In [7]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

In [8]:

```
athletes = pd.read_csv("D:\\Huge_datasets\\bgadoci-crossfit-data\\athletes.csv", low_r
leaderboards = pd.read_csv("D:\\Huge_datasets\\bgadoci-crossfit-data\\leaderboard.15.c
```

In [9]:

```
print(athletes.shape)
print(athletes.columns)
exercises_list = ['fran', 'helen', 'grace', 'filthy50', 'fgonebad',
        'run400', 'run5k', 'candj', 'snatch', 'deadlift', 'backsq', 'pullups']
nan_athletes = athletes[exercises_list].isna().sum(axis = 1)
athletes.head()
```

```
(423006, 28)
Index(['athlete_id', 'name', 'region', 'team', 'affiliate', 'gender', 'age',
       'height', 'weight', 'fran', 'helen', 'grace', 'filthy50', 'fgonebad',
       'run400', 'run5k', 'candj', 'snatch', 'deadlift', 'backsq', 'pullups',
       'eat', 'train', 'background', 'experience', 'schedule', 'howlong',
       'retrieved_datetime'],
      dtype='object')
```

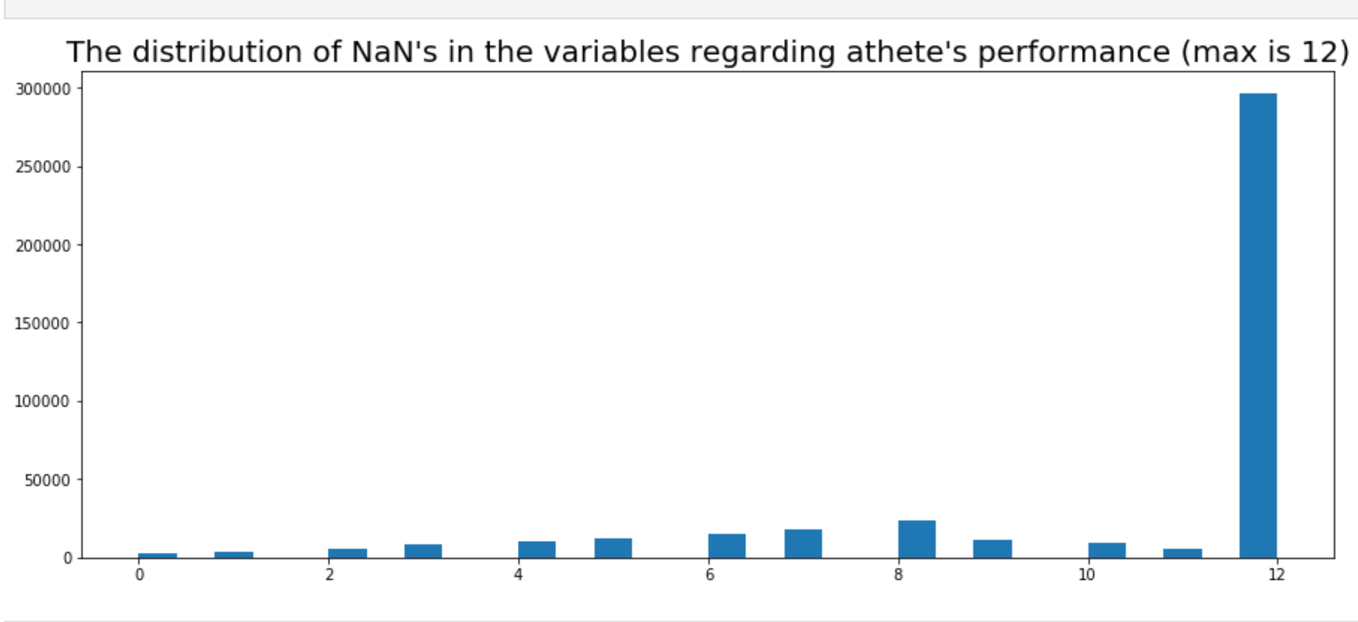
Out[9]:

	athlete_id	name	region	team	affiliate	gender	age	height	weight	fran	...	deadlift	backsq
0	2554.0	Pj Ablang	South West	Double Edge	Double Edge CrossFit	Male	24.0	70.0	166.0	NaN	...	400.0	305.0
1	3517.0	Derek Abdella	NaN	NaN	NaN	Male	42.0	70.0	190.0	NaN	...	NaN	NaN
2	4691.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
3	5164.0	Abo Brandon	Southern California	LAX CrossFit	LAX CrossFit	Male	40.0	67.0	NaN	211.0	...	375.0	325.0
4	5286.0	Bryce Abbey	NaN	NaN	NaN	Male	32.0	65.0	149.0	206.0	...	NaN	325.0

5 rows × 28 columns

In [10]:

```
plt.figure(figsize = (15,6))
plt.hist(nan_athletes, bins = 30)
plt.title("The distribution of NaN's in the variables regarding athete's performance
plt.show()
```



