In [26]:

```
# numpy and pandas for data manipulation
import numpy as np
import pandas as pd

# sklearn preprocessing for dealing with categorical variables
from sklearn.preprocessing import LabelEncoder

# File system manangement
import os

# Suppress warnings
import warnings
warnings.filterwarnings('ignore')

# matplotlib and seaborn for plotting
import matplotlib.pyplot as plt
import seaborn as sns
```

In [27]:

```
# List files available
print(os.listdir("../Colab Notebooks"))
```

```
['COMPSCIX433.3-009 Function.ipynb', 'COMPSCIX433.3-009', 'COMPSCIX433.1-002  Fashion MNIST reader.i
pynb', 'COMPSCIX433.1-002 How to Search Credit Card.ipynb', '.DS_Store', 'COMPSCIX433.6-007', 'kaggl
e', 'WorldRecords.csv', 'COMPSCIX433.1-002   Example 8 Metrics and Validation', 'fashionmnist', 'COM
PSCIX433.3-009 Assingment 2.ipynb', 'input', 'COMPSCIX433.1-002  Assignment Web Scraping 1.0.ipynb',
'COMPSCIX433.3-009 Data-Structures-PG.ipynb', 'Visualization.ipynb', 'COMPSCIX433.3-009 Pandas_DataF
rame-092519.ipynb', 'COMPSCIX433.3-009 Numpy.ipynb', 'COMPSCIX433.3-009 Assingment1.ipynb', 'Pandas_
DataFrame-093019.ipynb', 'WorldRecords_old.csv', 'ASSIGNMENT4.pdf', 'Scripts_export', 'COMPSCIX433.3
-009 Lecture1-basic-syntax.ipynb', 'COMPSCIX433.1-002  Example 10 Autoregression', 'COMPSCIX433.1-00
2  Assingment WebScraping.ipynb', 'COMPSCIX433.3-009 Practice.ipynb', 'test.pdf', 'COMPSCIX433.1-002
Example 7 Input Vectorization.ipynb', 'COMPSCIX433.1-002', 'COMPSCIX433.1-002 WebScraping Self Tryou
t 1.ipynb', 'implement-perceptron-algorithm-scratch-python.ipynb', 'COMPSCIX433.1-002 Example 4 Cate
gorical Data Analysis.ipynb', 'COMPSCIX433.1-002 Example 3 Tabulation and Chi-Square Testing.ipynb',
'COMPSCIX433.1-009 Example 9 Simple Classifiers and Regressors', 'COMPSCIX433.3-009 Assignment 4.ipy
nb', '.ipynb_checkpoints', 'Visualization.pdf', 'COMPSCIX433.1-002 How to Search Credit Card (1).ipy
nb', 'COMPSCIX433.3-009 Assignment 3.ipynb', 'COMPSCIX433.1-002 Example 6 A Simple Perceptron Classi
fier.ipynb', 'Data', 'COMPSCIX433.1-002 Example 2 Finding how people talk about dogs.ipynb', 'Edmund
s-Data Analysis - Cross Continent Review.ipynb', 'COMPSCIX433.1-002  Web Scraping Self Try.ipynb']
```

In [28]:

```
# Load dataset
# I cleaned Up the Data manually before loading as the Data was having a non UTF 8 Char.
app_data = pd.read_csv('../Colab Notebooks/WorldRecords_old.csv', engine='python')
print('Training data shape: ', app_data.shape)
app_data.head(25)
```

```
Training data shape:  (285, 7)
```

Out[28]:

	Event	Type	Record	Athlete	Nationality	Location	Year
0	Mens 100m	time	10.06	Bob Hayes	United States	Tokyo, Japan	1964
1	Mens 100m	time	10.03	Jim Hines	United States	Sacramento, USA	1968
2	Mens 100m	time	10.02	Charles Greene	United States	Mexico City, Mexico	1968
3	Mens 100m	time	9.95	Jim Hines	United States	Mexico City, Mexico	1968
4	Mens 100m	time	9.93	Calvin Smith	United States	Colorado Springs, USA	1983
5	Mens 100m	time	9.92	Carl Lewis	United States	Seoul, South Korea	1988
6	Mens 100m	time	9.90	Leroy Burrell	United States	New York, USA	1991
7	Mens 100m	time	9.86	Carl Lewis	United States	Tokyo, Japan	1991
8	Mens 100m	time	9.85	Leroy Burrell	United States	Lausanne, Switzerland	1994
9	Mens 100m	time	9.84	Donovan Bailey	Canada	Atlanta, USA	1996
10	Mens 100m	time	9.79	Maurice Greene	United States	Athens, Greece	1999
11	Mens 100m	time	9.78	Tim Montgomery	United States	Paris, France	2002
12	Mens 100m	time	9.77	Asafa Powell	Jamaica	Athens, Greece	2005
13	Mens 100m	time	9.74	Asafa Powell	Jamaica	Rieti, Italy	2007
14	Mens 100m	time	9.72	Usain Bolt	Jamaica	New York, USA	2008
15	Mens 100m	time	9.69	Usain Bolt	Jamaica	Beijing, China	2008
16	Mens 100m	time	9.58	Usain Bolt	Jamaica	Berlin, Germany	2009
17	Womens 100m	time	11.07	Wyomia Tyus	�United States	Mexico City, Mexico	1968
18	Womens 100m	time	11.07	Renate Stecher	�East Germany	Munich, West Germany	1972
19	Womens 100m	time	11.04	Inge Helten	�West Germany	F�rth, West Germany	1976
20	Womens 100m	time	11.01	Annegret Richter	�West Germany	Montreal, Canada	1976
21	Womens 100m	time	10.88	Marlies Oelsner	�East Germany	Dresden, East Germany	1977
22	Womens 100m	time	10.88	Marlies G�hr	�East Germany	Karl-Marx-Stadt, East Germany	1982
23	Womens 100m	time	10.81	Marlies G�hr	�East Germany	Berlin, East Germany	1983
24	Womens 100m	time	10.79	Evelyn Ashford	�United States	US Air Force Academy, United States	1983

In [29]:

```
#Cleaning the Non UTF 8 Char From Dataframe

app_data['Nationality'] = app_data['Nationality'].str.encode('ascii', 'ignore').str.decode('ascii')
app_data['Athlete'] = app_data['Athlete'].str.encode('ascii', 'ignore').str.decode('ascii')
app_data['Location'] = app_data['Location'].str.encode('ascii', 'ignore').str.decode('ascii')
print('Training data shape: ', app_data.shape)
app_data.head(25)
```

```
Training data shape:  (285, 7)
```

Out[29]:

	Event	Type	Record	Athlete	Nationality	Location	Year
0	Mens 100m	time	10.06	Bob Hayes	United States	Tokyo, Japan	1964
1	Mens 100m	time	10.03	Jim Hines	United States	Sacramento, USA	1968
2	Mens 100m	time	10.02	Charles Greene	United States	Mexico City, Mexico	1968
3	Mens 100m	time	9.95	Jim Hines	United States	Mexico City, Mexico	1968
4	Mens 100m	time	9.93	Calvin Smith	United States	Colorado Springs, USA	1983
5	Mens 100m	time	9.92	Carl Lewis	United States	Seoul, South Korea	1988
6	Mens 100m	time	9.90	Leroy Burrell	United States	New York, USA	1991
7	Mens 100m	time	9.86	Carl Lewis	United States	Tokyo, Japan	1991
8	Mens 100m	time	9.85	Leroy Burrell	United States	Lausanne, Switzerland	1994
9	Mens 100m	time	9.84	Donovan Bailey	Canada	Atlanta, USA	1996
10	Mens 100m	time	9.79	Maurice Greene	United States	Athens, Greece	1999
11	Mens 100m	time	9.78	Tim Montgomery	United States	Paris, France	2002
12	Mens 100m	time	9.77	Asafa Powell	Jamaica	Athens, Greece	2005
13	Mens 100m	time	9.74	Asafa Powell	Jamaica	Rieti, Italy	2007
14	Mens 100m	time	9.72	Usain Bolt	Jamaica	New York, USA	2008
15	Mens 100m	time	9.69	Usain Bolt	Jamaica	Beijing, China	2008
16	Mens 100m	time	9.58	Usain Bolt	Jamaica	Berlin, Germany	2009
17	Womens 100m	time	11.07	Wyomia Tyus	United States	Mexico City, Mexico	1968
18	Womens 100m	time	11.07	Renate Stecher	East Germany	Munich, West Germany	1972
19	Womens 100m	time	11.04	Inge Helten	West Germany	Frth, West Germany	1976
20	Womens 100m	time	11.01	Annegret Richter	West Germany	Montreal, Canada	1976
21	Womens 100m	time	10.88	Marlies Oelsner	East Germany	Dresden, East Germany	1977
22	Womens 100m	time	10.88	Marlies Ghr	East Germany	Karl-Marx-Stadt, East Germany	1982
23	Womens 100m	time	10.81	Marlies Ghr	East Germany	Berlin, East Germany	1983
24	Womens 100m	time	10.79	Evelyn Ashford	United States	US Air Force Academy, United States	1983

```
In [30]: #United States is present in data as USA and United State. Making it consistent
         app_data.replace("USA", "United States", inplace=True)
```

