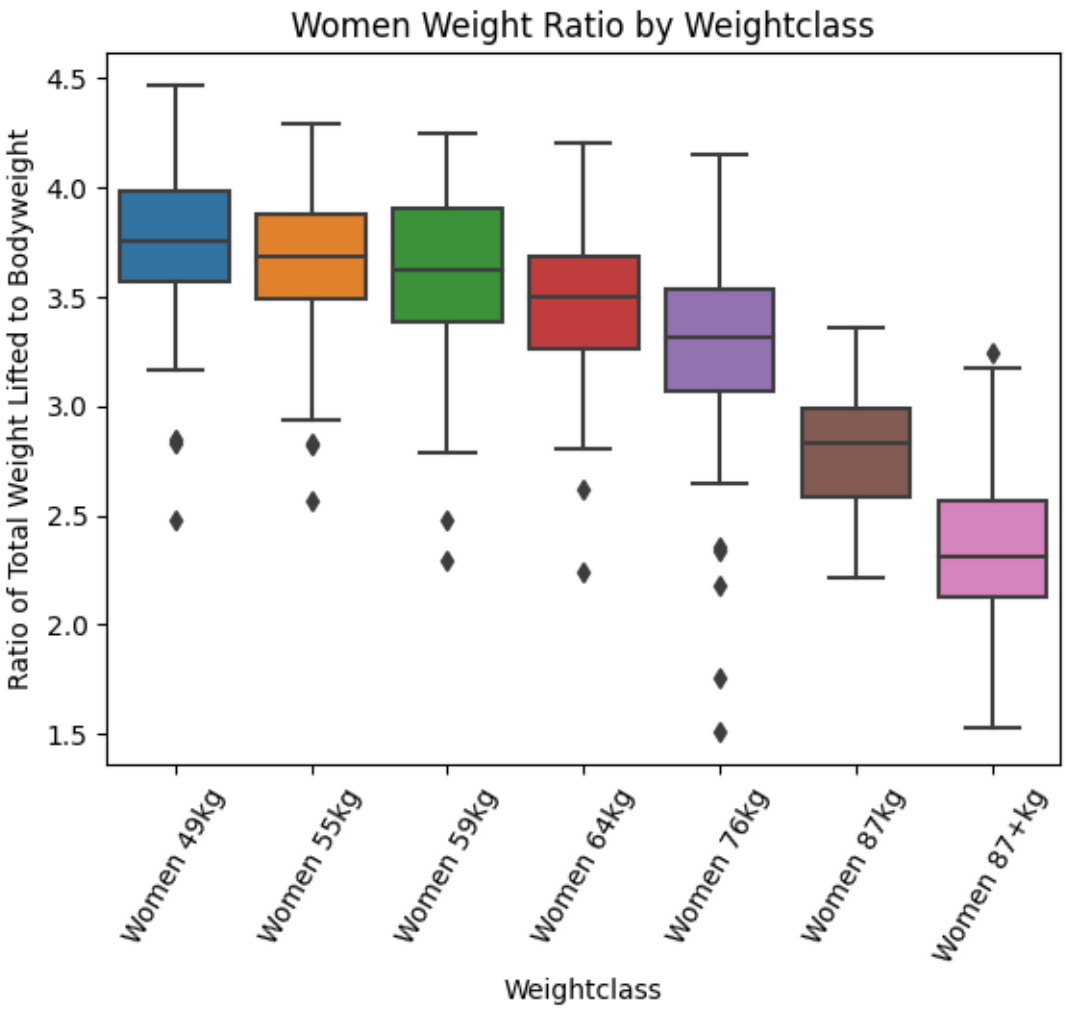
[20]:
```
data_men['Class'] = '' #create empty column for now
data_women['Class'] = ''
data_men.loc[data_men["BW"] < 56, "Class"] = 'Men 55kg'
data_men.loc[(data_men["BW"] >= 56) & (data_men["BW"] < 63), "Class"] = 'Men␣
 ↪62kg'
data_men.loc[(data_men["BW"] >= 63) & (data_men["BW"] < 70), "Class"] = 'Men␣
 ↪69kg'
data_men.loc[(data_men["BW"] >= 70) & (data_men["BW"] < 78), "Class"] = 'Men␣
 ↪77kg'
data_men.loc[(data_men["BW"] >= 78) & (data_men["BW"] < 86), "Class"] = 'Men␣
 ↪85kg'
data_men.loc[(data_men["BW"] >= 86) & (data_men["BW"] < 95), "Class"] = 'Men␣
 ↪94kg'
data_men.loc[(data_men["BW"] >= 95) & (data_men["BW"] < 106), "Class"] = 'Men␣
 ↪105kg'
data_men.loc[(data_men["BW"] >= 106), "Class"] = 'Men 105+kg'
print(data_men['Class'].value_counts())

data_women.loc[data_women["BW"] < 50, "Class"] = 'Women 49kg'
data_women.loc[(data_women["BW"] >= 50) & (data_women["BW"] < 56), "Class"] =␣
 ↪'Women 55kg'
data_women.loc[(data_women["BW"] >= 56) & (data_women["BW"] < 60), "Class"] =␣
 ↪'Women 59kg'
data_women.loc[(data_women["BW"] >= 60) & (data_women["BW"] < 65), "Class"] =␣
 ↪'Women 64kg'
data_women.loc[(data_women["BW"] >= 65) & (data_women["BW"] < 77), "Class"] =␣
 ↪'Women 76kg'
data_women.loc[(data_women["BW"] >= 77) & (data_women["BW"] < 88), "Class"] =␣
 ↪'Women 87kg'
data_women.loc[(data_women["BW"] >= 88), "Class"] = 'Women 87+kg'
print(data_women['Class'].value_counts())
```

```
Men 105+kg    110
Men 69kg      101
Men 77kg       97
Men 85kg       95
Men 94kg       91
```

```
Men 62kg       84
Men 55kg       76
Men 105kg      75
Name: Class, dtype: int64
Women 76kg     131
Women 59kg      88
Women 49kg      71
Women 55kg      67
Women 64kg      66
Women 87+kg     66
Women 87kg      16
Name: Class, dtype: int64
```

