Project 1 Report

Team members: Manish Meshram

Introduction

This project focuses on analysing the baseball data over the last 150 years. It aims to give insights on how runs scored by player can affect salary of that player. It also analyzes if weight of the player can affect the possibility of scoring more runs over the player's career. Lastly it also comments on the popularity of the sport based on new debutants each year.

Dataset

Baseball is a bat-and-ball game played between two opposing teams who take turns batting and fielding. The baseball dataset provided by SeanLahman.com gives us the data for last 150 years including batting, pitching, fielding, teams, salaries etc information. For this project, I have mostly focused on players' personal information, batting and players' salaries data. Most of the data was well formatted and clean. I needed to do a little munging to get the available data in the format that analyses needed.

Analysis technique

Most of the analyses in this project revolve around the batting data of players. To get started on this I retrieved the highest run scorers of all time. I have plotted the top 10 run scorers of all time with the help of bar chart. For getting the highest run scorers I have calculated total runs scored by each player throughout his career and the sorted the data with highest run scorers at the top.

Digging down further, I wanted to know if runs scored by the player in his career is affected by weight of the player since we know that an overweight person is less likely to run fast which is a precondition to steal runs on the field. To analyse this, I have plotted a scatterplot of runs scored by the player in his career vs weight of the player. To improve more on the plotted graph, I tried standardizing the variables and plotting the values again. Based on the results of aforementioned graphs, I was really curious to dig out more on this, I wanted to know if the pattern that we are getting can be generalized even if we select chunk of data. For doing this I selected the top 500 and bottom 500 players (from the total pool of approx. 18000 players) based on the runs scored by them in their career and plotted the scatterplot rendering Runs scored vs weight relation.

In addition to this I wanted to know if runs scored by the player can affect the salary of that player in any way. Comparing the mean of two groups' salaries seemed like a good idea to make a viable assertion. For this analysis, I compared the salary distributions of top 500 and bottom 500 players based on the runs scored by them in their career. From the salaries data I have observed that each player gets an increased salary every year he continues to play. So for this analysis I have considered the mean salary of a player throughout his career. For showing the results of this analysis box-plot is being used since it exactly gives us what I wanted to compare; the mean and the overall salary distributions of the two groups.

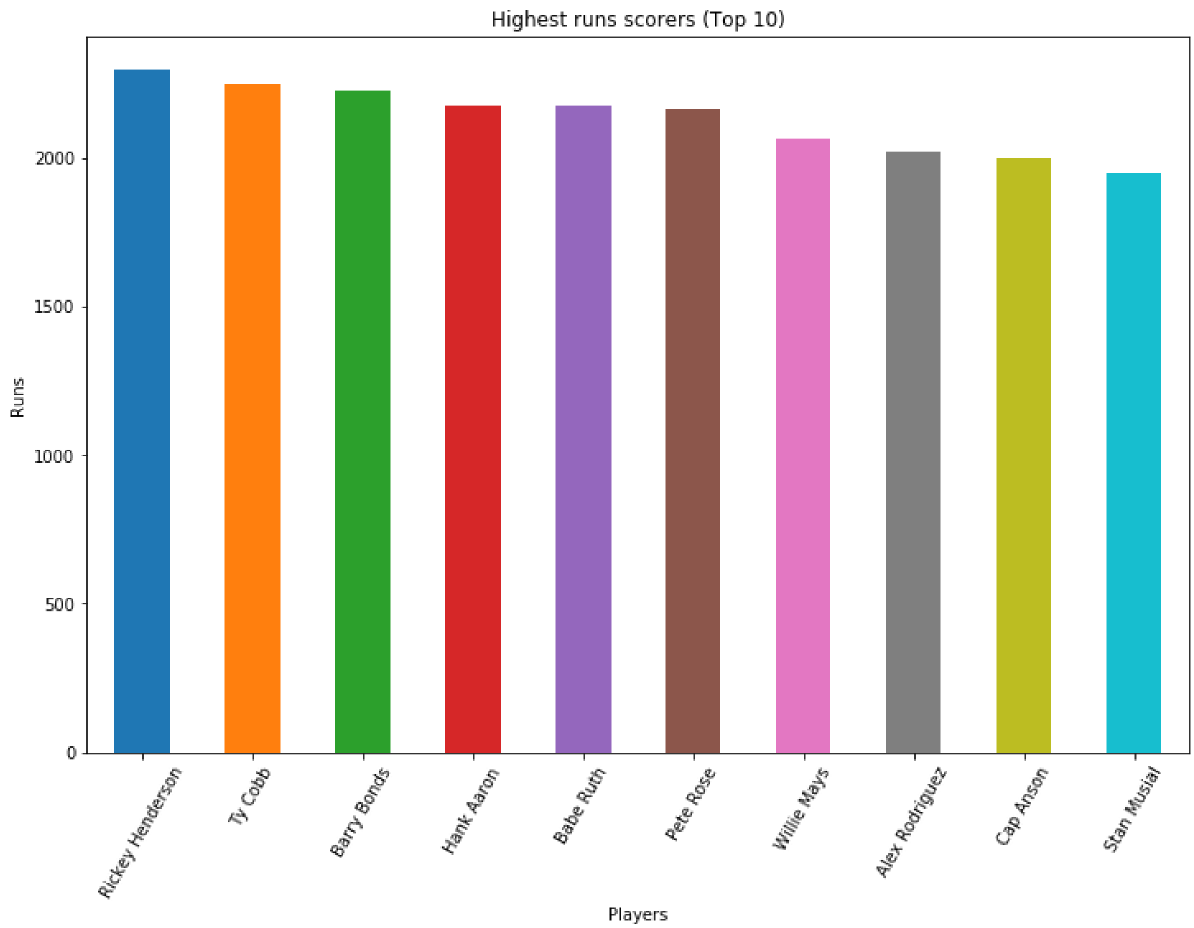
Finally I wanted to know the overall growth and popularity of baseball sport over the years. I wanted to know if there is any trend in the players onboarded every decade which in turn makes a remark about popularity of the sport and tells us if more people are interested in choosing baseball as a career as compared to previous years. It also aims to answer a question; whether it is a good sport to invest as a team sponsor. This observation is shown by using the bar chart.