Q1. How many different types of events (e.g. “Mens 100m”, “Womens shotput” etc) are represented in the dataset

```
In [31]: print("Total Number of Different Events: ", len(app_data['Event'].unique()))
         print("Names of Different Events: ", app_data['Event'].unique())

         # Number of each type of column
         app_data.dtypes.value_counts()
         app_data.select_dtypes('object').apply(pd.Series.nunique, axis = 0)
```

```
Total Number of Different Events:  10
Names of Different Events:  ['Mens 100m' 'Womens 100m' 'Mens 800m' 'Womens 800m' 'Mens TripleJump'
 'Mens Mile' 'Womens Mile' 'Mens Polevault' 'Mens Shotput'
 'Womens Shotput']
```

```
Out[31]: Event           10
         Type             2
         Athlete        150
         Nationality     42
         Location       147
         dtype: int64
```

Q2.In what year did Usain Bolt first break the world record for the Men’s 100m?

```
In [32]: print("First Year When Usain Bolt Broke the Men 100 m race : ", app_data[(app_data['Event'] == 'Mens
          100m') & (app_data['Athlete'] == 'Usain Bolt')].Year.min())
```

```
First Year When Usain Bolt Broke the Men 100 m race :  2008
```

Q3. Which variable tells us the record setting time or distance? The variable name in the data set is? What type of the variable is this?

In [33]:

```
# Type colum tells us if the event record is in time or distance.
app_data['Type'].value_counts()
# The variable name is Type
print("Data Type for Type Variable : ", app_data['Type'].dtypes)
```

```
Data Type for Type Variable :  object
```

Q4. Create a subset of the dataset that contains only the world record cases for men’s shotput and women’s shotput

In [34]:

```
data_subset = app_data[(app_data['Event'] == 'Womens Shotput') | (app_data['Event'] == 'Mens Shotput'
)]
print("Subset Data Type : ", data_subset.head())
```

```
Subset Data Type :              Event      Type  Record           Athlete    Nationality  \
205  Mens Shotput  distance   17.68  Charlie Fonville  United States
206  Mens Shotput  distance   17.79         Jim Fuchs  United States
207  Mens Shotput  distance   17.82         Jim Fuchs  United States
208  Mens Shotput  distance   17.90         Jim Fuchs  United States
209  Mens Shotput  distance   17.95         Jim Fuchs  United States

               Location  Year
205      Lawrence, U.S.  1948
206        Oslo, Norway  1949
207   Los Angeles, U.S.  1950
208       Visby, Sweden  1950
209  Eskilstuna, Sweden  1950
```

Q5. Create a scatter plot of the year and record shotput distance one for men and one for women.