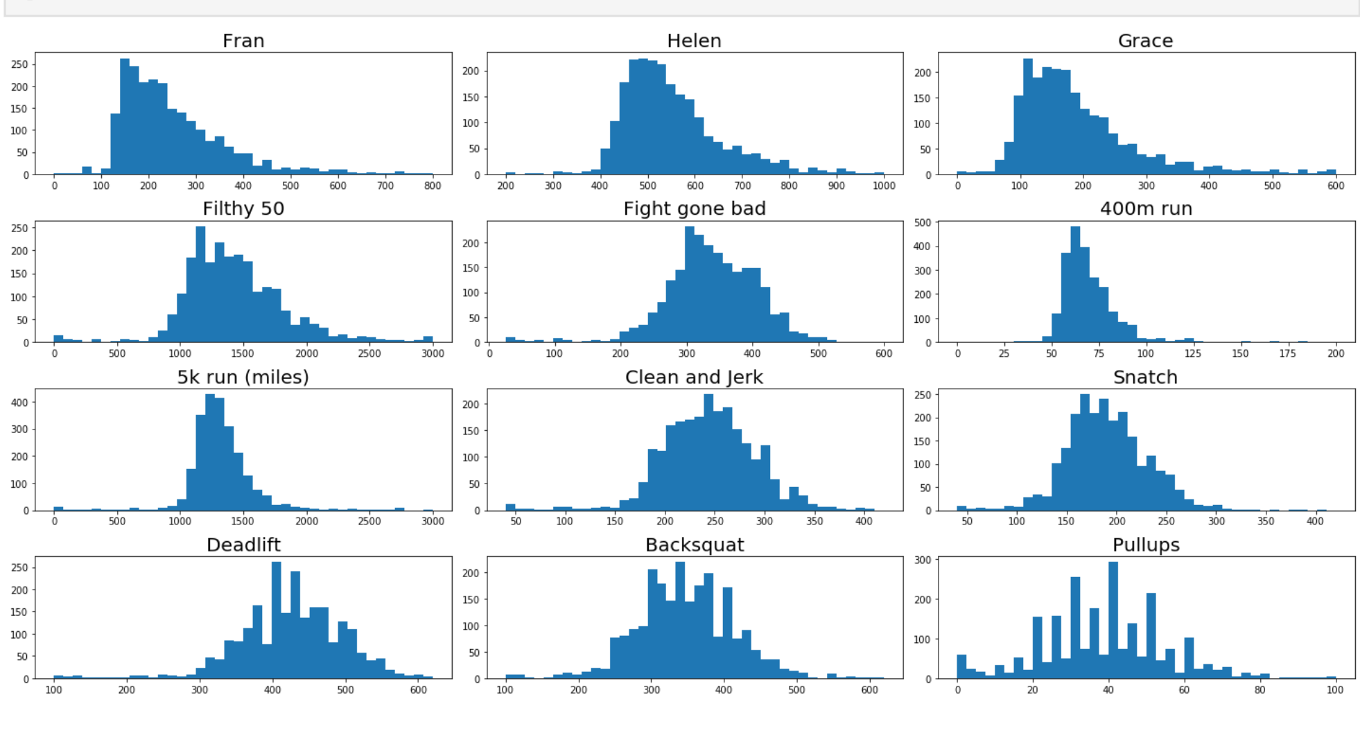
In [11]:

```
athletes2 = athletes.iloc[nan_athletes.values < 1,]
males = athletes2.iloc[(athletes2["gender"] == "Male").values]
females = athletes2.iloc[(athletes2["gender"] == "Female").values,]
print(males.shape)
```

```
(2388, 28)
```

In [12]:

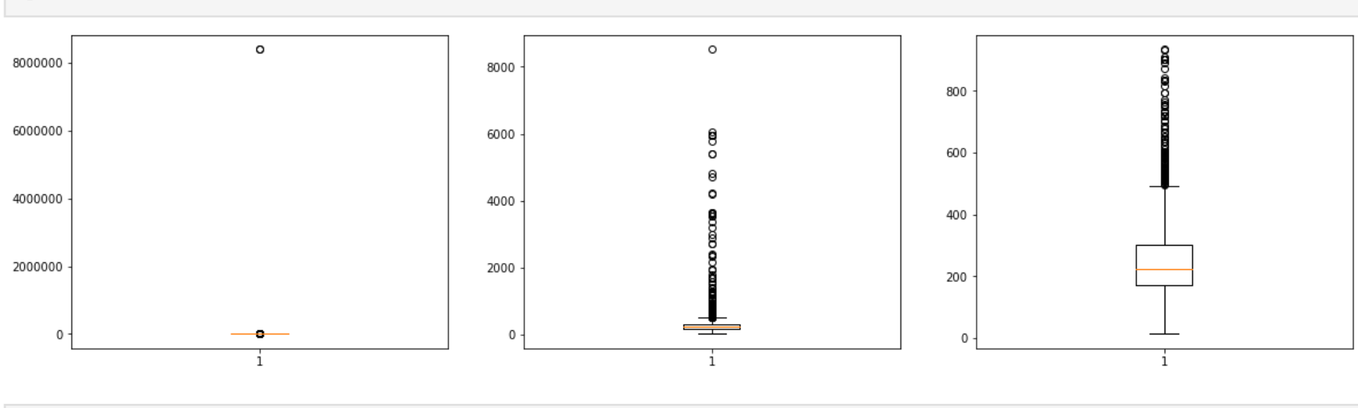
```
fig = plt.figure(figsize = (20,10))
plt.subplot(431)
plt.title("Fran", size = 20)
plt.hist(males["fran"], range=[0, 800], bins = 40)
plt.subplot(432)
plt.title("Helen", size = 20)
plt.hist(males["helen"], range=[200, 1000], bins = 40)
plt.subplot(433)
plt.title("Grace", size = 20)
plt.hist(males["grace"], range=[0, 600], bins = 40)
plt.subplot(434)
plt.title("Filthy 50", size = 20)
plt.hist(males["filthy50"], range=[0, 3000], bins = 40)
plt.subplot(435)
plt.title("Fight gone bad", size = 20)
plt.hist(males["fgonebad"], range=[25, 600], bins = 40)
plt.subplot(436)
plt.title("400m run", size = 20)
plt.hist(males["run400"], range=[0, 200], bins = 40)
plt.subplot(437)
plt.title("5k run (miles)", size = 20)
plt.hist(males["run5k"], range=[0, 3000], bins = 40)
plt.subplot(438)
plt.title("Clean and Jerk", size = 20)
plt.hist(males["candj"], range=[40, 420], bins = 40)
plt.subplot(439)
plt.title("Snatch", size = 20)
plt.hist(males["snatch"], range=[40, 420], bins = 40)
plt.subplot(4,3,10)
plt.title("Deadlift", size = 20)
plt.hist(males["deadlift"], range=[100, 620], bins = 40)
plt.subplot(4,3,11)
plt.title("Backsquat", size = 20)
plt.hist(males["backsq"], range=[100, 620], bins = 40)
plt.subplot(4,3,12)
plt.title("Pullups", size = 20)
plt.hist(males["pullups"], range=[0, 100], bins = 40)
fig.tight_layout()
plt.show()
```