Results

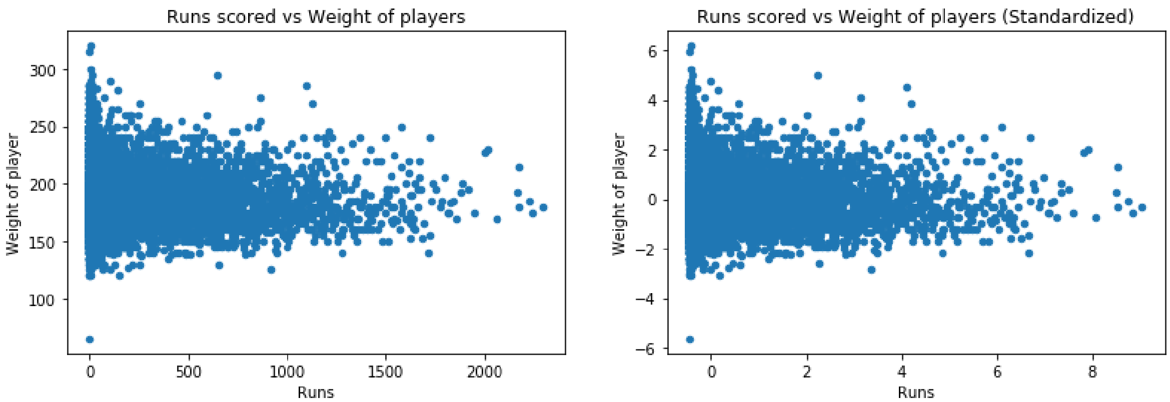
This project uses players batting data in most of the analyses.

Top run scorers of all time

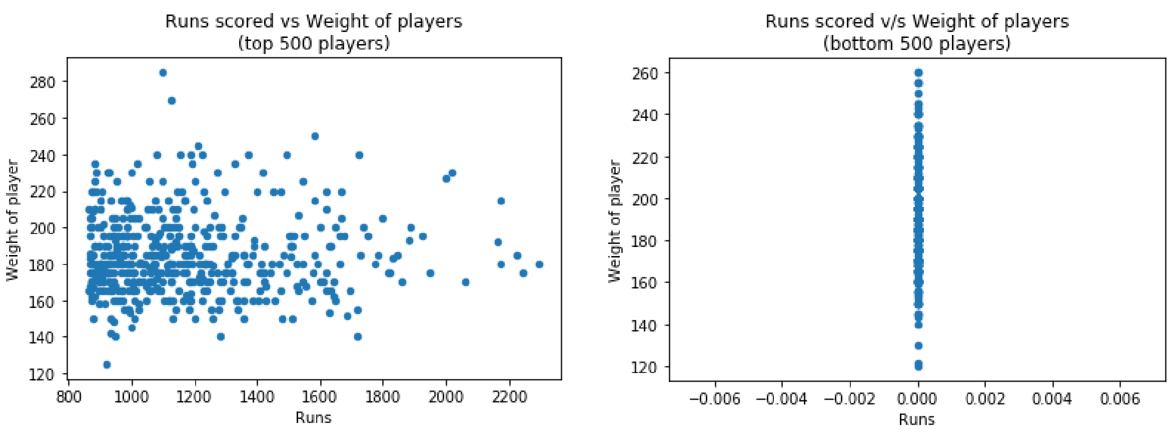
The graph below shows us the top players who scored the most runs in their career.



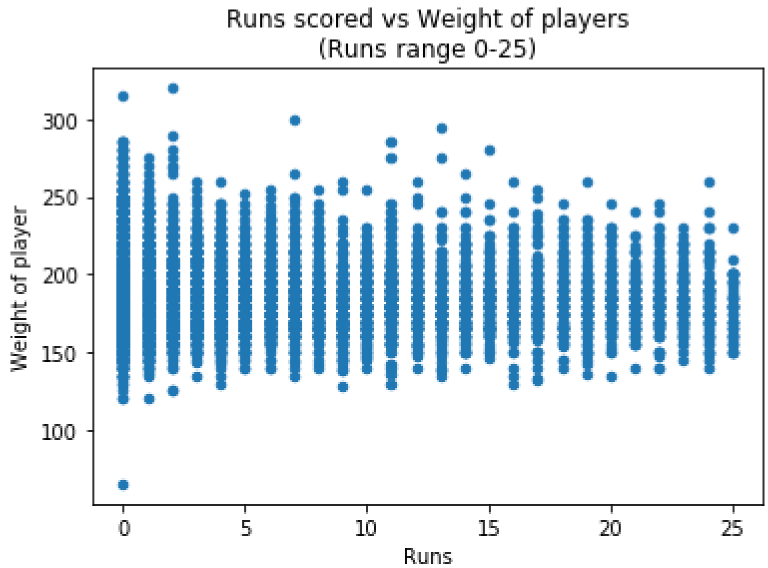
Relation between runs scored and weight of players



When I plotted runs scored by players against weight of the players, I have seen a relation between these two attributes. From the above graphs it can be seen that most of the highest run hitters have weights close to 200 lbs. To dig deeper in this I have used the top 500 and bottom 500 players (based on the runs they scored in their respective careers) out of around 18000 players and did the same analysis again. These analysis can be depicted by below graphs.