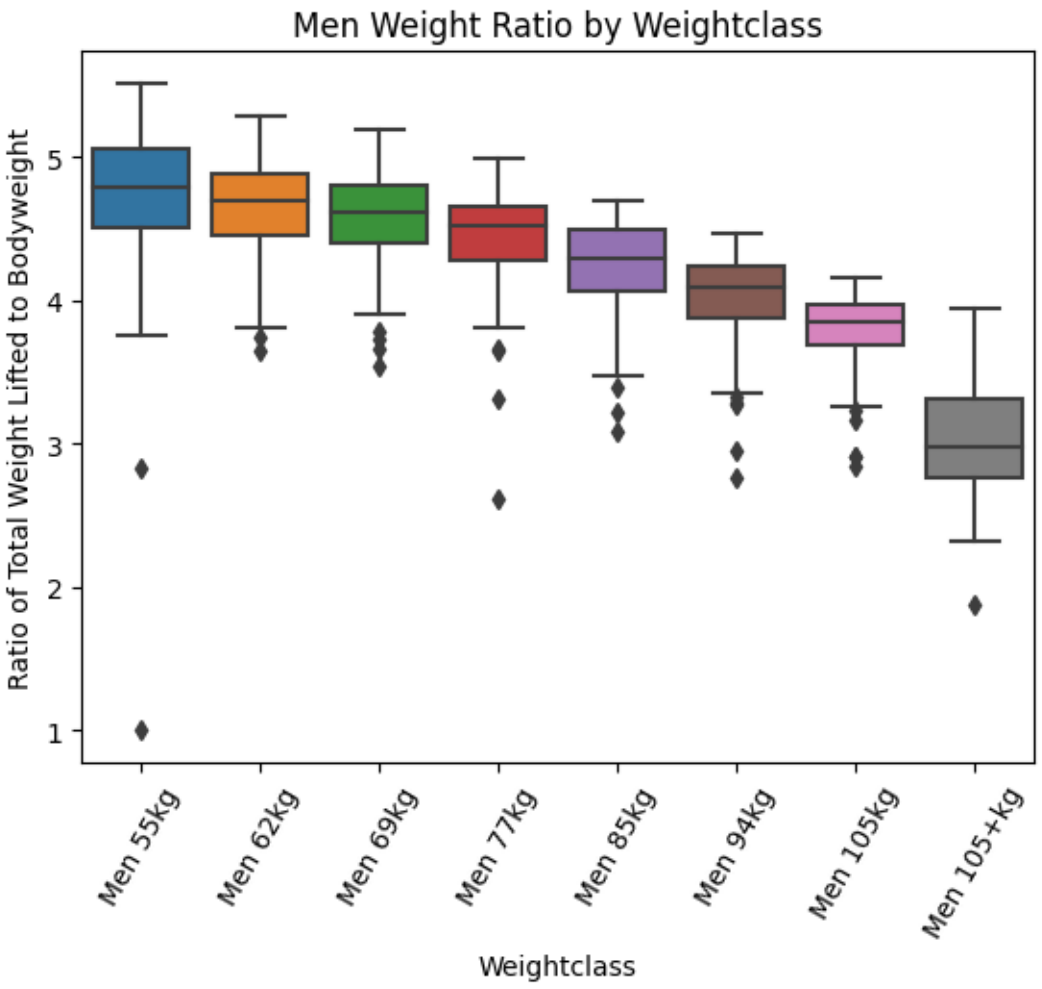
[21]:
```
ax = sns.boxplot(data = data_women, x = 'Class', y = 'Total_to_BW', order =␣
 ↪['Women 49kg', 'Women 55kg', 'Women 59kg', 'Women 64kg', 'Women 76kg',␣
 ↪'Women 87kg', 'Women 87+kg'])
ax.set_xticklabels(ax.get_xticklabels(),rotation=60)
ax.set(xlabel ="Weightclass", ylabel = "Ratio of Total Weight Lifted to␣
 ↪Bodyweight", title ='Women Weight Ratio by Weightclass')
```