Outliers

In [13]:

```
plt.figure(figsize = (20,5))
plt.subplot(131)
plt.boxplot(males["fran"])
plt.subplot(132)
plt.boxplot(males["fran"][males["fran"] < 10000] )
plt.subplot(133)
plt.boxplot(males["fran"][males["fran"] < 1000] )
plt.show()
```



In [14]:

```
males["fran"].quantile(0.95)
```

Out[14]:

```
587.6000000000004
```

Leaderboards

Leaderboards: dataset from 2015 open part of Crossfit Games. We have 5 workuts there. https://games.crossfit.com/workouts/open/2015#workoutsTab1

In [15]:

```
leaderboards.head()
```

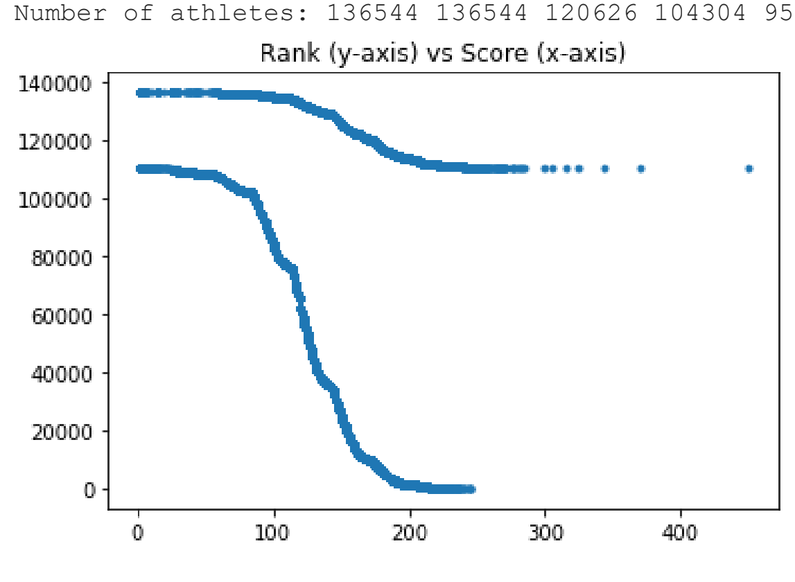
Out[15]:

	year	division	stage	athlete_id	rank	score	retrieved_datetime	scaled
0	15	1	5.0	1690.0	154.0	366.0	2015-03-31 21:44:44	0
1	15	1	5.0	1998.0	5950.0	497.0	2015-03-31 21:44:44	0
2	15	1	5.0	2206.0	768.0	404.0	2015-03-31 21:44:44	0
3	15	1	5.0	2559.0	294.0	379.0	2015-03-31 21:44:44	0
4	15	1	5.0	2811.0	1946.0	437.0	2015-03-31 21:44:44	0

In [16]:

```
print("Workout tags:", leaderboards["stage"].unique())
#division = 1 --> males

event1 = leaderboards[(leaderboards["stage"] == 1).values * (leaderboards["division"]
event11 = leaderboards[(leaderboards["stage"] == 1.1).values * (leaderboards["divisior
event2 = leaderboards[(leaderboards["stage"] == 2).values * (leaderboards["division"]
event3 = leaderboards[(leaderboards["stage"] == 3).values * (leaderboards["division"]
event4 = leaderboards[(leaderboards["stage"] == 4).values * (leaderboards["division"]
event5 = leaderboards[(leaderboards["stage"] == 5).values * (leaderboards["division"]

print("Number of athletes:",event1.shape[0], event11.shape[0],event2.shape[0], event3
plt.plot(event1["score"].values, event1["rank"].values, ".")
plt.title("Rank (y-axis) vs Score (x-axis)")
plt.show()
```

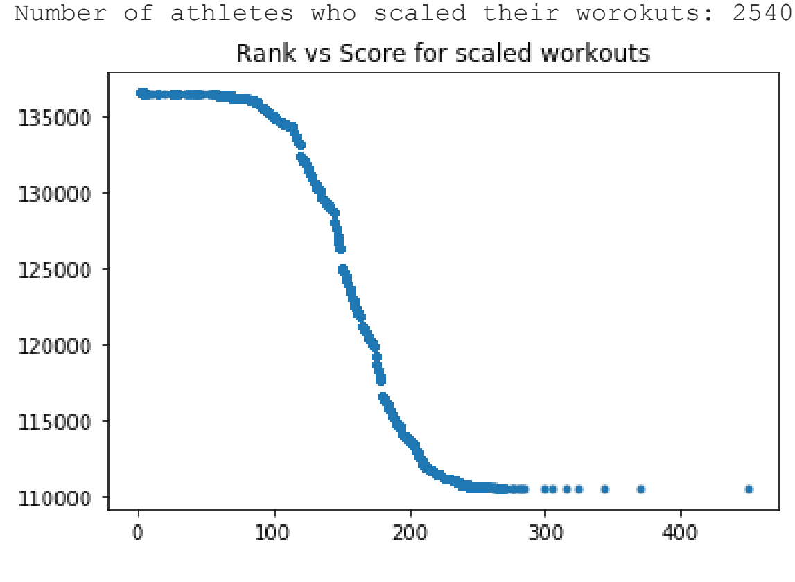
```
Workout tags: [5.  1.  1.1 2.  3.  4. ]
Number of athletes: 136544 136544 120626 104304 95224 84229
```



In [17]:

```
event1rx = event1.iloc[(event1["scaled"] == 0).values,]
print("Number of athletes who didn't scale their worokuts:", event1rx.shape[0])
event1sc = event1.iloc[(event1["scaled"] == 1).values,]
print("Number of athletes who scaled their worokuts:", event1sc.shape[0])

plt.plot(event1sc["score"], event1sc["rank"], ".")
plt.title("Rank vs Score for scaled workouts")
plt.show()
```

```
Number of athletes who didn't scale their worokuts: 111136
Number of athletes who scaled their worokuts: 25408
```