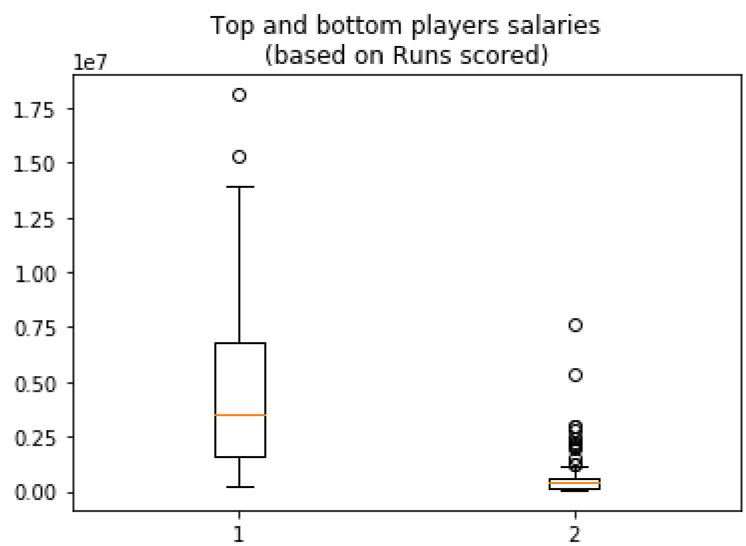
From the above graphs we can see that, the top players tend to follow the weak trend as depicted by previous graphs, most of the high hitters are around 180 lbs. However, this is not true for bottom 500 players. But we cannot expect it to be true since all the bottom players scored exactly 0 runs, this may be because the bottom players are actually more active in fielding or pitching side of the game. The other way to see if there is any trend is by plotting the players who scored runs between a range. I did it for the range 0-25 runs and the result is as follows.



The above graph does show the trend we discussed before but it actually considers 11950 players out of approx. 18000 players in the dataset. Overall, I can say that we cannot strongly generalize the observed trend.

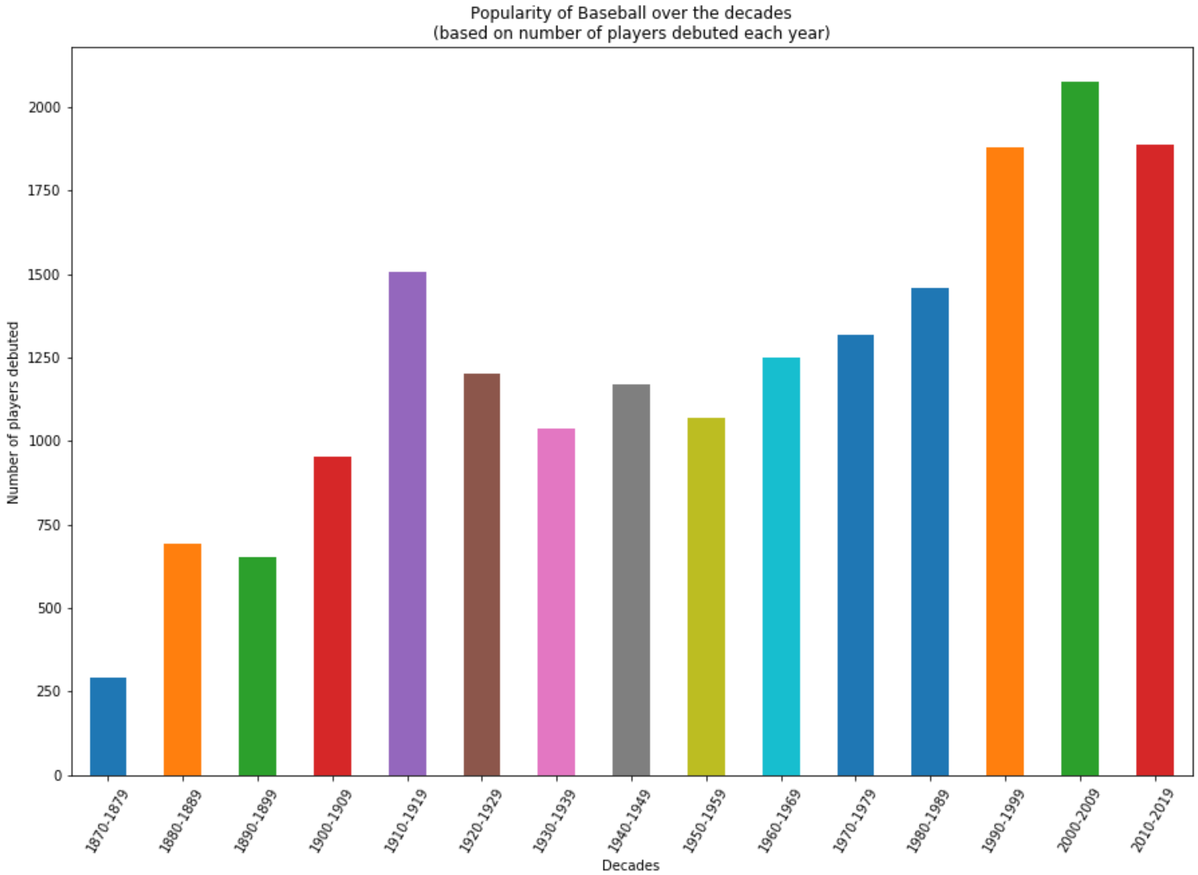
Salaries and runs scored by the players

The distributions of average salaries of top 500 and bottom 500 players(based on the runs they scored in their careers) are explained by the following graph:



From this graph we can say that the players who has hit more runs are likely to get more salaries as compared to players who didn't.

Popularity of baseball over the decades (based on the players debut each year)



From the above graph we can see that more players are choosing baseball as a career option and are getting successful in that. Popularity of baseball is definitely on rise and it seems to be a good sport to invest if individuals and companies are planning to get involved in the sport.

