```
# Import the google drive folders that contain the data
from google.colab import drive
drive.mount('/content/drive')
```

```
     Mounted at /content/drive
```

```
%cd /content/drive/MyDrive/DSC680/Weeks5-8/Week8/datasets/
```

```
     /content/drive/MyDrive/DSC680/Weeks5-8/Week8/datasets
```

```
%ls
```

```
     meets.csv             openpowerlifting_full-cleaned.csv  pml-training_full.csv
     megaGymDataset.csv    openpowerlifting_short.csv
     openpowerlifting.csv  pml-testing.csv
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df1 = pd.read_csv('./meets.csv')
df1.head()
```

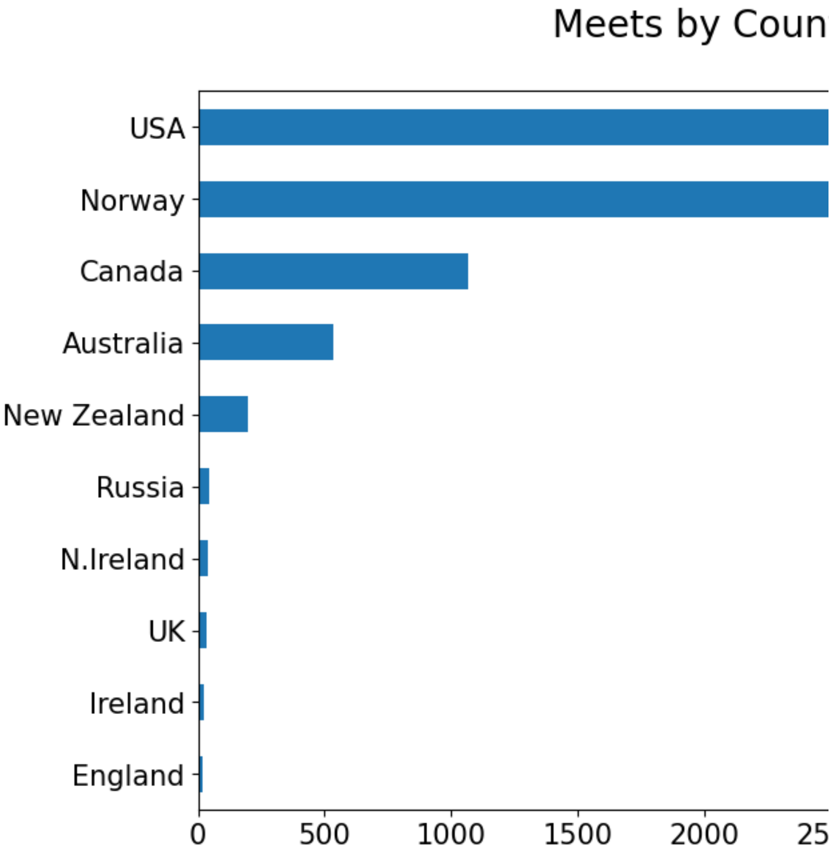
	MeetID	MeetPath	Federation	Date	MeetCountry	MeetState	MeetTown	M
0	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte	201 ' Pov
1	1	365strong/1602	365Strong	2016-11-19	USA	MO	Ozark	Than Pov

```
df1.describe()
```

	MeetID
count	8482.000000
mean	4240.500000
std	2448.686825
min	0.000000
25%	2120.250000
50%	4240.500000
75%	6360.750000
max	8481.000000

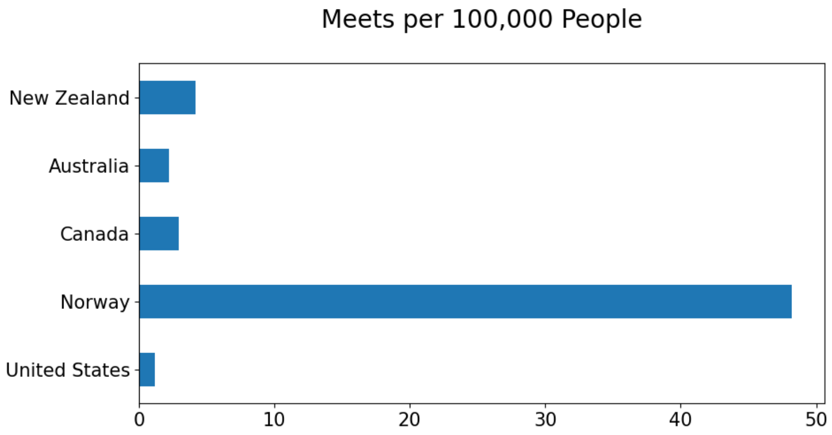
```
df1['Date'] = pd.to_datetime(df1['Date'])
df1['Month'] = df1['Date'].apply(lambda x:x.month)
df1['Year'] = df1['Date'].apply(lambda x:x.year)
```

```
plt.figure(figsize=(10,7))
df1['MeetCountry'].value_counts()[:10].sort_values(ascending=True).plot(kind='barh')
plt.title('Meets by Country\n',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
print('Meets by Country:\n')
print(df1['MeetCountry'].value_counts()[:10])
```



```
Meets by Country:

USA            3894
Norway         2521
Canada         1066
Australia       532
New Zealand     195
Russia           43
```