[21]:
```
[Text(0.5, 0, 'Weightclass'),
 Text(0, 0.5, 'Ratio of Total Weight Lifted to Bodyweight'),
 Text(0.5, 1.0, 'Women Weight Ratio by Weightclass')]
```



[22]:
```
ax1 = sns.boxplot(data = data_men, x = 'Class', y = 'Total_to_BW', order =␣
↪['Men 55kg', 'Men 62kg', 'Men 69kg', 'Men 77kg', 'Men 85kg', 'Men 94kg',␣
↪'Men 105kg', 'Men 105+kg'])
ax1.set_xticklabels(ax1.get_xticklabels(),rotation=60)
ax1.set(xlabel ="Weightclass", ylabel = "Ratio of Total Weight Lifted to␣
↪Bodyweight", title ='Men Weight Ratio by Weightclass')
```

[22]:
```
[Text(0.5, 0, 'Weightclass'),
 Text(0, 0.5, 'Ratio of Total Weight Lifted to Bodyweight'),
 Text(0.5, 1.0, 'Men Weight Ratio by Weightclass')]
```



For both men and women, there is a general decrease in the ratio of the total weight lifted to bodyweight as the weightclass gets heavier.

Now I'll break the data apart in order to put back together aggregates of each weightclass into a new dataframe.

[23]:
```
data_men_55 = data_men[(data_men['BW'] < 56)]
data_men_62 = data_men[(data_men['BW'] < 63) & (data_men['BW'] >= 56)]
data_men_69 = data_men[(data_men['BW'] < 70) & (data_men['BW'] >= 63)]
data_men_77 = data_men[(data_men['BW'] < 78) & (data_men['BW'] >= 70)]
data_men_85 = data_men[(data_men['BW'] < 86) & (data_men['BW'] >= 78)]
data_men_94 = data_men[(data_men['BW'] < 95) & (data_men['BW'] >= 86)]
data_men_105 = data_men[(data_men['BW'] < 106) & (data_men['BW'] >= 94)]
data_men_heavy = data_men[(data_men['BW'] >= 106)]
data_women_49 = data_women[(data_women['BW'] < 50)]
data_women_55 = data_women[(data_women['BW'] < 56) & (data_women['BW'] >= 50)]
data_women_59 = data_women[(data_women['BW'] < 60) & (data_women['BW'] >= 56)]
```

```
data_women_64 = data_women[(data_women['BW'] < 65) & (data_women['BW'] >= 60)]
data_women_76 = data_women[(data_women['BW'] < 77) & (data_women['BW'] >= 65)]
data_women_87 = data_women[(data_women['BW'] < 88) & (data_women['BW'] >= 77)]
data_women_heavy = data_women[(data_women['BW'] >= 88)]
```

[24]:
```
categories = [[data_men_55], [data_men_62], [data_men_69], [data_men_77],␣
 ↪[data_men_85], [data_men_94], [data_men_105], \
[data_men_heavy], [data_women_49], [data_women_55], [data_women_59],␣
 ↪[data_women_64], [data_women_76], [data_women_87], \
[data_women_heavy]]

weightclasses = [['Men 55kg'], ['Men 62kg'], ['Men 69kg'], ['Men 77kg'], ['Men␣
 ↪85kg'], ['Men 94kg'], \
['Men 105kg'], ['Men 105kg+'], ['Women 49kg'], ['Women 55kg'], ['Women 59kg'],␣
 ↪['Women 64kg'], \
['Women 76kg'], ['Women 87kg'], ['Women 87kg+']]

columns_lst = ['Class', 'SN_to_BW_avg %', 'SN_to_BW_med %', 'SN_to_BW_max %',␣
 ↪'CJ_to_BW_avg %', 'CJ_to_BW_med %', 'CJ_to_BW_max %', 'Total_to_BW_avg %',␣
 ↪'Total_to_BW_med %', 'Total_to_BW_max %']

count = 0
for division in categories:
    weightclasses[count].append(categories[count][0]['SN_to_BW'].mean() * 100)
    weightclasses[count].append(categories[count][0]['SN_to_BW'].median() * 100)
    weightclasses[count].append(categories[count][0]['SN_to_BW'].max() * 100)
    weightclasses[count].append(categories[count][0]['CJ_to_BW'].mean() * 100)
    weightclasses[count].append(categories[count][0]['CJ_to_BW'].median() * 100)
    weightclasses[count].append(categories[count][0]['CJ_to_BW'].max() * 100)
    weightclasses[count].append(categories[count][0]['Total_to_BW'].mean() *␣
 ↪100)
    weightclasses[count].append(categories[count][0]['Total_to_BW'].median() *␣
 ↪100)
    weightclasses[count].append(categories[count][0]['Total_to_BW'].max() * 100)
    count += 1
```

This dataframe with show the ratio of competition weight lifted to body weight by percentage. It is usually the case that heavier competitors will be able to lift higher weights, so normalizing by bodyweight makes for a more interesting analysis.

