Weightlifting

March 22, 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.

[1]:
```
# Pandas is a software library written for the Python programming language for␣
 ↪data manipulation and analysis.
import pandas as pd
# NumPy is a library for the Python programming language, adding support for␣
 ↪large, multi-dimensional arrays and matrices, along with a large collection␣
 ↪of high-level mathematical functions to operate on these arrays
import numpy as np
```

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

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

[3]:
```
os.chdir(r"C:\Users\jflieder\Desktop\Code\Data Science Portfolio\Data Science␣
 ↪projects")
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			
4	4	Sergio Álvarez (CUB)	55.66	120.0			

```
3              160.0       285.0        4
4              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()
```

[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	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

[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 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. Knowing the bodyweight limits of each weight class, I can divide this data set by weight class. I can further specify by gender. This will be imperfect because the weightclass limits are not held consistent from 2000 through 2020.

[13]:
```
data_men_55 = data_men[(data_men['BW'] < 56) & (data_men['BW'] > 10)]
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['BW'] > 10)]
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)]
```

[14]:
```
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.

[15]:
```
data_ratios = pd.DataFrame(data = weightclasses, columns= columns_lst)
data_ratios
```

[15]:

	Class	SN_to_BW_avg %	SN_to_BW_med %	SN_to_BW_max %	\
0	Men 55kg	191.761529	211.756006	247.213233	
1	Men 62kg	193.036333	209.846672	247.693055	
2	Men 69kg	189.934326	207.002904	239.408009	
3	Men 77kg	191.054864	201.796655	230.378758	
4	Men 85kg	171.350942	190.001188	218.443736	
5	Men 94kg	173.537062	185.703520	202.527544	
6	Men 105kg	150.246970	173.094145	190.912562	
7	Men 105kg+	131.785935	132.054490	179.063361	
8	Women 49kg	147.688090	166.736140	206.524042	
9	Women 55kg	158.319247	163.210445	192.380952	
10	Women 59kg	156.771028	160.572930	193.355599	
11	Women 64kg	147.602479	157.055412	184.472249	
12	Women 76kg	138.521347	147.532189	185.857413	
13	Women 87kg	123.843775	127.791563	146.152090	
14	Women 87kg+	101.978005	103.772896	147.590656	

	CJ_to_BW_avg %	CJ_to_BW_med %	CJ_to_BW_max %	Total_to_BW_avg %	\
0	210.238738	260.611853	305.316092	378.852674	
1	205.541189	255.933534	288.336582	372.144935	
2	205.396237	249.142690	283.512649	373.502203	
3	214.144160	244.890014	280.876755	390.519825	
4	185.344778	230.851190	257.536197	337.173922	
5	197.838920	222.293470	249.144568	360.528058	
6	159.324890	205.300381	225.800305	291.966887	
7	148.516111	160.697888	217.431193	270.473599	
8	178.386348	203.076923	246.523388	320.290448	
9	192.538343	205.988938	248.576850	345.778878	
10	187.062503	198.646605	239.334027	335.574149	
11	171.455265	191.974823	235.803657	310.755833	
12	159.375983	181.183734	229.417744	288.077403	
13	146.285255	155.208094	189.540991	263.017377	
14	122.286979	128.186354	176.913303	219.540450	

	Total_to_BW_med %	Total_to_BW_max %
0	466.329708	551.364943
1	461.988339	529.383196
2	453.080900	519.773190
3	445.767470	499.656829
4	417.555082	469.973890
5	405.580792	446.963216
6	375.403439	416.189385
7	290.160907	394.495413
8	368.432070	446.691951

9	367.339170	428.898208
10	355.027680	424.284236
11	345.816733	420.275906
12	327.288184	415.275156
13	280.254777	335.693081
14	229.989997	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.