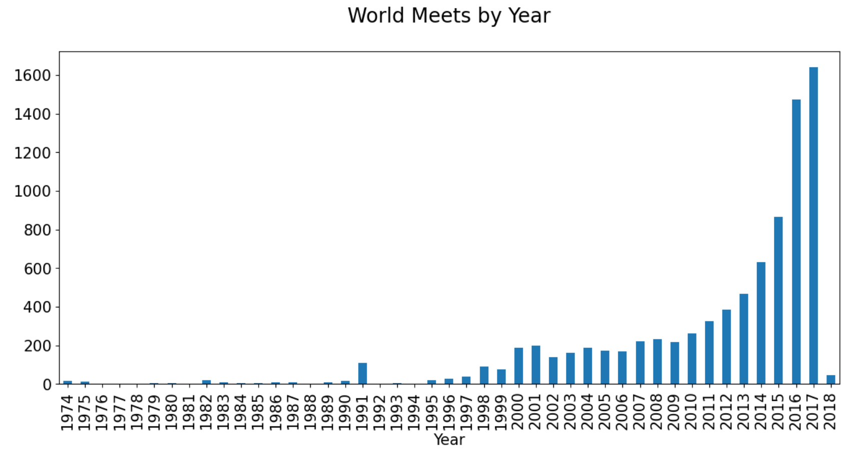
```
#Current country populations, as of April 12, 2018

usPop = 325700000
norwayPop = 5230000
canadaPop = 36290000
aussiePop = 24130000
kiwiPop = 4690000


perCapDict = {'United States': len(df1[df1['MeetCountry']=='USA']) / usPop * 100000,
            'Norway': len(df1[df1['MeetCountry']=='Norway']) / norwayPop * 100000,
            'Canada': len(df1[df1['MeetCountry']=='Canada']) / canadaPop * 100000,
            'Australia': len(df1[df1['MeetCountry']=='Australia']) / aussiePop * 100000,
            'New Zealand': len(df1[df1['MeetCountry']=='New Zealand']) / kiwiPop * 100000}

perCapDf = pd.Series(perCapDict)


plt.figure(figsize=(10,5))
perCapDf.plot(kind='barh')
plt.title('Meets per 100,000 People\n',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
print('Meets per 100,000 People\n')
for k,v in perCapDict.items():
    print(k[:6] + ': ',(round(v,2)))
```



```
Meets per 100,000 People

United:  1.2
Norway:  48.2
Canada:  2.94
Austra:  2.2
New Ze:  4.16
```

```
plt.figure(figsize=(10,6))
df1.groupby(['Month'])['Month'].count().plot(kind='bar')
plt.title('World Meets by Month\n',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
print(df1.groupby(['Month'])['Month'].count())
```

```
plt.figure(figsize=(14,6))
df1.groupby(['Year'])['Year'].count().plot(kind='bar')
plt.title('World Meets by Year\n',fontsize=20)
plt.xlabel('Year',fontsize=15)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



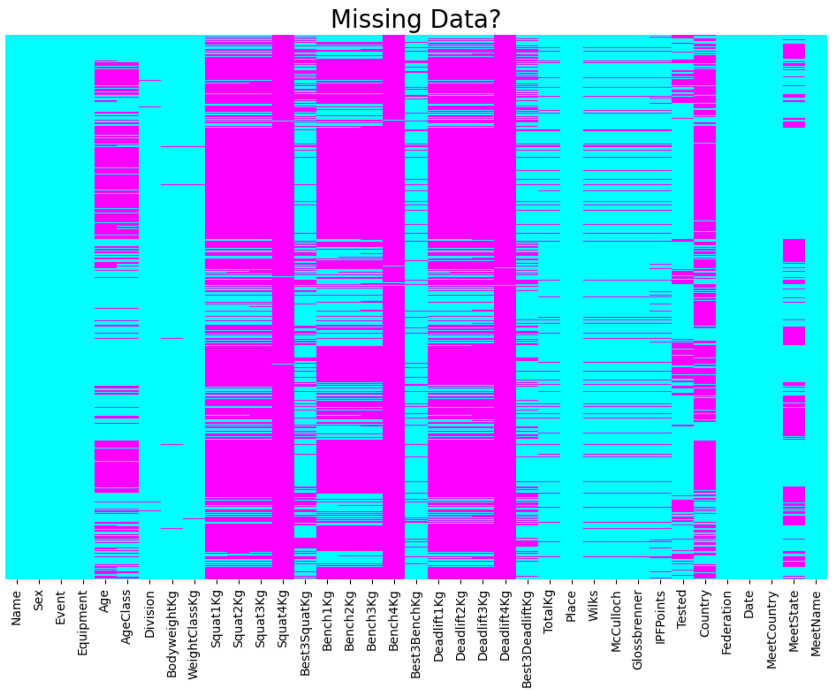
```
df = pd.read_csv('./openpowerlifting.csv')
```