Project 1 Code

Preparing the data

In [135]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

In [136]:
```
people = pd.read_csv('baseballdatabank-master/core/People.csv', usecols=[
display(people.head())
display('Length of people:', len(people))

batting = pd.read_csv('baseballdatabank-master/core/Batting.csv', usecols=
runs = batting.groupby('playerID')['R'].agg(['sum']).reset_index()
display(runs.head())
display('Length of sum runs:', len(runs))

runs_desc = runs.sort_values(['sum'], ascending=[False]).reset_index(drop=
display(runs_desc.head(10))

merged_df = runs_desc.merge(people, how = 'inner', on = ['playerID'])

# Adding full name
merged_df['name']= merged_df['nameFirst'] + ' ' +  merged_df['nameLast']
display(merged_df.head())

# Dropping NaN values
merged_df = merged_df.dropna()

# Adding debut year
merged_df['debutYear']= merged_df['debut'].str.split('-').str[0]
merged_df['debutYear'] = merged_df['debutYear'].astype('int')
display(merged_df.head())
```

	playerID	nameFirst	nameLast	weight	debut
0	aardsda01	David	Aardsma	215.0	2004-04-06
1	aaronha01	Hank	Aaron	180.0	1954-04-13
2	aaronto01	Tommie	Aaron	190.0	1962-04-10
3	aasedo01	Don	Aase	190.0	1977-07-26
4	abadan01	Andy	Abad	184.0	2001-09-10

'Length of people:'

19370

	playerID	sum
0	aardsda01	0
1	aaronha01	2174
2	aaronto01	102
3	aasedo01	0
4	abadan01	1

'Length of sum runs:'

19182

	playerID	sum
0	henderi01	2295
1	cobbty01	2246
2	bondsba01	2227
3	aaronha01	2174
4	ruthba01	2174
5	rosepe01	2165
6	mayswi01	2062
7	rodrial01	2021
8	ansonca01	1999
9	musiast01	1949

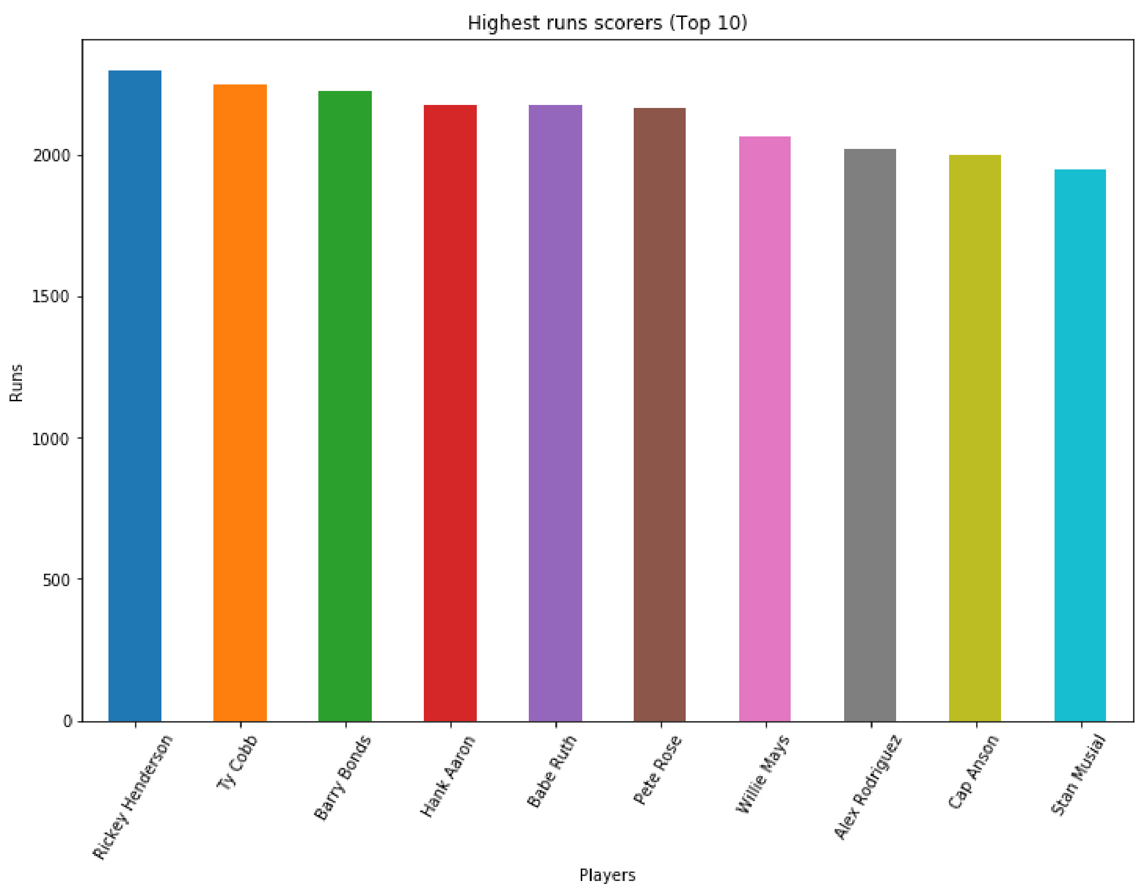
	playerID	sum	nameFirst	nameLast	weight	debut	name
0	henderi01	2295	Rickey	Henderson	180.0	1979-06-24	Rickey Henderson
1	cobbty01	2246	Ty	Cobb	175.0	1905-08-30	Ty Cobb
2	bondsba01	2227	Barry	Bonds	185.0	1986-05-30	Barry Bonds
3	aaronha01	2174	Hank	Aaron	180.0	1954-04-13	Hank Aaron
4	ruthba01	2174	Babe	Ruth	215.0	1914-07-11	Babe Ruth

	playerID	sum	nameFirst	nameLast	weight	debut	name	debutYear
0	henderi01	2295	Rickey	Henderson	180.0	1979-06-24	Rickey Henderson	1979
1	cobbty01	2246	Ty	Cobb	175.0	1905-08-30	Ty Cobb	1905
2	bondsba01	2227	Barry	Bonds	185.0	1986-05-30	Barry Bonds	1986
3	aaronha01	2174	Hank	Aaron	180.0	1954-04-13	Hank Aaron	1954
4	ruthba01	2174	Babe	Ruth	215.0	1914-07-11	Babe Ruth	1914