[16]:
```
data_ratios_men = data_ratios.iloc[0:8]
data_ratios_women = data_ratios.iloc[8:]
```

[17]:
```
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
```

[17]:

	SN_to_BW_avg % marginal change	SN_to_BW_med % marginal change	\
Men 55kg	N/A	N/A	
Men 62kg	1.274803	-1.909334	
Men 69kg	-3.102007	-2.843768	
Men 77kg	1.120538	-5.206249	
Men 85kg	-19.703922	-11.795467	
Men 94kg	2.18612	-4.297667	
Men 105kg	-23.290091	-12.609376	
Men 105kg+	-18.461036	-41.039655	

```
           SN_to_BW_max % marginal change CJ_to_BW_avg % marginal change  \
Men 55kg                              N/A                           N/A
Men 62kg                         0.479822                      -4.69755
Men 69kg                        -8.285046                     -0.144951
Men 77kg                        -9.029251                      8.747923
Men 85kg                       -11.935022                     -28.799382
Men 94kg                       -15.916192                      12.494142
Men 105kg                      -11.614982                      -38.51403
Men 105kg+                     -11.849201                     -10.80878

           CJ_to_BW_med % marginal change CJ_to_BW_max % marginal change  \
Men 55kg                              N/A                           N/A
Men 62kg                        -4.678319                     -16.97951
Men 69kg                        -6.790844                     -4.823933
Men 77kg                        -4.252676                     -2.635894
Men 85kg                       -14.038824                     -23.340558
Men 94kg                        -8.55772                      -8.391629
Men 105kg                      -16.993089                     -23.344263
Men 105kg+                     -44.602493                     -8.369112

           Total_to_BW_avg % marginal change  \
Men 55kg                              N/A
Men 62kg                        -6.707739
Men 69kg                         1.357268
Men 77kg                        17.017622
Men 85kg                       -53.345903
Men 94kg                        23.354136
Men 105kg                      -68.561171
Men 105kg+                     -21.493289

           Total_to_BW_med % marginal change Total_to_BW_max % marginal change
Men 55kg                              N/A                               N/A
Men 62kg                        -4.341369                         -21.981747
Men 69kg                        -8.907439                          -9.610006
Men 77kg                         -7.31343                         -20.116361
Men 85kg                       -28.212389                         -29.682939
Men 94kg                        -11.97429                         -23.010674
Men 105kg                      -30.177353                         -30.773831
Men 105kg+                     -85.242531                         -21.693972
```

[18]:
```
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
```

[18]:

	SN_to_BW_avg % marginal change	SN_to_BW_med % marginal change	\
Women 49kg	N/A	N/A	
Women 55kg	10.631157	-3.525695	
Women 59kg	-1.548219	-2.637516	
Women 64kg	-9.16855	-3.517518	
Women 76kg	-9.081131	-9.523224	
Women 87kg	-14.677572	-19.740626	
Women 87kg+	-21.865771	-24.018668	

	SN_to_BW_max % marginal change	CJ_to_BW_avg % marginal change	\
Women 49kg	N/A	N/A	
Women 55kg	-14.143089	14.151996	
Women 59kg	0.974646	-5.475841	
Women 64kg	-8.88335	-15.607237	
Women 76kg	1.385164	-12.079282	
Women 87kg	-39.705323	-13.090729	
Women 87kg+	1.438566	-23.998275	

	CJ_to_BW_med % marginal change	CJ_to_BW_max % marginal change	\
Women 49kg	N/A	N/A	
Women 55kg	2.912015	2.053462	
Women 59kg	-7.342332	-9.242823	
Women 64kg	-6.671782	-3.53037	
Women 76kg	-10.791089	-6.385914	
Women 87kg	-25.97564	-39.876752	
Women 87kg+	-27.021739	-12.627688	

	Total_to_BW_avg % marginal change	\
Women 49kg	N/A	
Women 55kg	25.48843	
Women 59kg	-10.204728	
Women 64kg	-24.818316	
Women 76kg	-22.678431	
Women 87kg	-25.060025	
Women 87kg+	-43.476927	

	Total_to_BW_med % marginal change	\
Women 49kg	N/A	
Women 55kg	-1.0929	
Women 59kg	-12.31149	
Women 64kg	-9.210947	
Women 76kg	-18.528549	
Women 87kg	-47.033407	
Women 87kg+	-50.26478	

	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.

More specific data can be found based by looking at certain competition years, honing in on a certain weightclass, etc. For now, this is a general exploration of the data.