[25]:
```
data_ratios = pd.DataFrame(data = weightclasses, columns= columns_lst)
data_ratios
```

[25]:

	Class	SN_to_BW_avg %	SN_to_BW_med %	SN_to_BW_max %	\
0	Men 55kg	209.982380	212.765062	247.213233	
1	Men 62kg	208.567223	210.543597	247.693055	
2	Men 69kg	206.387522	207.462687	239.408009	

```
3       Men 77kg      200.027236      202.086050      230.378758
4       Men 85kg      190.185981      194.437898      218.443736
5       Men 94kg      180.572967      185.821697      202.527544
6      Men 105kg      171.788052      176.308015      194.680851
7     Men 105kg+      136.370646      134.547340      179.063361
8     Women 49kg      165.887892      167.469123      206.524042
9     Women 55kg      162.389223      163.210445      190.621426
10    Women 59kg      158.637441      160.572930      193.355599
11    Women 64kg      154.074871      157.541809      184.472249
12    Women 76kg      146.385584      148.267961      185.857413
13    Women 87kg      124.027880      128.143253      146.152090
14   Women 87kg+      103.147620      103.232114      147.590656

    CJ_to_BW_avg %  CJ_to_BW_med %  CJ_to_BW_max %  Total_to_BW_avg %  \
0        261.797018      268.632547      305.316092         470.345833
1        257.925291      260.590337      288.336582         466.492514
2        252.502294      255.102041      283.512649         458.889816
3        243.018676      248.106555      280.876755         443.032316
4        232.469516      236.714404      257.536197         422.655497
5        219.698051      224.167378      249.144568         400.265117
6        206.583373      210.330659      232.978723         378.371425
7        166.150673      163.035189      217.431193         302.521319
8        208.887793      209.248797      246.523388         374.775685
9        204.146263      207.263738      248.576850         366.535487
10       199.934871      199.966804      239.334027         358.572312
11       189.809755      192.278425      235.803657         343.884626
12       181.469060      183.398995      229.417744         327.854644
13       155.500046      155.369024      189.540991         279.527926
14       129.750512      129.460277      176.913303         232.898132

    Total_to_BW_med %  Total_to_BW_max %
0          478.809234         551.364943
1          469.667013         529.383196
2          461.560862         519.773190
3          451.881755         499.656829
4          429.987608         469.973890
5          409.643695         446.963216
6          386.208225         427.659574
7          297.545840         394.495413
8          374.947633         446.691951
9          368.271955         428.898208
10         362.209404         424.284236
11         350.045316         420.275906
12         331.337325         415.275156
13         282.688897         335.693081
14         231.376100         324.503959
```

From this dataframe data_ratios, the maximums, averages, and medians of bodyweight-to-weight-

lifted ratios for each weightclass can be seen. I'll next quantify the marginal changes from each weightclass relative to the next lowest weightclass.

[26]:
```
data_ratios_men = data_ratios.iloc[0:8]
data_ratios_women = data_ratios.iloc[8:]
```

[27]:
```
columns_lst_marginal = ['SN_to_BW_avg % marginal change', 'SN_to_BW_med %␣
 ↪marginal change', 'SN_to_BW_max % marginal change', 'CJ_to_BW_avg % marginal␣
 ↪change', 'CJ_to_BW_med % marginal change', 'CJ_to_BW_max % marginal change',␣
 ↪'Total_to_BW_avg % marginal change', 'Total_to_BW_med % marginal change',␣
 ↪'Total_to_BW_max % marginal change']

weightclasses_men = ['Men 55kg', 'Men 62kg', 'Men 69kg', 'Men 77kg', 'Men␣
 ↪85kg', 'Men 94kg', \
'Men 105kg', 'Men 105kg+']

weightclasses_women = ['Women 49kg', 'Women 55kg', 'Women 59kg', 'Women 64kg', \
'Women 76kg', 'Women 87kg', 'Women 87kg+']

first_row_marginal = ['N/A', 'N/A', 'N/A', 'N/A', 'N/A', 'N/A', 'N/A', 'N/A',␣
 ↪'N/A']
data_marginal_men = pd.DataFrame([first_row_marginal], columns =␣
 ↪columns_lst_marginal, index = weightclasses_men)

for num in range(1, len(data_marginal_men)):
    new_row = data_ratios_men.iloc[num][1:] - data_ratios_men.iloc[num-1][1:]
    data_marginal_men.iloc[num] = new_row
data_marginal_men
```

[27]:

	SN_to_BW_avg % marginal change	SN_to_BW_med % marginal change	\
Men 55kg	N/A	N/A	
Men 62kg	-1.415157	-2.221465	
Men 69kg	-2.179702	-3.080911	
Men 77kg	-6.360286	-5.376637	
Men 85kg	-9.841255	-7.648152	
Men 94kg	-9.613014	-8.616201	
Men 105kg	-8.784915	-9.513682	
Men 105kg+	-35.417407	-41.760675	