In [18]:

```
n = 1000
best_event1 = set(event1[event1["rank"] <= n]["athlete_id"])
best_event11 = set(event11[event11["rank"] <= n]["athlete_id"])
best_event2 = set(event2[event2["rank"] <= n]["athlete_id"])
best_event3 = set(event3[event3["rank"] <= n]["athlete_id"])
best_event4 = set(event4[event4["rank"] <= n]["athlete_id"])
best_event5 = set(event5[event5["rank"] <= n]["athlete_id"])
print(len(best_event1), len(best_event11), len(best_event2), len(best_event3), len(bes
top_males = best_event1.intersection(best_event11).intersection(best_event2).intersect
len(top_males)
```

```
1125 1003 1043 1004 1000 1037
```

Out[18]:

```
89
```

In [19]:

```
top_df = pd.DataFrame(index = range(len(top_males)), columns = males.columns)

for i, id in enumerate(top_males):
    #print(i)
    top_df.iloc[i, ] = athletes[athletes["athlete_id"] == id].values

top_df["event1_rank"] = np.nan
top_df["event11_rank"] = np.nan
top_df["event2_rank"] = np.nan
top_df["event3_rank"] = np.nan
top_df["event4_rank"] = np.nan
top_df["event5_rank"] = np.nan

for i, id in enumerate(top_males):
    scores = leaderboards[leaderboards["athlete_id"] == id]["rank"].values
    for j in range(6):
        top_df.iloc[i, j-6] = scores[j]

top_df.head()
```

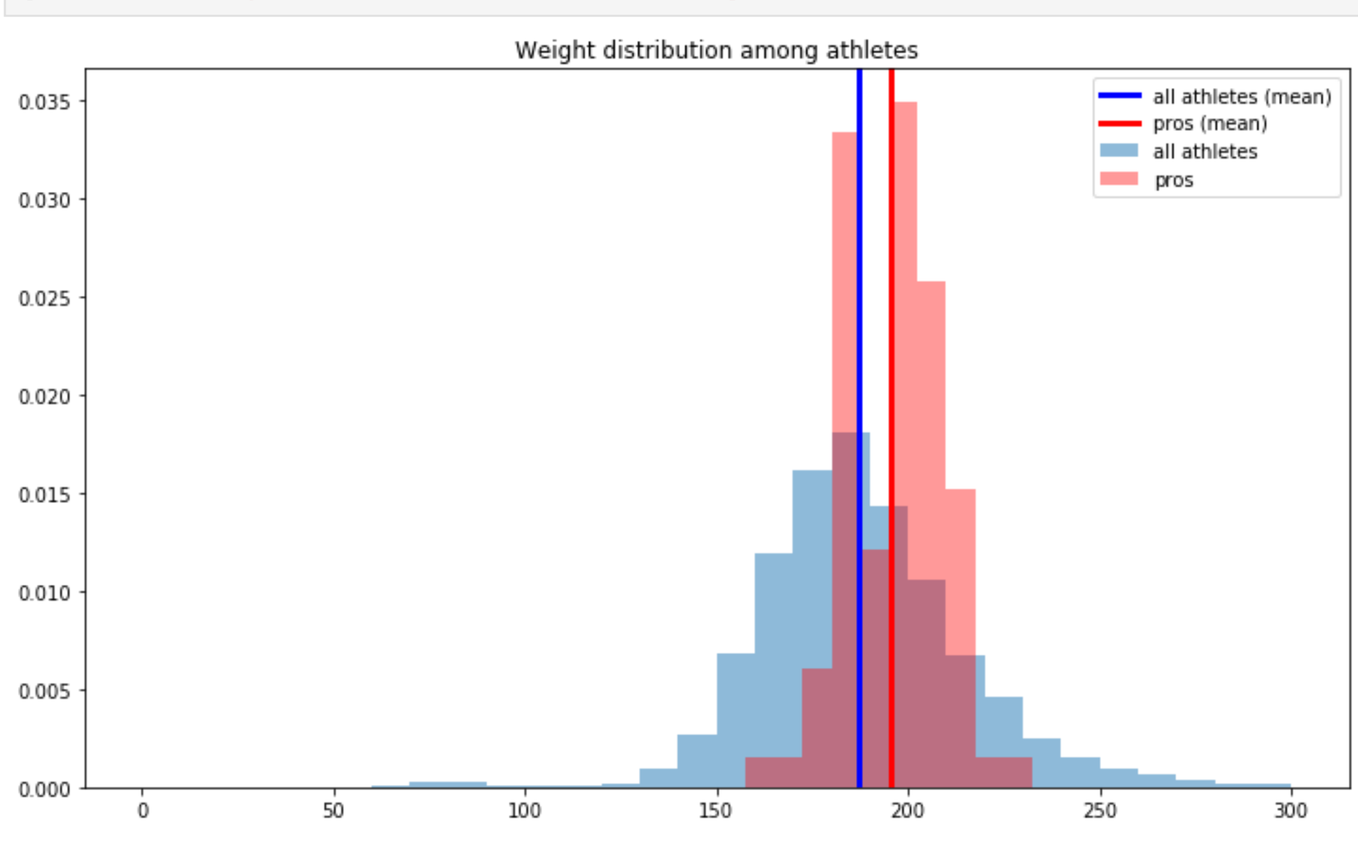
Out[19]:

	athlete_id	name	region	team	affiliate	gender	age	height	weight	fran	...	experience
0	102914	Dylan Kelly	North Central	CrossFit COMO	CrossFit COMO	Male	22	70	215	154	...	I began CrossFit by trying it alone (without a...
1	153604	Mathew Fraser	North East	Champlain Valley CrossFit	Champlain Valley CrossFit	Male	25	66	NaN	127	...	NaN
2	120333	Jason Smith	Africa	CrossFit Kyalami	CrossFit Kyalami	Male	30	73	195	149	...	I began CrossFit with a coach (e.g. at an affi...
3	15378	Casey Haines	Mid Atlantic	CrossFit Explode	CrossFit Explode	Male	23	73	210	NaN	...	NaN
4	30740	Ben Stoneberg	North West	CrossFit Intensify	CrossFit Intensify	Male	27	67	180	130	...	I began CrossFit with a coach (e.g. at an affi...