Highest runs scorers

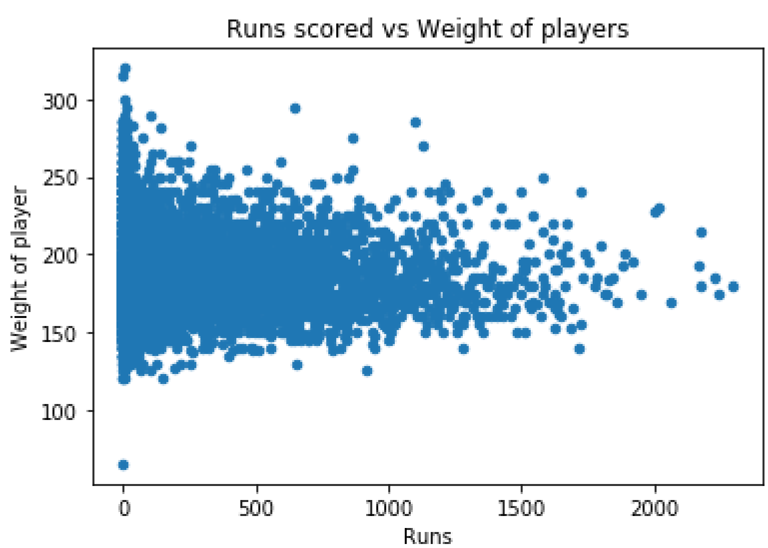
In [137]:
```
most_runs_df = merged_df[:10]
display(most_runs_df)

ax = most_runs_df.plot.bar(x='name', y='sum', rot=60, figsize=(12,8), lege
plt.ylabel('Runs')
plt.xlabel('Players')
plt.title('Highest runs scorers (Top 10)')
plt.show()
```

	playerID	sum	nameFirst	nameLast	weight	debut	name	debutYear
0	henderi01	2295	Rickey	Henderson	180.0	1979-06-24	Rickey Henderson	1979
1	cobbty01	2246	Ty	Cobb	175.0	1905-08-30	Ty Cobb	1905
2	bondsba01	2227	Barry	Bonds	185.0	1986-05-30	Barry Bonds	1986
3	aaronha01	2174	Hank	Aaron	180.0	1954-04-13	Hank Aaron	1954
4	ruthba01	2174	Babe	Ruth	215.0	1914-07-11	Babe Ruth	1914
5	rosepe01	2165	Pete	Rose	192.0	1963-04-08	Pete Rose	1963
6	mayswi01	2062	Willie	Mays	170.0	1951-05-25	Willie Mays	1951
7	rodrial01	2021	Alex	Rodriguez	230.0	1994-07-08	Alex Rodriguez	1994
8	ansonca01	1999	Cap	Anson	227.0	1871-05-06	Cap Anson	1871
9	musiast01	1949	Stan	Musial	175.0	1941-09-17	Stan Musial	1941



Scatterplot - Runs scored vs Weights of players

In [138]:
```
ax = merged_df.plot.scatter(x='sum', y='weight')
plt.xlabel('Runs')
plt.ylabel('Weight of player')
plt.title('Runs scored vs Weight of players')
plt.show()
```



Standardizing the variables

In [139]:
```
from sklearn import preprocessing
from scipy import stats

merged_df['sum_std'] = stats.zscore(merged_df['sum'])
merged_df['weight_std'] = stats.zscore(merged_df['weight'])

merged_df
```

Out[139]:

	playerID	sum	nameFirst	nameLast	weight	debut	name	debutYear	sum_std
0	henderi01	2295	Rickey	Henderson	180.0	1979-06-24	Rickey Henderson	1979	9.039421
1	cobbty01	2246	Ty	Cobb	175.0	1905-08-30	Ty Cobb	1905	8.837104
2	bondsba01	2227	Barry	Bonds	185.0	1986-05-30	Barry Bonds	1986	8.758655
3	aaronha01	2174	Hank	Aaron	180.0	1954-04-13	Hank Aaron	1954	8.539823
4	ruthba01	2174	Babe	Ruth	215.0	1914-07-11	Babe Ruth	1914	8.539823
5	rosepe01	2165	Pete	Rose	192.0	1963-04-08	Pete Rose	1963	8.502663
6	mayswi01	2062	Willie	Mays	170.0	1951-05-25	Willie Mays	1951	8.077385
7	rodrial01	2021	Alex	Rodriguez	230.0	1994-07-08	Alex Rodriguez	1994	7.908100
8	ansonca01	1999	Cap	Anson	227.0	1871-05-06	Cap Anson	1871	7.817264
9	musiast01	1949	Stan	Musial	175.0	1941-09-17	Stan Musial	1941	7.610818
10	jeterde01	1923	Derek	Jeter	195.0	1995-05-29	Derek Jeter	1995	7.503467
11	gehrilo01	1888	Lou	Gehrig	200.0	1923-06-15	Lou Gehrig	1923	7.358955
12	speaktr01	1882	Tris	Speaker	193.0	1907-09-12	Tris Speaker	1907	7.334182
13	ottme01	1859	Mel	Ott	170.0	1926-04-27	Mel Ott	1926	7.239217
14	biggicr01	1844	Craig	Biggio	185.0	1988-06-26	Craig Biggio	1988	7.177283
15	robinfr02	1829	Frank	Robinson	183.0	1956-04-17	Frank Robinson	1956	7.115349
16	collied01	1821	Eddie	Collins	175.0	1906-09-17	Eddie Collins	1906	7.082318
17	yastrca01	1816	Carl	Yastrzemski	175.0	1961-04-11	Carl Yastrzemski	1961	7.061674
18	willite01	1798	Ted	Williams	205.0	1939-04-20	Ted Williams	1939	6.987353