	SN_to_BW_max % marginal change	CJ_to_BW_avg % marginal change	\
Men 55kg	N/A	N/A	
Men 62kg	0.479822	-3.871728	
Men 69kg	-8.285046	-5.422997	
Men 77kg	-9.029251	-9.483618	
Men 85kg	-11.935022	-10.54916	
Men 94kg	-15.916192	-12.771465	
Men 105kg	-7.846693	-13.114678	

```
Men 105kg+                      -15.61749                       -40.4327

           CJ_to_BW_med % marginal change CJ_to_BW_max % marginal change  \
Men 55kg                              N/A                             N/A
Men 62kg                        -8.042209                       -16.97951
Men 69kg                        -5.488296                       -4.823933
Men 77kg                        -6.995486                       -2.635894
Men 85kg                       -11.392151                      -23.340558
Men 94kg                       -12.547026                       -8.391629
Men 105kg                      -13.836719                      -16.165845
Men 105kg+                      -47.29547                      -15.547531

           Total_to_BW_avg % marginal change  \
Men 55kg                             N/A
Men 62kg                          -3.853319
Men 69kg                          -7.602698
Men 77kg                         -15.857499
Men 85kg                          -20.37682
Men 94kg                          -22.39038
Men 105kg                        -21.893691
Men 105kg+                       -75.850107

           Total_to_BW_med % marginal change Total_to_BW_max % marginal change
Men 55kg                             N/A                             N/A
Men 62kg                          -9.142221                       -21.981747
Men 69kg                          -8.106151                        -9.610006
Men 77kg                          -9.679107                       -20.116361
Men 85kg                         -21.894147                       -29.682939
Men 94kg                         -20.343913                       -23.010674
Men 105kg                         -23.43547                       -19.303642
Men 105kg+                       -88.662385                       -33.164162
```

[28]:
```
data_marginal_women = pd.DataFrame([first_row_marginal], columns =␣
 ↪columns_lst_marginal, index = weightclasses_women)

for num in range(1, len(data_marginal_women)):
    new_row = data_ratios_women.iloc[num][1:] - data_ratios_women.iloc[num-1][1:
 ↪]
    new_row.name = weightclasses_women[num]
    data_marginal_women.iloc[num] = new_row
data_marginal_women
```

[28]:
```
            SN_to_BW_avg % marginal change SN_to_BW_med % marginal change  \
Women 49kg                             N/A                             N/A
Women 55kg                       -3.498669                       -4.258677
Women 59kg                       -3.751782                       -2.637516
Women 64kg                        -4.56257                       -3.031121
```

Women 76kg	-7.689287	-9.273848
Women 87kg	-22.357704	-20.124707
Women 87kg+	-20.88026	-24.911139

	SN_to_BW_max % marginal change	CJ_to_BW_avg % marginal change	\
Women 49kg	N/A	N/A	
Women 55kg	-15.902616	-4.741529	
Women 59kg	2.734173	-4.211393	
Women 64kg	-8.88335	-10.125116	
Women 76kg	1.385164	-8.340695	
Women 87kg	-39.705323	-25.969014	
Women 87kg+	1.438566	-25.749535	

	CJ_to_BW_med % marginal change	CJ_to_BW_max % marginal change	\
Women 49kg	N/A	N/A	
Women 55kg	-1.985058	2.053462	
Women 59kg	-7.296934	-9.242823	
Women 64kg	-7.688379	-3.53037	
Women 76kg	-8.87943	-6.385914	
Women 87kg	-28.029971	-39.876752	
Women 87kg+	-25.908747	-12.627688	

	Total_to_BW_avg % marginal change	\
Women 49kg	N/A	
Women 55kg	-8.240198	
Women 59kg	-7.963175	
Women 64kg	-14.687686	
Women 76kg	-16.029982	
Women 87kg	-48.326717	
Women 87kg+	-46.629794	

	Total_to_BW_med % marginal change	\
Women 49kg	N/A	
Women 55kg	-6.675678	
Women 59kg	-6.06255	
Women 64kg	-12.164089	
Women 76kg	-18.70799	
Women 87kg	-48.648428	
Women 87kg+	-51.312797	

	Total_to_BW_max % marginal change
Women 49kg	N/A
Women 55kg	-17.793743
Women 59kg	-4.613972
Women 64kg	-4.00833
Women 76kg	-5.00075
Women 87kg	-79.582075

```
Women 87kg+                          -11.189122
```

These last two dataframes show for each weightclass and competition lift amount (snatch, clean & jerk, and total), what the percentage change of lift amount to bodyweight ratio is for that figure compared to that from the next lighter weightclass.

Now I'll give a barplot for these differences for total weight lifted to bodyweight average for each men and women.

[29]:
```
data_marginal_men_trimmed = data_marginal_men.iloc[1:] #to get rid of the blank␣
↪row of the lightest weightclass
data_marginal_men_trimmed.set_index([pd.Index(['55kg to 62kg', '62kg to 69kg',␣
↪'69kg to 77kg', '77kg to 85kg', '85kg to 94kg', '94kg to 105kg', '105kg to␣
↪unlimited'])], inplace = True)
data_marginal_men_trimmed
```