```
<ipython-input-14-6d0c17c0d39f>:1: DtypeWarning: Columns (35) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('./openpowerlifting.csv')
```

```
df.head()
```

	Name	Sex	Event	Equipment	Age	AgeClass	Division	BodyweightKg	Weigl
0	Abbie Murphy	F	SBD	Wraps	29.0	24-34	F-OR	59.8	
1	Abbie Tuong	F	SBD	Wraps	29.0	24-34	F-OR	58.5	
2	Ainslee Hooper	F	B	Raw	40.0	40-44	F-OR	55.4	
3	Amy Moldenhauer	F	SBD	Wraps	23.0	20-23	F-OR	60.0	

```
plt.figure(figsize=(12,8))
sns.heatmap(df.isnull(),cmap='cool',cbar=False,yticklabels=False)
plt.title('Missing Data?',fontsize=20)
plt.show()
```



```
df.drop(['Squat4Kg','Bench4Kg','Deadlift4Kg'],axis=1,inplace=True)
```

```
df.head()
```

1 to 5 of 5 entries Filter

×

index:
to

Name:

Sex:

Event:

Equipment:

Age:
to

AgeClass:

Division:

```
df['Name'].value_counts()[:10]
```

```
Alan Aerts          214
Jose Hernandez      204
Sverre Paulsen      191
Erik Rasmussen      186
Bonnie Aerts        165
Zbyněk Krejča       154
Bjørn Grønvold      154
```

```
Jackie Blasbery    150
Max Bristow        147
Hana Takáčová      145
Name: Name, dtype: int64
```

```
print('Number of unique divisions: ' + str(df['Division'].nunique()))
```

```
Number of unique divisions: 4842
```

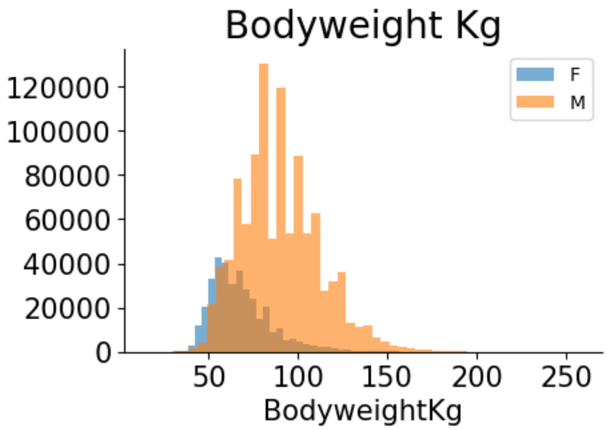
```
def age_class(x):
    if x < 13:
        return 'CHILD'
    if x >= 13 and x <= 17:
        return 'YOUTH'
    if x >= 18 and x <= 34:
        return 'ADULT'
    if x >= 35:
        return 'MASTERS'
```