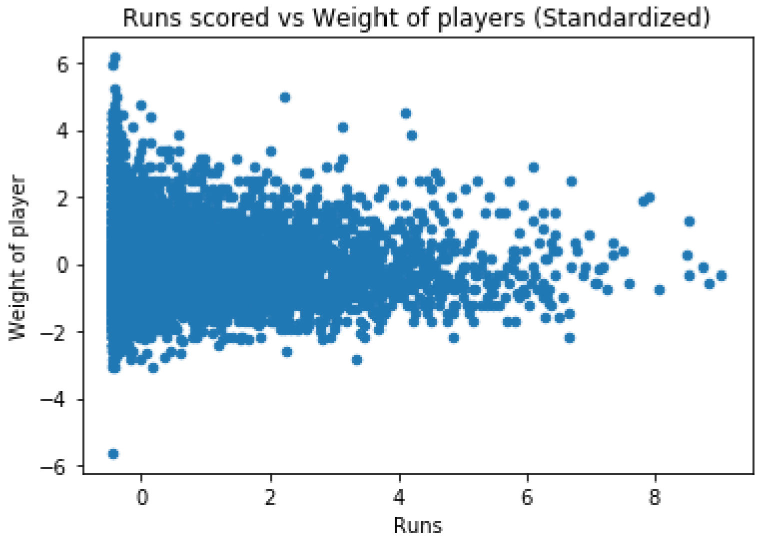
	playerID	sum	nameFirst	nameLast	weight	debut	name	debutYear	sum_std
19	molitpa01	1782	Paul	Molitor	185.0	1978-04-07	Paul Molitor	1978	6.921291
20	gehrich01	1774	Charlie	Gehringer	180.0	1924-09-22	Charlie Gehringer	1924	6.888260
21	foxxji01	1751	Jimmie	Foxx	195.0	1925-05-01	Jimmie Foxx	1925	6.793295
22	wagneho01	1739	Honus	Wagner	200.0	1897-07-19	Honus Wagner	1897	6.743748
23	orourji01	1729	Jim	O'Rourke	185.0	1872-04-26	Jim O'Rourke	1872	6.702459
24	pujolal01	1723	Albert	Pujols	240.0	2001-04-02	Albert Pujols	2001	6.677685
25	burkeje01	1720	Jesse	Burkett	155.0	1890-04-22	Jesse Burkett	1890	6.665298
26	keelewi01	1719	Willie	Keeler	140.0	1892-09-30	Willie Keeler	1892	6.661170
27	hamilbi01	1697	Billy	Hamilton	165.0	1888-07-31	Billy Hamilton	1888	6.570334
28	mcphebi01	1684	Bid	McPhee	152.0	1882-05-02	Bid McPhee	1882	6.516658
29	mantlmi01	1677	Mickey	Mantle	195.0	1951-04-17	Mickey Mantle	1951	6.487755
...	...	...	...	...	...	...	...	...	...
19151	leeza01	0	Zach	Lee	227.0	2015-07-25	Zach Lee	2015	-0.436424
19152	leroyjo01	0	John	LeRoy	175.0	1997-09-26	John LeRoy	1997	-0.436424
19153	lerouch01	0	Chris	Leroux	225.0	2009-05-26	Chris Leroux	2009	-0.436424
19154	lerewan01	0	Anthony	Lerew	225.0	2005-09-04	Anthony Lerew	2005	-0.436424
19155	leovijo01	0	John	Leovich	200.0	1941-05-01	John Leovich	1941	-0.436424
19156	leoporu01	0	Rudy	Leopold	160.0	1928-07-04	Rudy Leopold	1928	-0.436424
19157	leoniz01	0	Izzy	Leon	160.0	1945-06-21	Izzy Leon	1945	-0.436424
19158	leonedo01	0	Dominic	Leone	210.0	2014-04-06	Dominic Leone	2014	-0.436424
19159	leonda01	0	Danny	Leon	170.0	1992-06-06	Danny Leon	1992	-0.436424
19160	leonar02	0	Arcenio	Leon	222.0	2017-05-28	Arcenio Leon	2017	-0.436424
19161	leonar01	0	Arnold	Leon	210.0	2015-04-22	Arnold Leon	2015	-0.436424
19162	leonade01	0	Dennis	Leonard	190.0	1974-09-04	Dennis Leonard	1974	-0.436424

	playerID	sum	nameFirst	nameLast	weight	debut	name	debutYear	sum_std
19164	lennoed02	0	Ed	Lennon	170.0	1928-06-30	Ed Lennon	1928	-0.436424
19165	lembost01	0	Steve	Lembo	185.0	1950-09-16	Steve Lembo	1950	-0.436424
19166	lemanda01	0	Dave	Lemanczyk	235.0	1973-04-15	Dave Lemanczyk	1973	-0.436424
19167	lelivbi01	0	Bill	Lelivelt	195.0	1909-07-19	Bill Lelivelt	1909	-0.436424
19168	leitndu01	0	Dummy	Leitner	120.0	1901-06-29	Dummy Leitner	1901	-0.436424
19169	leithbi01	0	Bill	Leith	208.0	1899-09-25	Bill Leith	1899	-0.436424
19170	leitema02	0	Mark	Leiter	195.0	2017-04-28	Mark Leiter	2017	-0.436424
19171	leistjo01	0	John	Leister	200.0	1987-05-28	John Leister	1987	-0.436424
19172	leipeda01	0	Dave	Leiper	160.0	1984-09-02	Dave Leiper	1984	-0.436424
19173	leinhbi01	0	Bill	Leinhauser	150.0	1912-05-18	Bill Leinhauser	1912	-0.436424
19174	leifeel01	0	Elmer	Leifer	170.0	1921-09-07	Elmer Leifer	1921	-0.436424
19175	leicejo01	0	Jon	Leicester	220.0	2004-06-09	Jon Leicester	2004	-0.436424
19176	lehrno01	0	Norm	Lehr	168.0	1926-05-20	Norm Lehr	1926	-0.436424
19177	lehewji01	0	Jim	Lehew	185.0	1961-09-13	Jim Lehew	1961	-0.436424
19178	lehenre01	0	Regis	Leheny	180.0	1932-05-21	Regis Leheny	1932	-0.436424
19179	leftwph01	0	Phil	Leftwich	205.0	1993-07-29	Phil Leftwich	1993	-0.436424
19180	leflewa01	0	Wade	Lefler	162.0	1924-04-16	Wade Lefler	1924	-0.436424
19181	zychto01	0	Tony	Zych	190.0	2015-09-04	Tony Zych	2015	-0.436424