[29]:
```
                   SN_to_BW_avg % marginal change  \
55kg to 62kg                            -1.415157
62kg to 69kg                            -2.179702
69kg to 77kg                            -6.360286
77kg to 85kg                            -9.841255
85kg to 94kg                            -9.613014
94kg to 105kg                           -8.784915
105kg to unlimited                     -35.417407

                   SN_to_BW_med % marginal change  \
55kg to 62kg                            -2.221465
62kg to 69kg                            -3.080911
69kg to 77kg                            -5.376637
77kg to 85kg                            -7.648152
85kg to 94kg                            -8.616201
94kg to 105kg                           -9.513682
105kg to unlimited                     -41.760675

                   SN_to_BW_max % marginal change  \
55kg to 62kg                             0.479822
62kg to 69kg                            -8.285046
69kg to 77kg                            -9.029251
77kg to 85kg                           -11.935022
85kg to 94kg                           -15.916192
94kg to 105kg                           -7.846693
105kg to unlimited                      -15.61749

                   CJ_to_BW_avg % marginal change  \
55kg to 62kg                            -3.871728
62kg to 69kg                            -5.422997
69kg to 77kg                            -9.483618
77kg to 85kg                            -10.54916
```

```
85kg to 94kg                          -12.771465
94kg to 105kg                         -13.114678
105kg to unlimited                      -40.4327

                  CJ_to_BW_med % marginal change  \
55kg to 62kg                           -8.042209
62kg to 69kg                           -5.488296
69kg to 77kg                           -6.995486
77kg to 85kg                          -11.392151
85kg to 94kg                          -12.547026
94kg to 105kg                         -13.836719
105kg to unlimited                     -47.29547

                  CJ_to_BW_max % marginal change  \
55kg to 62kg                           -16.97951
62kg to 69kg                           -4.823933
69kg to 77kg                           -2.635894
77kg to 85kg                          -23.340558
85kg to 94kg                           -8.391629
94kg to 105kg                         -16.165845
105kg to unlimited                    -15.547531

                  Total_to_BW_avg % marginal change  \
55kg to 62kg                             -3.853319
62kg to 69kg                             -7.602698
69kg to 77kg                            -15.857499
77kg to 85kg                             -20.37682
85kg to 94kg                             -22.39038
94kg to 105kg                           -21.893691
105kg to unlimited                      -75.850107

                  Total_to_BW_med % marginal change  \
55kg to 62kg                             -9.142221
62kg to 69kg                             -8.106151
69kg to 77kg                             -9.679107
77kg to 85kg                            -21.894147
85kg to 94kg                            -20.343913
94kg to 105kg                            -23.43547
105kg to unlimited                      -88.662385

                  Total_to_BW_max % marginal change
55kg to 62kg                            -21.981747
62kg to 69kg                             -9.610006
69kg to 77kg                            -20.116361
77kg to 85kg                            -29.682939
85kg to 94kg                            -23.010674
94kg to 105kg                           -19.303642
```

```
105kg to unlimited                          -33.164162
```

[30]:
```
data_marginal_women_trimmed = data_marginal_women.iloc[1:] #to get rid of the␣
↪blank row of the lightest weightclass
data_marginal_women_trimmed.set_index([pd.Index(['49kg to 55kg', '55kg to␣
↪59kg', '59kg to 64kg', '64kg to 76kg', '76kg to 87kg', '87kg to␣
↪unlimited'])], inplace = True)
data_marginal_women_trimmed
```