5 rows × 34 columns

In [20]:

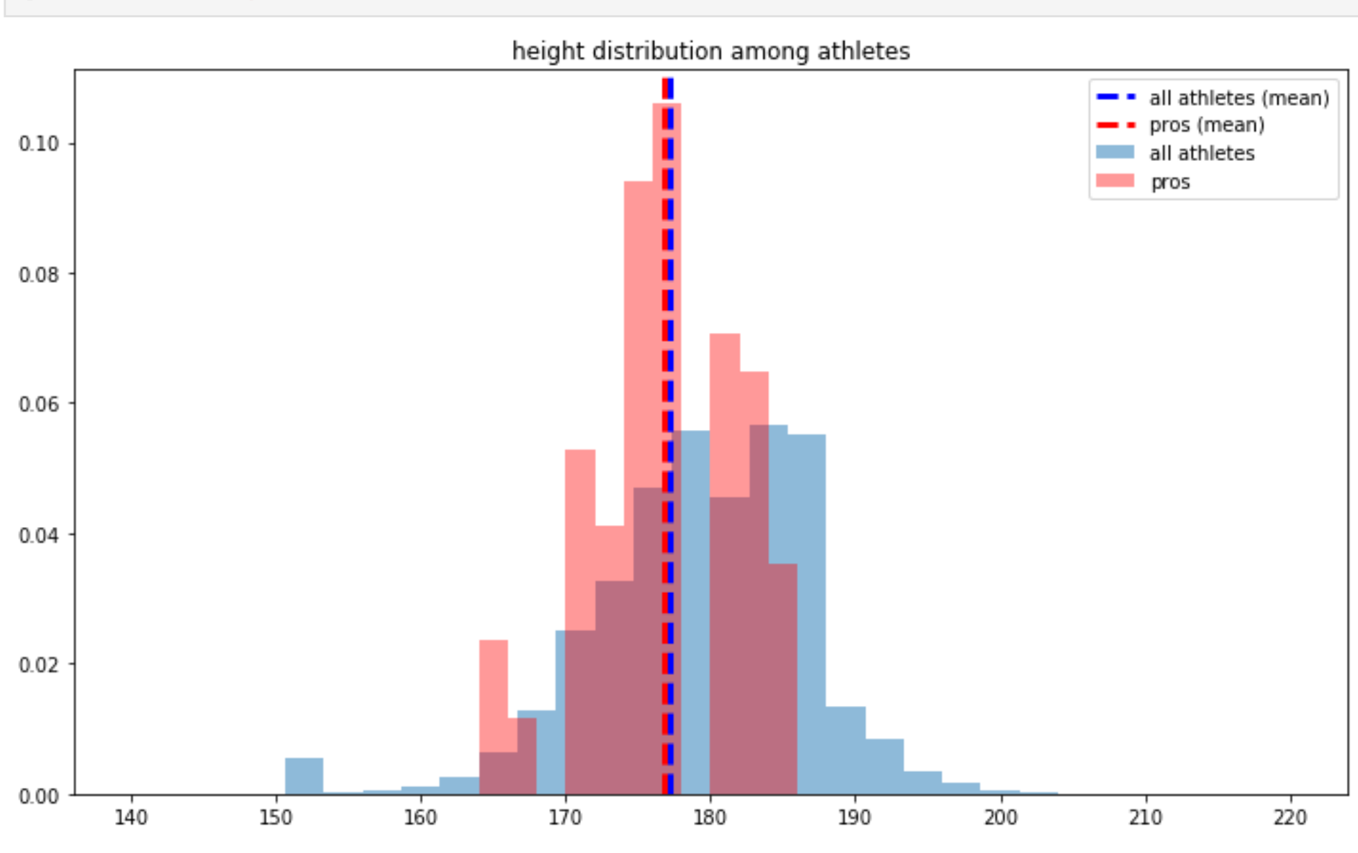
```
all_weight = athletes[athletes["gender"] == "Male"]["weight"].dropna().values
top_weight = top_df["weight"].dropna().values
plt.figure(figsize = (12,7))
plt.hist(all_weight, range = [0, 300], alpha = 0.5, density = True, bins = 30)
plt.hist(top_weight, range = [0, 300], color = "red", density = True, bins = 40, alpha
plt.axvline(x = all_weight[all_weight < 400].mean(), color = "blue", lw = 3)
plt.axvline(x = top_weight.mean(), color = "red", lw = 3)
plt.legend(labels = ["all athletes (mean)", "pros (mean)", "all athletes", "pros"])
plt.title("Weight distribution among athletes")

plt.show()
print("Pros sd =", top_weight.var()**0.5, "\nAll athletes sd =", all_weight[all_weight
print("The weight of athletes was cut at 400 pounds (without this cut the sd = 64, due
```



```
Pros sd = 12.099914409573657
All athletes sd = 27.84588284463546
The weight of athletes was cut at 400 pounds (without this cut the sd = 64, due to out
liers)
```

In [21]:

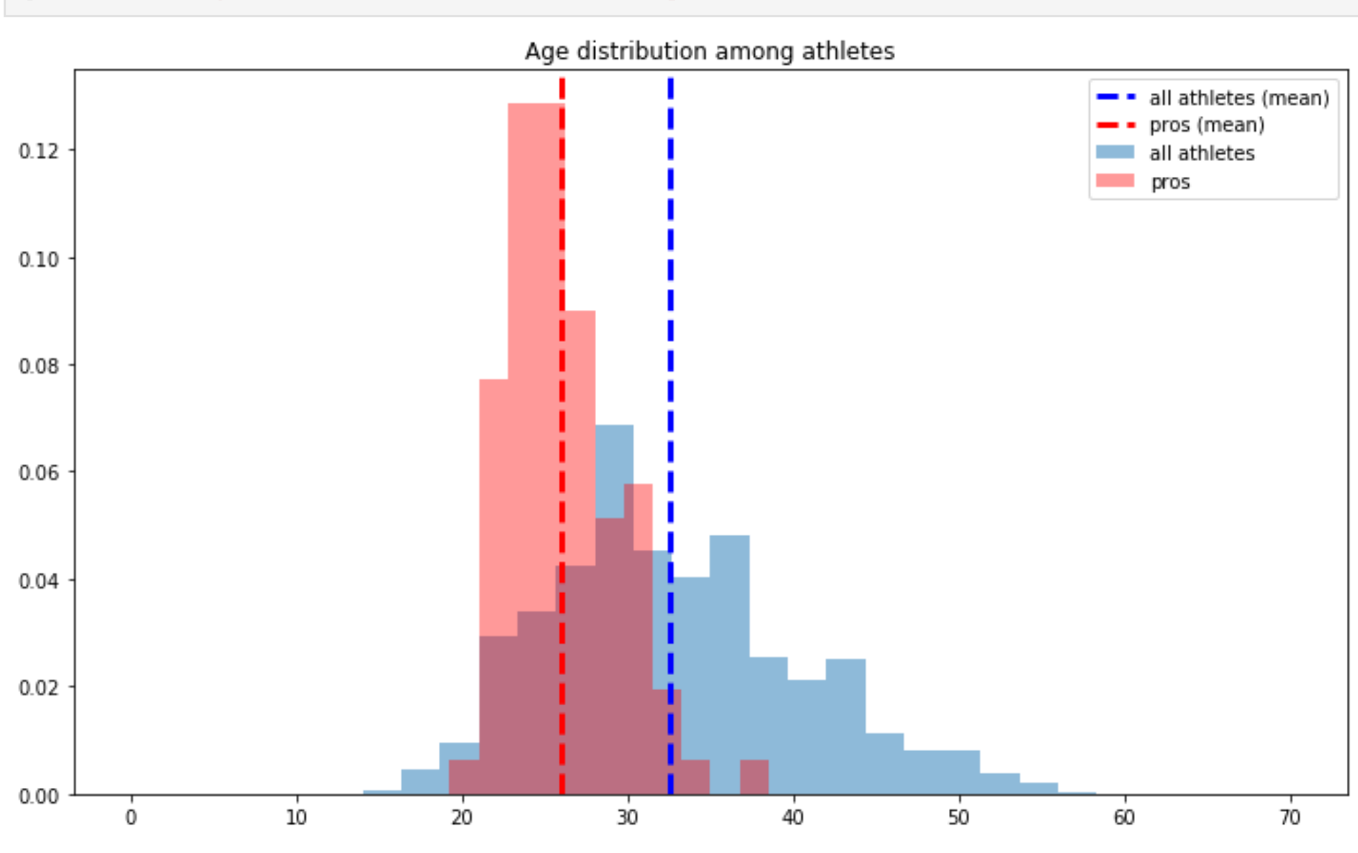
```
all_height = athletes[athletes["gender"] == "Male"]["height"].dropna().values*2.54
top_height = top_df["height"].dropna().values*2.54
plt.figure(figsize = (12,7))
plt.hist(all_height, range = [140, 220], alpha = 0.5, density = True, bins = 30)
plt.hist(top_height, range = [140, 220], color = "red", density = True, bins = 40, alp
plt.axvline(x = all_height[all_height < 250].mean(), color = "blue", lw = 3, ls = "--'
plt.axvline(x = top_height.mean(), color = "red", lw = 3, ls = "--")
plt.legend(labels = ["all athletes (mean)", "pros (mean)", "all athletes", "pros"])
plt.title("height distribution among athletes")

plt.show()
print("Pros sd =", top_height.var()**0.5, "\nAll atlhetes sd =", all_height[all_height
print("The height of athletes was cut at 250 cm (due to outliers)")
```



```
Pros sd = 5.184911161328133
All atlhetes sd = 18.093539326005942
The height of athletes was cut at 250 cm (due to outliers)
```

In [22]:

```
all_age = athletes[athletes["gender"] == "Male"]["age"].dropna().values
top_age = top_df["age"].dropna().values
plt.figure(figsize = (12,7))
plt.hist(all_age, range = [0, 70], alpha = 0.5, density = True, bins = 30)
plt.hist(top_age, range = [0, 70], color = "red", density = True, bins = 40, alpha = (
plt.axvline(x = all_age[all_age < 250].mean(), color = "blue", lw = 3, ls = "--")
plt.axvline(x = top_age.mean(), color = "red", lw = 3, ls = "--")
plt.legend(labels = ["all athletes (mean)", "pros (mean)", "all athletes", "pros"])
plt.title("Age distribution among athletes")

plt.show()
print("Pros sd =", top_age.var()**0.5, "\nAll athletes sd =", all_age[all_age < 70].va
print("The age of athletes was cut at 70 years")
```