```
df['AgeClass'] = df['Age'].apply(age_class)
```

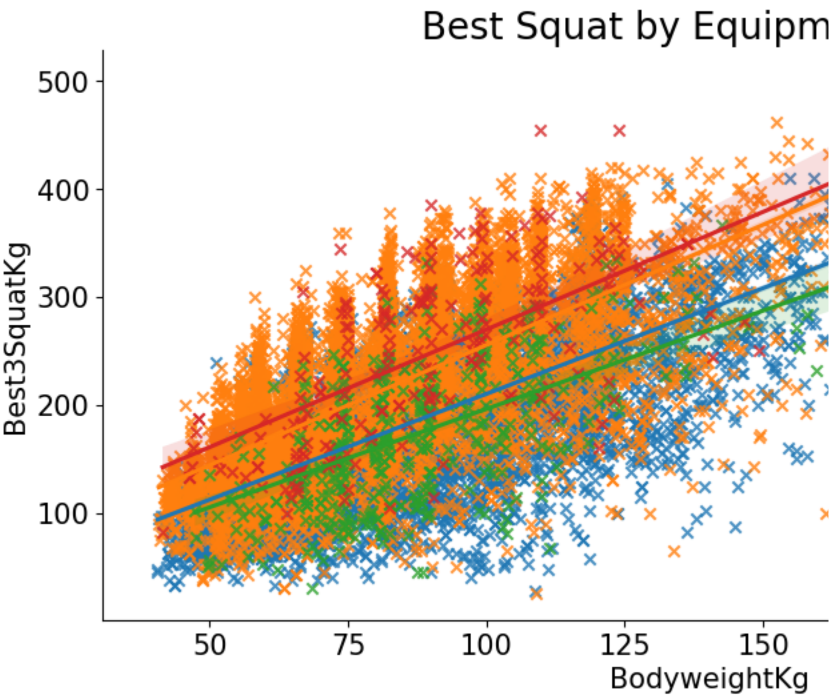
```
def squatBody(x):
    return x['Best3SquatKg'] / x['BodyweightKg']

def benchBody(x):
    return x['Best3BenchKg'] / x['BodyweightKg']

def deadliftBody(x):
    return x['Best3DeadliftKg'] / x['BodyweightKg']

def totalLiftBody(x):
    return x['TotalKg'] / x['BodyweightKg']
```

```
df['Squat / BW'] = df.apply(squatBody,axis=1)
df['Bench / BW'] = df.apply(benchBody,axis=1)
df['Deadlift / BW'] = df.apply(deadliftBody,axis=1)
df['Total / BW'] = df.apply(totalLiftBody,axis=1)
```

```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'BodyweightKg',bins=50,alpha=.6)
plt.title('Bodyweight Kg',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.legend(loc=1)
plt.show()
```

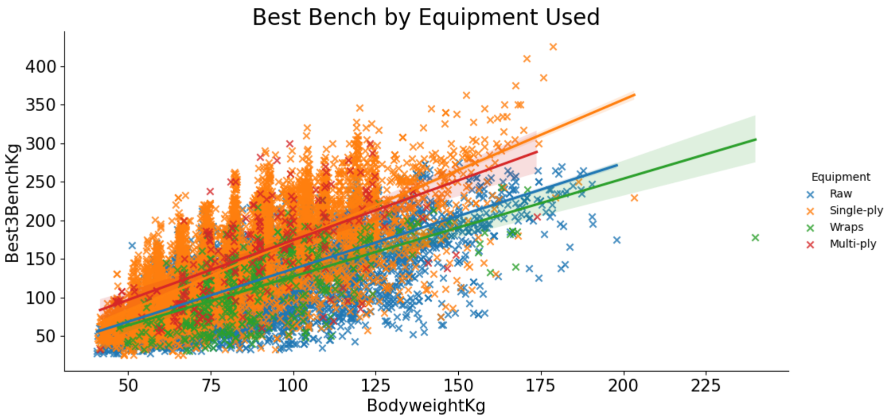


```
sns.lmplot(x='BodyweightKg',
           y='Best3SquatKg',
           data=df.dropna(),
           hue='Equipment',
           markers='x',
           aspect=2)
plt.title('Best Squat by Equipment Used',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('Best3SquatKg',fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.show()
print('Equipment Used by Lifters:\n')
print(df['Equipment'].dropna().value_counts())
```



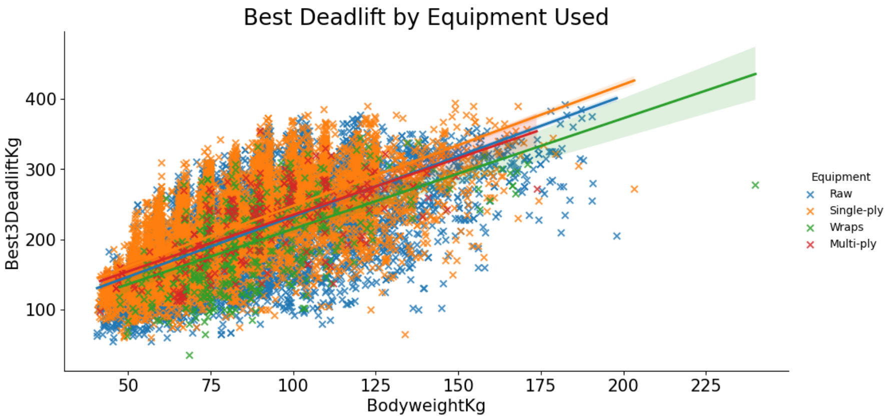
```
Equipment Used by Lifters:

Single-ply    787141
Raw           467421
Wraps         103739
Multi-ply      65035
Straps            18
Name: Equipment, dtype: int64
```

```
sns.lmplot(x='BodyweightKg',
           y='Best3BenchKg',
           data=df.dropna(),
           hue='Equipment',
           markers='x',
           aspect=2)
plt.title('Best Bench by Equipment Used',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('Best3BenchKg',fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.show()
print('Equipment Used by Lifters:\n')
print(df['Equipment'].dropna().value_counts())
```



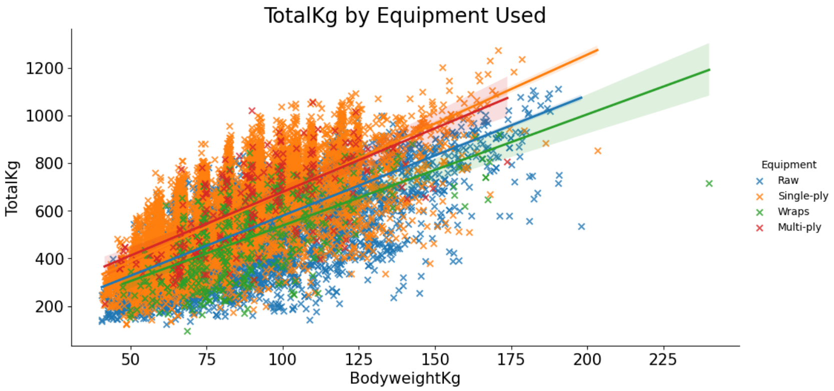
```
Equipment Used by Lifters:

Single-ply    787141
Raw           467421
Wraps         103739
Multi-ply      65035
```

```
sns.lmplot(x='BodyweightKg',
           y='Best3DeadliftKg',
           data=df.dropna(),
           hue='Equipment',
           markers='x',
           aspect=2)
plt.title('Best Deadlift by Equipment Used',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('Best3DeadliftKg',fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.show()
print('Equipment Used by Lifters:\n')
print(df['Equipment'].dropna().value_counts())
```



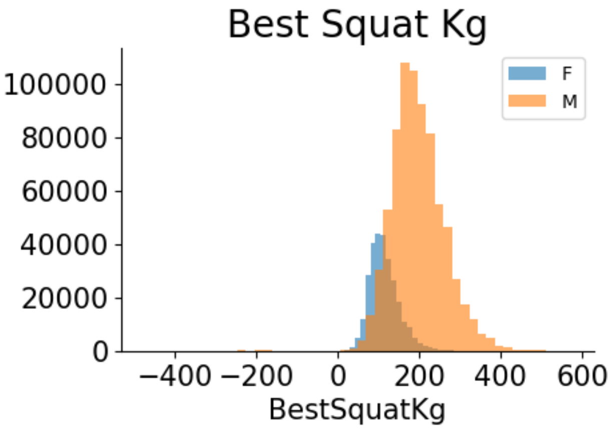
```
Equipment Used by Lifters:

Single-ply    787141
Raw           467421
Wraps         103739
Multi-ply      65035
Straps            18
Name: Equipment, dtype: int64
```

```
sns.lmplot(x='BodyweightKg',
           y='TotalKg',
           data=df.dropna(),
           hue='Equipment',
           markers='x',
           aspect=2)
plt.title('TotalKg by Equipment Used',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('TotalKg',fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.show()
print('Equipment Used by Lifters:\n')
print(df['Equipment'].dropna().value_counts())
```

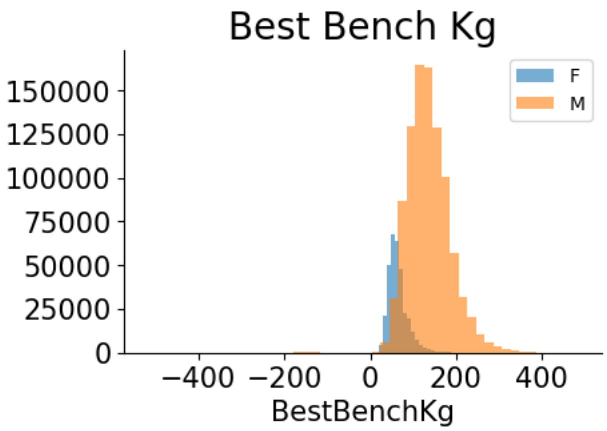