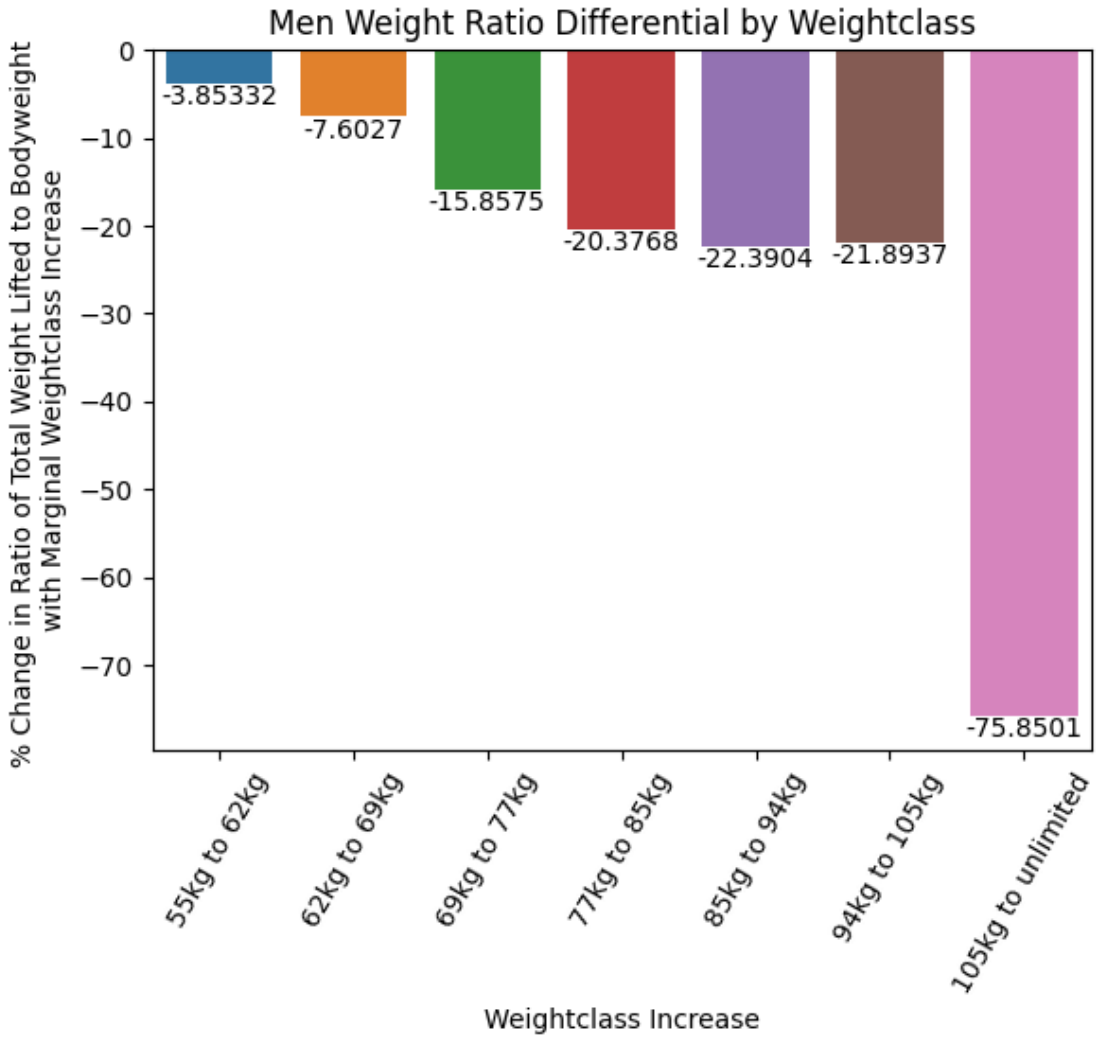
[30]:
```
                  SN_to_BW_avg % marginal change  \
49kg to 55kg                           -3.498669
55kg to 59kg                           -3.751782
59kg to 64kg                            -4.56257
64kg to 76kg                           -7.689287
76kg to 87kg                          -22.357704
87kg to unlimited                      -20.88026

                  SN_to_BW_med % marginal change  \
49kg to 55kg                           -4.258677
55kg to 59kg                           -2.637516
59kg to 64kg                           -3.031121
64kg to 76kg                           -9.273848
76kg to 87kg                          -20.124707
87kg to unlimited                     -24.911139

                  SN_to_BW_max % marginal change  \
49kg to 55kg                          -15.902616
55kg to 59kg                            2.734173
59kg to 64kg                            -8.88335
64kg to 76kg                            1.385164
76kg to 87kg                          -39.705323
87kg to unlimited                       1.438566

                  CJ_to_BW_avg % marginal change  \
49kg to 55kg                           -4.741529
55kg to 59kg                           -4.211393
59kg to 64kg                          -10.125116
64kg to 76kg                           -8.340695
76kg to 87kg                          -25.969014
87kg to unlimited                     -25.749535

                  CJ_to_BW_med % marginal change  \
49kg to 55kg                           -1.985058
55kg to 59kg                           -7.296934
59kg to 64kg                           -7.688379
64kg to 76kg                            -8.87943
76kg to 87kg                          -28.029971
```

```
87kg to unlimited                    -25.908747

                 CJ_to_BW_max % marginal change  \
49kg to 55kg                           2.053462
55kg to 59kg                          -9.242823
59kg to 64kg                           -3.53037
64kg to 76kg                          -6.385914
76kg to 87kg                         -39.876752
87kg to unlimited                    -12.627688

                 Total_to_BW_avg % marginal change  \
49kg to 55kg                           -8.240198
55kg to 59kg                           -7.963175
59kg to 64kg                          -14.687686
64kg to 76kg                          -16.029982
76kg to 87kg                          -48.326717
87kg to unlimited                     -46.629794

                 Total_to_BW_med % marginal change  \
49kg to 55kg                           -6.675678
55kg to 59kg                            -6.06255
59kg to 64kg                          -12.164089
64kg to 76kg                           -18.70799
76kg to 87kg                          -48.648428
87kg to unlimited                     -51.312797

                 Total_to_BW_max % marginal change
49kg to 55kg                          -17.793743
55kg to 59kg                           -4.613972
59kg to 64kg                            -4.00833
64kg to 76kg                            -5.00075
76kg to 87kg                          -79.582075
87kg to unlimited                     -11.189122
```

[31]:
```
ax2 = sns.barplot(data = data_marginal_men_trimmed, x =␣
 ↪data_marginal_men_trimmed.index, y = 'Total_to_BW_avg % marginal change')
ax2.set_xticklabels(ax2.get_xticklabels(),rotation=60) #to have the xtick␣
 ↪labels be more vertical
ax2.set(xlabel ="Weightclass Increase", ylabel = "% Change in Ratio of Total␣
 ↪Weight Lifted to Bodyweight \nwith Marginal Weightclass Increase", title␣
 ↪='Men Weight Ratio Differential by Weightclass')
ax2.bar_label(ax2.containers[0]) #to display the values
```