18439 rows × 10 columns

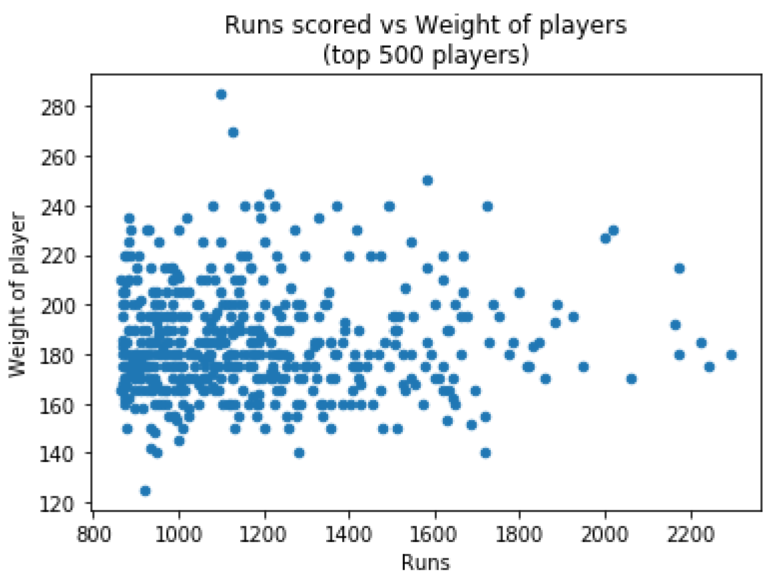
Runs scored vs Weight of players (Standardized)

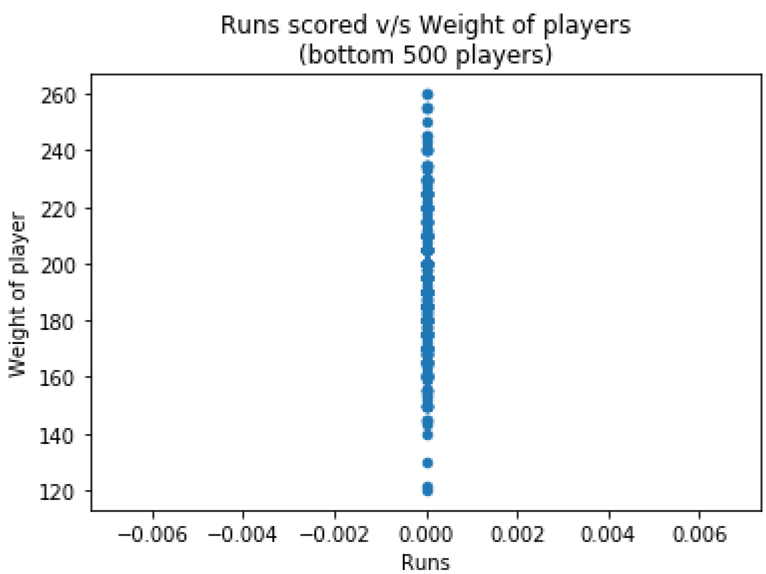
In [140]:
```
ax = merged_df.plot.scatter(x='sum_std', y='weight_std')
plt.xlabel('Runs')
plt.ylabel('Weight of player')
plt.title('Runs scored vs Weight of players (Standardized)')
plt.show()
```

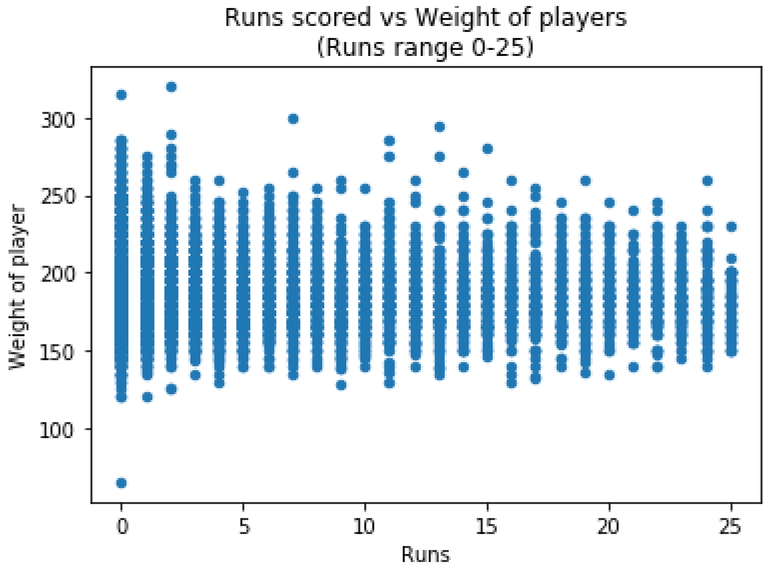


Digging further in Runs scored vs Weight of players

In [141]:
```
top_500_players = merged_df[:500]
bottom_500_players = merged_df[-500:].reset_index(drop=True)

ax = top_500_players.plot.scatter(x='sum', y='weight')
plt.xlabel('Runs')
plt.ylabel('Weight of player')
plt.title('Runs scored vs Weight of players\n(top 500 players)')
plt.show()

ax = bottom_500_players.plot.scatter(x='sum', y='weight')
plt.xlabel('Runs')
plt.ylabel('Weight of player')
plt.title('Runs scored v/s Weight of players\n(bottom 500 players)')
plt.show()

bottom_players_runs_0_to_26 = merged_df[-11950:]
ax = bottom_players_runs_0_to_26.plot.scatter(x='sum', y='weight')
plt.xlabel('Runs')
plt.ylabel('Weight of player')
plt.title('Runs scored vs Weight of players\n(Runs range 0-25)')
plt.show()
```