```
Equipment Used by Lifters:

Single-ply    787141
Raw           467421
Wraps         103739
Multi-ply      65035
Straps            18
Name: Equipment, dtype: int64
```

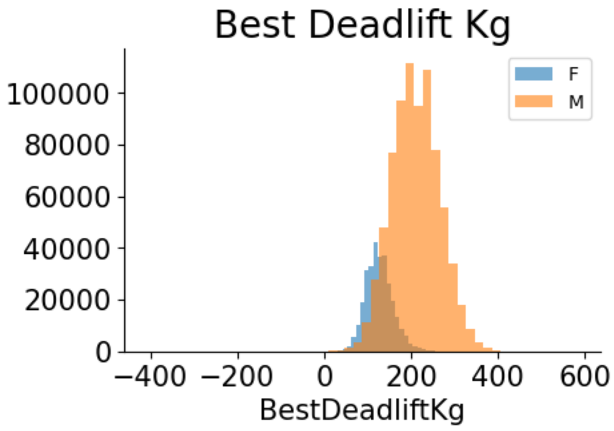
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Best3SquatKg',bins=50,alpha=.6)
plt.title('Best Squat Kg',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('BestSquatKg',fontsize=15)
plt.legend(loc=1)
plt.show()
```



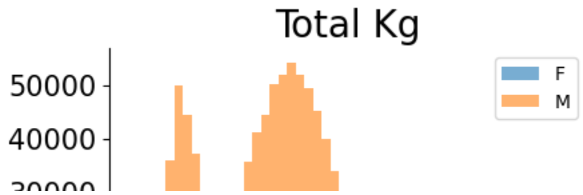
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Best3BenchKg',bins=50,alpha=.6)
plt.title('Best Bench Kg',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('BestBenchKg',fontsize=15)
plt.legend(loc=1)
plt.show()
```



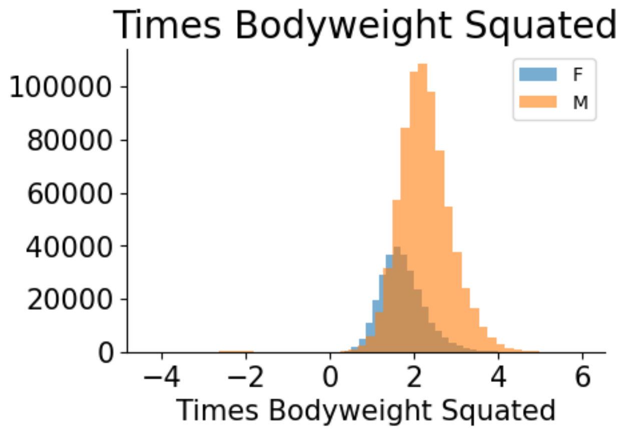
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Best3DeadliftKg',bins=50,alpha=.6)
plt.title('Best Deadlift Kg',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('BestDeadliftKg',fontsize=15)
plt.legend(loc=1)
plt.show()
```



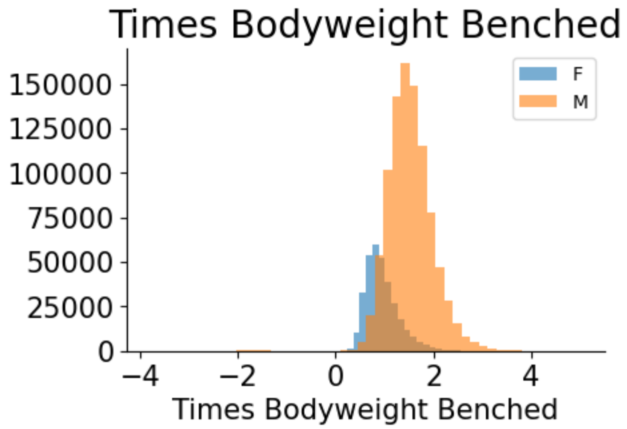
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'TotalKg',bins=50,alpha=.6)
plt.title('Total Kg',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('TotalKg',fontsize=15)
plt.legend(loc=1)
plt.show()
```



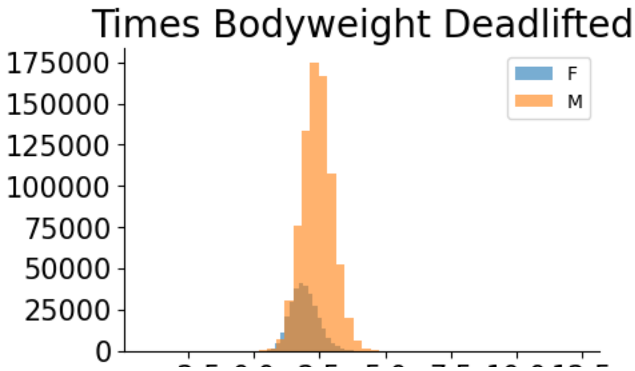
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Squat / BW',bins=50,alpha=.6)
plt.title('Times Bodyweight Squated',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('Times Bodyweight Squated',fontsize=15)
plt.legend(loc=1)
plt.show()
```