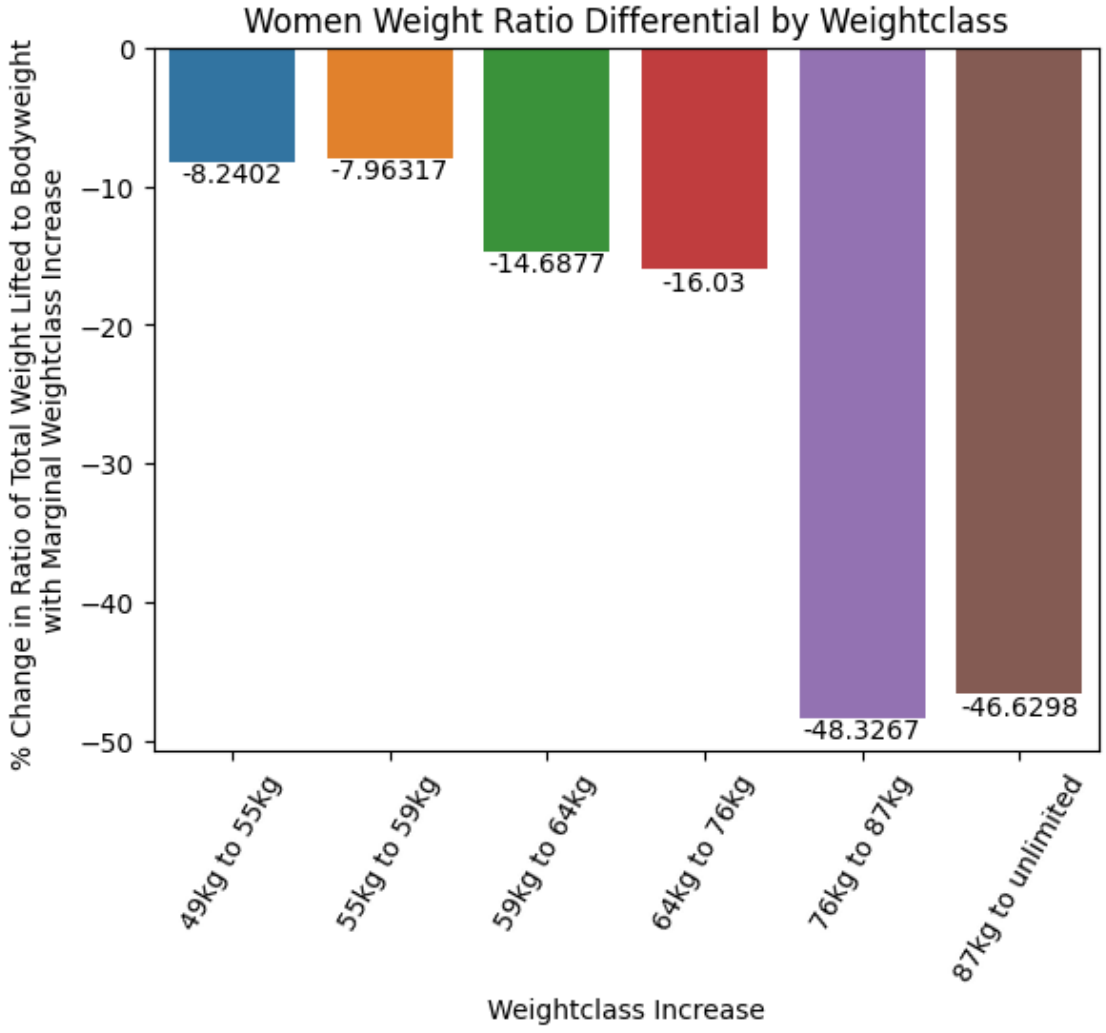
[31]:
```
[Text(0, 0, '-3.85332'),
 Text(0, 0, '-7.6027'),
 Text(0, 0, '-15.8575'),
 Text(0, 0, '-20.3768'),
```

```
Text(0, 0, '-22.3904'),
Text(0, 0, '-21.8937'),
Text(0, 0, '-75.8501')]
```



[32]:
```
ax3 = sns.barplot(data = data_marginal_women_trimmed, x =␣
 ↪data_marginal_women_trimmed.index, y = 'Total_to_BW_avg % marginal change')
ax3.set_xticklabels(ax3.get_xticklabels(),rotation=60) #to have the xtick␣
 ↪labels be more vertical
ax3.set(xlabel ="Weightclass Increase", ylabel = "% Change in Ratio of Total␣
 ↪Weight Lifted to Bodyweight \nwith Marginal Weightclass Increase", title␣
 ↪='Women Weight Ratio Differential by Weightclass')
ax3.bar_label(ax3.containers[0]) #to display the values
```

[32]: [Text(0, 0, '-8.2402'),
 Text(0, 0, '-7.96317'),
 Text(0, 0, '-14.6877'),
 Text(0, 0, '-16.03'),
 Text(0, 0, '-48.3267'),
 Text(0, 0, '-46.6298')]



More specific data can be found by, for example, isolating data from certain competition years. For now, this is a general exploration of the data.