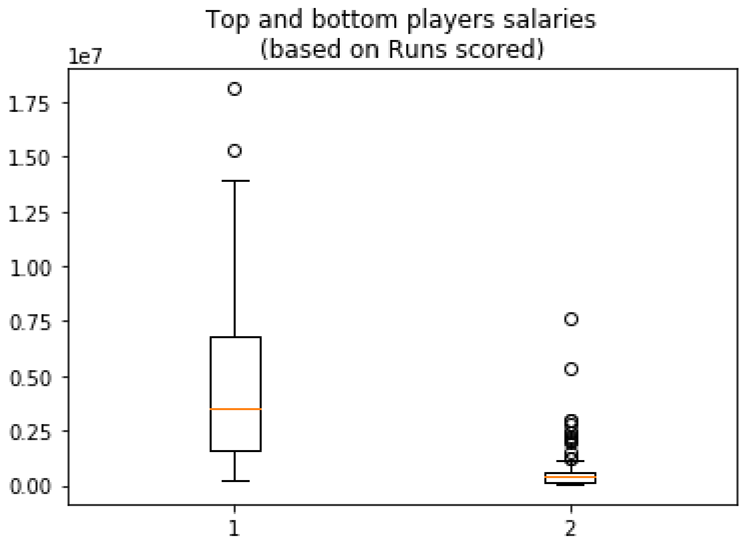


Runs affecting salaries

In [142]:
```
salaries = pd.read_csv('baseballdatabank-master/core/Salaries.csv', useco

# Getting salaries for top 500 players
player_salaries_df = pd.DataFrame(columns=["avgSalary"])
df_index = 0
for index, row in top_500_players.iterrows():
    playerID = row['playerID']
    avg_salary = salaries.loc[salaries['playerID'] == playerID].salary.mea
    player_salaries_df.loc[df_index] = [avg_salary]
    df_index += 1
top_500_players_with_salaries = pd.concat([top_500_players, player_salari
top_500_players_with_salaries = top_500_players_with_salaries.dropna()
top_500_players_with_salaries

# Getting salaries for bottom 500 players
player_salaries_df = pd.DataFrame(columns=["avgSalary"])
df_index = 0
for index, row in bottom_500_players.iterrows():
    playerID = row['playerID']
    avg_salary = salaries.loc[salaries['playerID'] == playerID].salary.mea
    player_salaries_df.loc[df_index] = [avg_salary]
    df_index += 1
bottom_500_players_with_salaries = pd.concat([bottom_500_players, player_s
bottom_500_players_with_salaries = bottom_500_players_with_salaries.dropna
```

In [143]:
```
data = [top_500_players_with_salaries['avgSalary'], bottom_500_players_wi
fig1, ax1 = plt.subplots()
ax1.set_title('Top and bottom players salaries\n(based on Runs scored)')
ax1.boxplot(data)
```

Out[143]:
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7f1659e55ac8>,
  <matplotlib.lines.Line2D at 0x7f1659e55358>,
  <matplotlib.lines.Line2D at 0x7f166ccbc978>,
  <matplotlib.lines.Line2D at 0x7f166ccbcf60>],
 'caps': [<matplotlib.lines.Line2D at 0x7f1659e55d30>,
  <matplotlib.lines.Line2D at 0x7f1659e552b0>,
  <matplotlib.lines.Line2D at 0x7f166ccbc5f8>,
  <matplotlib.lines.Line2D at 0x7f166ccbc0b8>],
 'boxes': [<matplotlib.lines.Line2D at 0x7f1659e55a58>,
  <matplotlib.lines.Line2D at 0x7f166cfa9588>],
 'medians': [<matplotlib.lines.Line2D at 0x7f166cfa9978>,
  <matplotlib.lines.Line2D at 0x7f166ce7c0b8>],
 'fliers': [<matplotlib.lines.Line2D at 0x7f166cfa9630>,
  <matplotlib.lines.Line2D at 0x7f166ce7c320>],
 'means': []}
```



Popularity of Baseball over the decades(based on number of players debuted each year)

In [144]:
```
min_year = merged_df['debutYear'].min()
max_year = merged_df['debutYear'].max()

min_year
max_year

min_year = 1870
max_year = 2020

baseball_pop_df = pd.DataFrame(columns=['minYear','maxYear'])
df_index = 0

i = min_year
while(i < max_year):
    min_y = i
    i = i+9
    max_y = i
    i = i+1
    baseball_pop_df.loc[df_index] = [str(min_y), str(max_y)]
    df_index += 1

baseball_pop_df

count_df = pd.DataFrame(columns=['playersOnboarded'])
df_index = 0

for index, row in baseball_pop_df.iterrows():
    min_y = int(row['minYear'])
    max_y = int(row['maxYear'])
    count_df.loc[df_index] = len(merged_df.loc[(merged_df['debutYear'] >=
    df_index += 1


baseball_pop_df = pd.concat([baseball_pop_df, count_df], axis=1)
baseball_pop_df['decade'] = baseball_pop_df['minYear'] + '-' +  baseball_
display(baseball_pop_df)



ax = baseball_pop_df.plot.bar(x='decade', y='playersOnboarded', rot=60, f
plt.ylabel('Number of players debuted')
plt.xlabel('Decades')
plt.title('Popularity of Baseball over the decades\n(based on number of p
plt.show()
```

	minYear	maxYear	playersOnboarded	decade
0	1870	1879	292	1870-1879
1	1880	1889	692	1880-1889
2	1890	1899	654	1890-1899
3	1900	1909	951	1900-1909
4	1910	1919	1508	1910-1919

	minYear	maxYear	playersOnboarded	decade
5	1920	1929	1200	1920-1929
6	1930	1939	1038	1930-1939
7	1940	1949	1168	1940-1949
8	1950	1959	1070	1950-1959
9	1960	1969	1251	1960-1969
10	1970	1979	1316	1970-1979
11	1980	1989	1458	1980-1989
12	1990	1999	1879	1990-1999
13	2000	2009	2076	2000-2009
14	2010	2019	1886	2010-2019