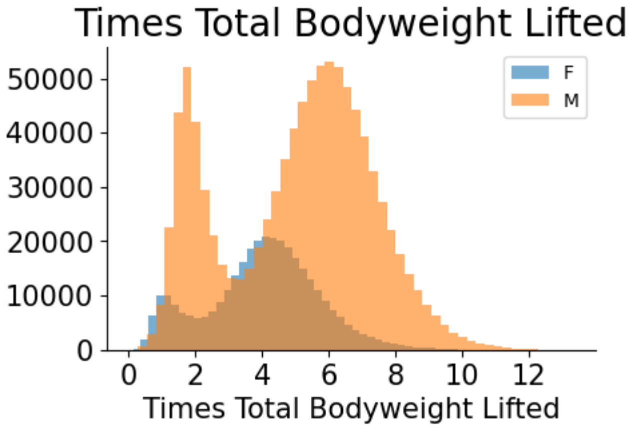
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Bench / BW',bins=50,alpha=.6)
plt.title('Times Bodyweight Benched',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('Times Bodyweight Benched',fontsize=15)
plt.legend(loc=1)
plt.show()
```



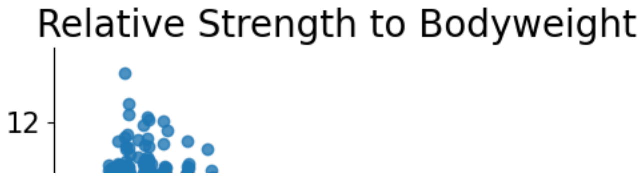
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Deadlift / BW',bins=50,alpha=.6)
plt.title('Times Bodyweight Deadlifted',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('Times Bodyweight Deadlifted',fontsize=15)
plt.legend(loc=1)
plt.show()
```



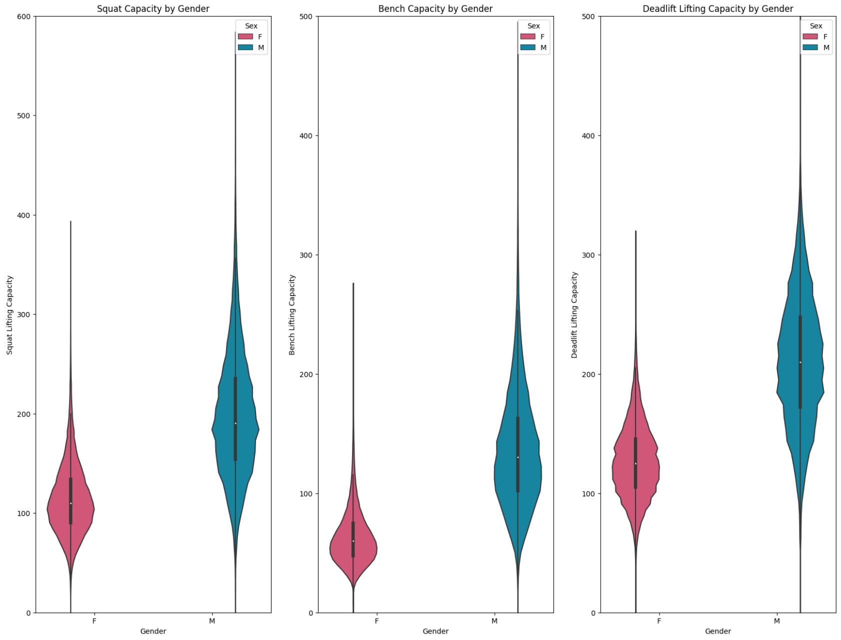
```
g = sns.FacetGrid(df,hue='Sex',aspect=1.5,legend_out=True)
g = g.map(plt.hist,'Total / BW',bins=50,alpha=.6)
plt.title('Times Total Bodyweight Lifted',fontsize=20)
plt.yticks(fontsize=15)
plt.xticks(fontsize=15)
plt.xlabel('Times Total Bodyweight Lifted',fontsize=15)
plt.legend(loc=1)
plt.show()
```



```
sns.lmplot(data=df.dropna(),
           x='BodyweightKg',
           y='Total / BW',
           hue='Sex')
plt.title('Relative Strength to Bodyweight',fontsize=20)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.ylabel('Total / BW',fontsize=15)
plt.xlabel('BodyweightKg',fontsize=15)
plt.show()
```



```
import seaborn as sns
plt.figure(figsize = (20,15))

plt.subplot(1,3,1)

plt.ylim(0,600)
sns.violinplot(data = df, x = 'Sex', y = 'Best3SquatKg',hue = 'Sex', scale = 'count',dodge = True, palette = ['#e54370','#0093b7'])
plt.style.use("fast")
plt.title('Squat Capacity by Gender')
plt.xlabel('Gender')
plt.ylabel('Squat Lifting Capacity')


plt.subplot(1,3,2)
plt.ylim(0,500)
plt.style.use("fast")
sns.violinplot(data = df, x = 'Sex', y = 'Best3BenchKg',hue = 'Sex',scale = 'count',dodge = True, palette = ['#e54370','#0093b7'])
plt.xlabel('Gender')
plt.ylabel('Bench Lifting Capacity')
plt.title('Bench Capacity by Gender')


plt.subplot(1,3,3)
plt.ylim(0,500)
plt.style.use("fast")
sns.violinplot(data = df, x = 'Sex', y = 'Best3DeadliftKg',hue = 'Sex',scale = 'count',dodge = True, palette = ['#e54370','#0093b7'])
plt.xlabel('Gender')
plt.ylabel('Deadlift Lifting Capacity')
plt.title('Deadlift Lifting Capacity by Gender')


plt.show()
```



```
df.shape
```