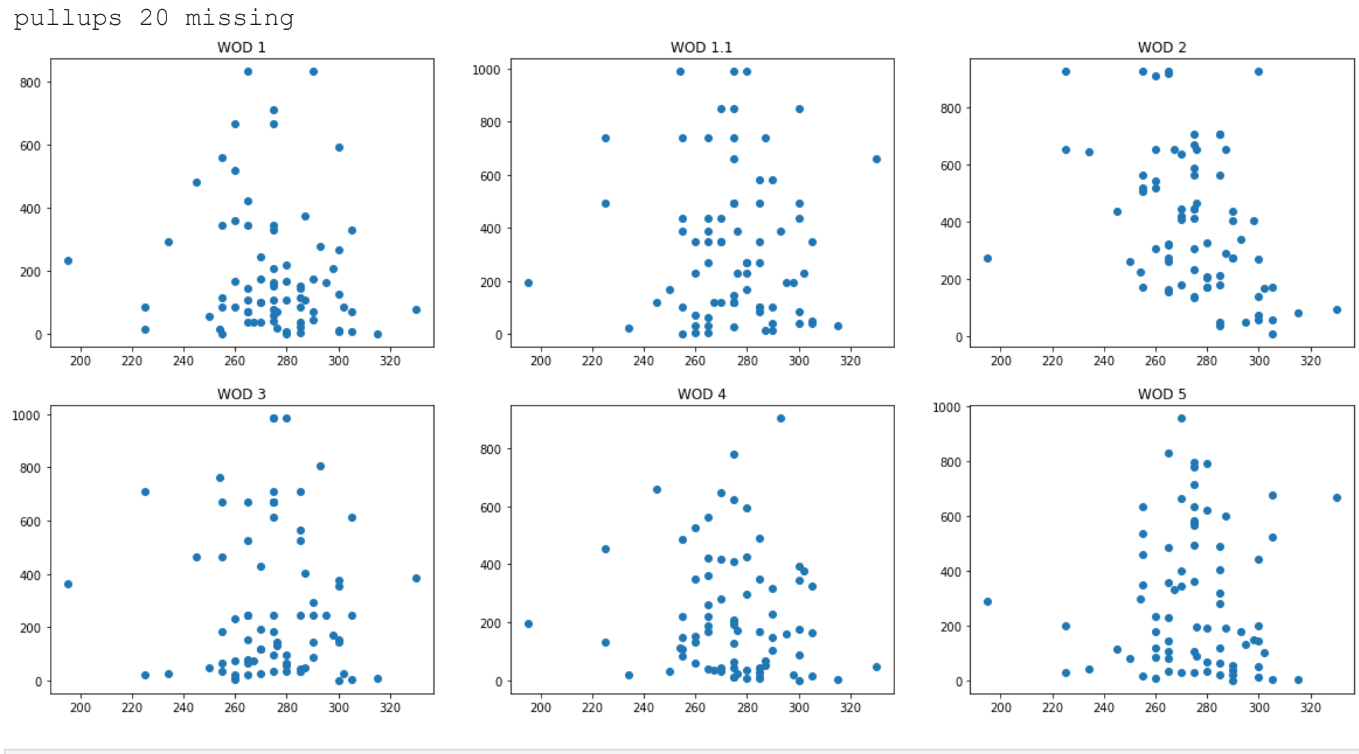


```
Pros sd = 3.263614622855205
All athletes sd = 7.665577205014458
The age of athletes was cut at 70 years
```

In [23]:

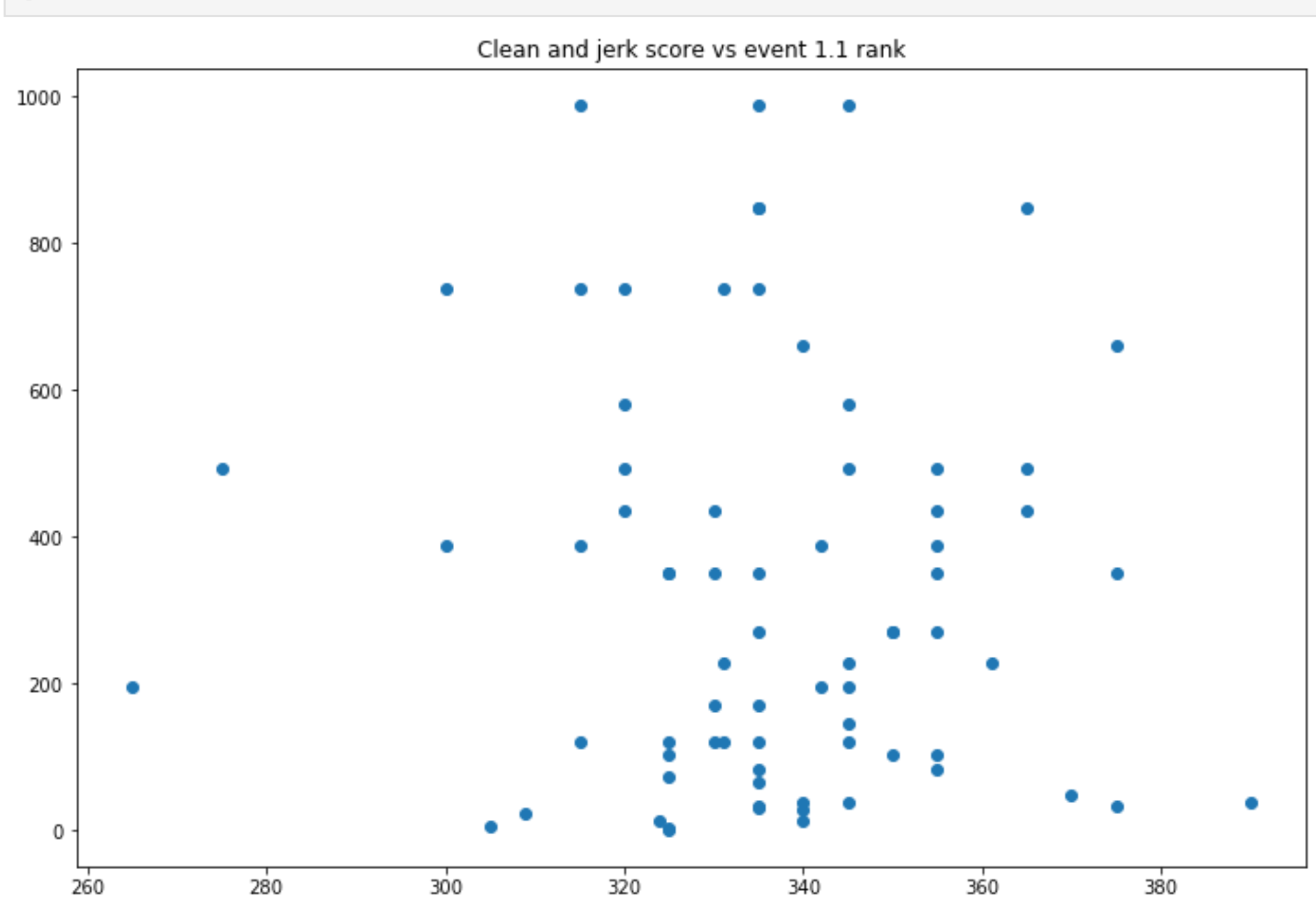
```
for exercise in exercises_list:
    print(exercise ,top_df[exercise].isna().sum(), "missing")

plt.figure(figsize = (20, 10))
df = top_df[top_df["snatch"] > 0]
plt.subplot(231)
plt.title("WOD 1")
plt.plot(df["snatch"], df["event1_rank"], "o")
plt.subplot(232)
plt.title("WOD 1.1")
plt.plot(df["snatch"], df["event11_rank"], "o")
plt.subplot(233)
plt.title("WOD 2")
plt.plot(df["snatch"], df["event2_rank"], "o")
plt.subplot(234)
plt.title("WOD 3")
plt.plot(df["snatch"], df["event3_rank"], "o")
plt.subplot(235)
plt.title("WOD 4")
plt.plot(df["snatch"], df["event4_rank"], "o")
plt.subplot(236)
plt.title("WOD 5")
plt.plot(df["snatch"], df["event5_rank"], "o")

plt.show()
```

```
fran 18 missing
helen 49 missing
grace 28 missing
filthy50 53 missing
fgonebad 26 missing
run400 48 missing
run5k 45 missing
candj 11 missing
snatch 10 missing
deadlift 11 missing
backsq 12 missing
pullups 20 missing
```