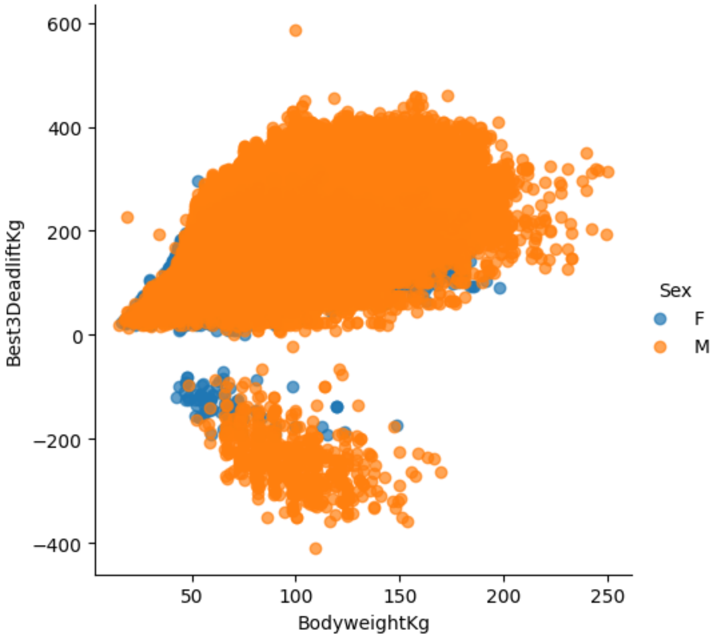
```
(1423354, 38)
```

```
df.dtypes
```

```
Name                object
Sex                 object
Event               object
Equipment           object
Age                float64
AgeClass            object
Division            object
BodyweightKg       float64
WeightClassKg       object
Squat1Kg           float64
Squat2Kg           float64
Squat3Kg           float64
Best3SquatKg       float64
Bench1Kg           float64
Bench2Kg           float64
Bench3Kg           float64
Best3BenchKg       float64
Deadlift1Kg        float64
Deadlift2Kg        float64
Deadlift3Kg        float64
Best3DeadliftKg    float64
TotalKg            float64
Place               object
Wilks              float64
McCulloch          float64
Glossbrenner       float64
IPFPoints          float64
Tested              object
Country             object
Federation          object
Date                object
MeetCountry         object
MeetState           object
MeetName            object
Squat / BW         float64
Bench / BW         float64
Deadlift / BW      float64
Total / BW         float64
dtype: object
```

```
sns.lmplot(y='Best3DeadliftKg', x='BodyweightKg', hue='Sex',
           data=df,
           fit_reg=False, scatter_kws={'alpha':0.7})
```

```
<seaborn.axisgrid.FacetGrid at 0x7f7af90015a0>
```



✓ 14s completed at 4:16 PM