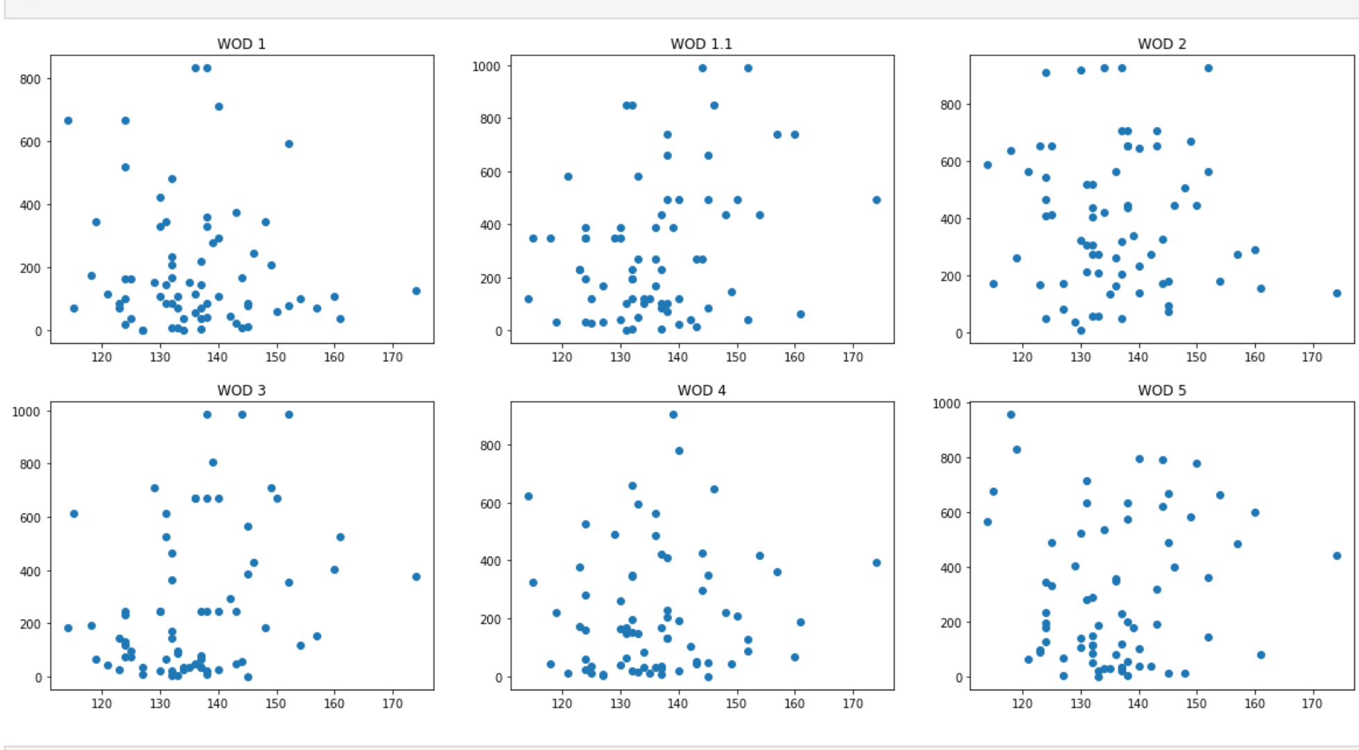
In [29]:

```
plt.figure(figsize = (12, 8))
df = top_df[top_df["candj"] > 0]

plt.plot(df["candj"], df["event11_rank"], "o")
plt.title("Clean and jerk score vs event 1.1 rank")#event 1.1 = max clen and jerk

plt.show()
```



In [24]:

```
plt.figure(figsize = (20, 10))
df = top_df[top_df["fran"] > 0]
plt.subplot(231)
plt.title("WOD 1")
plt.plot(df["fran"], df["event1_rank"], "o")
plt.subplot(232)
plt.title("WOD 1.1")
plt.plot(df["fran"], df["event11_rank"], "o")
plt.subplot(233)
plt.title("WOD 2")
plt.plot(df["fran"], df["event2_rank"], "o")
plt.subplot(234)
plt.title("WOD 3")
plt.plot(df["fran"], df["event3_rank"], "o")
plt.subplot(235)
plt.title("WOD 4")
plt.plot(df["fran"], df["event4_rank"], "o")
plt.subplot(236)
plt.title("WOD 5")
plt.plot(df["fran"], df["event5_rank"], "o")

plt.show()
```



In [25]:

```
plt.figure(figsize = (20, 10))
df = top_df[top_df["deadlift"] > 0]
plt.subplot(231)
plt.title("WOD 1")
plt.plot(df["deadlift"], df["event1_rank"], "o")
plt.subplot(232)
plt.title("WOD 1.1")
plt.plot(df["deadlift"], df["event11_rank"], "o")
plt.subplot(233)
plt.title("WOD 2")
plt.plot(df["deadlift"], df["event2_rank"], "o")
plt.subplot(234)
plt.title("WOD 3")
plt.plot(df["deadlift"], df["event3_rank"], "o")
plt.subplot(235)
plt.title("WOD 4")
plt.plot(df["deadlift"], df["event4_rank"], "o")
plt.subplot(236)
plt.title("WOD 5")
plt.plot(df["deadlift"], df["event5_rank"], "o")

plt.show()
```