In [35]:

```
data_subset.shape
```

Out[35]:

```
(80, 7)
```

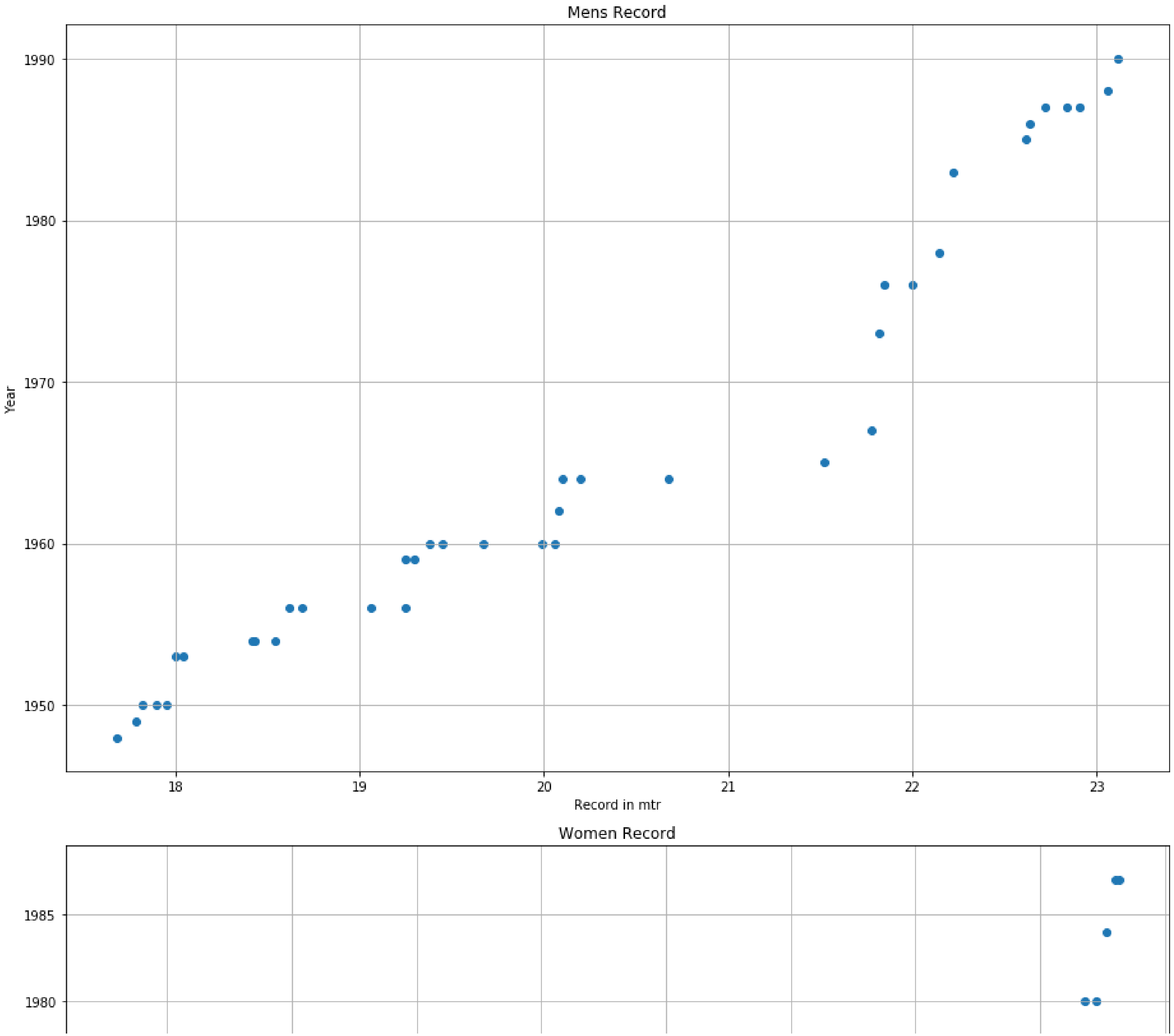
In [36]:

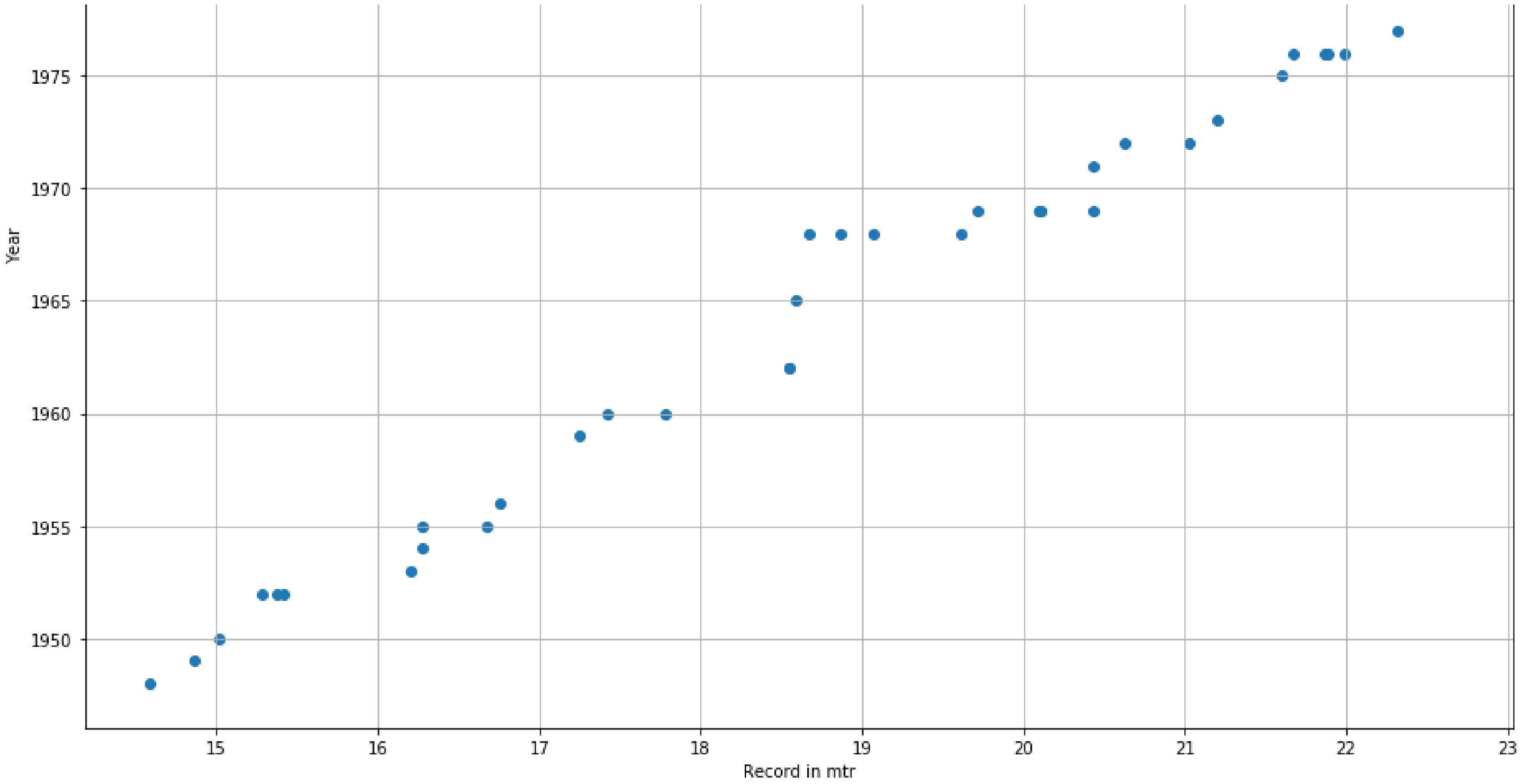
```
t = data_subset['Year']
print(len(t))
s = data_subset['Record']
print(len(s))
```

```
80
80
```

In [37]:

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(13,18))

tw = data_subset[data_subset['Event'] == 'Womens Shotput']['Year']
tm = data_subset[data_subset['Event'] == 'Mens Shotput']['Year']

sw = data_subset[data_subset['Event'] == 'Womens Shotput']['Record']
sm = data_subset[data_subset['Event'] == 'Mens Shotput']['Record']

plt.subplot(2, 1, 1)
plt.scatter(sm, tm)
plt.xlabel('Record in mtr')
plt.ylabel('Year')
plt.title('Mens Record')
plt.grid(True)


plt.subplot(2, 1, 2)

plt.scatter(sw, tw)
plt.xlabel('Record in mtr')
plt.ylabel('Year')
plt.title('Women Record')
plt.grid(True)


plt.tight_layout()
plt.show()
```


Mens Record
1990
1980
1970
1960
1950
Year
18
19
20
21
22
23
Record in mtr
Women Record
1985
1980



Q6. Find the average/mean time for each event. How many athletes have time more than average in each event.

In [38]:

```
print(app_data.groupby('Event')['Record'].agg([np.mean, np.sum, np.std]))
```

```
                       mean      sum        std
Event                                          
Mens 100m          9.848824   167.43   0.130330
Mens 800m        105.345833  2528.30   3.832241
Mens Mile        237.493750  7599.80   8.982490
Mens Polevault     5.608909   308.49   0.400480
Mens Shotput      20.194872   787.60   1.829229
Mens TripleJump   16.671200   416.78   0.824537
Womens 100m       10.880000   108.80   0.180801
Womens 800m      127.851724  3707.70  10.377951
Womens Mile      263.653846  3427.50   8.366363
Womens Shotput    19.139756   784.73   2.572595
```

In [39]:

```
mean_val = app_data.groupby('Event')['Record'].agg([np.mean])
```

In [40]:

```
mean_val
```

Out[40]:

	mean
Event	
Mens 100m	9.848824
Mens 800m	105.345833
Mens Mile	237.493750
Mens Polevault	5.608909
Mens Shotput	20.194872
Mens TripleJump	16.671200
Womens 100m	10.880000
Womens 800m	127.851724
Womens Mile	263.653846
Womens Shotput	19.139756

In [41]:

```
app_data.head()
```

Out[41]:

	Event	Type	Record	Athlete	Nationality	Location	Year
0	Mens 100m	time	10.06	Bob Hayes	United States	Tokyo, Japan	1964
1	Mens 100m	time	10.03	Jim Hines	United States	Sacramento, USA	1968
2	Mens 100m	time	10.02	Charles Greene	United States	Mexico City, Mexico	1968
3	Mens 100m	time	9.95	Jim Hines	United States	Mexico City, Mexico	1968
4	Mens 100m	time	9.93	Calvin Smith	United States	Colorado Springs, USA	1983

In [42]:

```
app_data['mean'] = app_data.groupby('Event')['Record'].transform('mean')
app_data.head(20)
```

Out[42]:

	Event	Type	Record	Athlete	Nationality	Location	Year	mean
0	Mens 100m	time	10.06	Bob Hayes	United States	Tokyo, Japan	1964	9.848824
1	Mens 100m	time	10.03	Jim Hines	United States	Sacramento, USA	1968	9.848824
2	Mens 100m	time	10.02	Charles Greene	United States	Mexico City, Mexico	1968	9.848824
3	Mens 100m	time	9.95	Jim Hines	United States	Mexico City, Mexico	1968	9.848824
4	Mens 100m	time	9.93	Calvin Smith	United States	Colorado Springs, USA	1983	9.848824
5	Mens 100m	time	9.92	Carl Lewis	United States	Seoul, South Korea	1988	9.848824
6	Mens 100m	time	9.90	Leroy Burrell	United States	New York, USA	1991	9.848824
7	Mens 100m	time	9.86	Carl Lewis	United States	Tokyo, Japan	1991	9.848824
8	Mens 100m	time	9.85	Leroy Burrell	United States	Lausanne, Switzerland	1994	9.848824
9	Mens 100m	time	9.84	Donovan Bailey	Canada	Atlanta, USA	1996	9.848824
10	Mens 100m	time	9.79	Maurice Greene	United States	Athens, Greece	1999	9.848824
11	Mens 100m	time	9.78	Tim Montgomery	United States	Paris, France	2002	9.848824
12	Mens 100m	time	9.77	Asafa Powell	Jamaica	Athens, Greece	2005	9.848824
13	Mens 100m	time	9.74	Asafa Powell	Jamaica	Rieti, Italy	2007	9.848824
14	Mens 100m	time	9.72	Usain Bolt	Jamaica	New York, USA	2008	9.848824
15	Mens 100m	time	9.69	Usain Bolt	Jamaica	Beijing, China	2008	9.848824
16	Mens 100m	time	9.58	Usain Bolt	Jamaica	Berlin, Germany	2009	9.848824
17	Womens 100m	time	11.07	Wyomia Tyus	United States	Mexico City, Mexico	1968	10.880000
18	Womens 100m	time	11.07	Renate Stecher	East Germany	Munich, West Germany	1972	10.880000
19	Womens 100m	time	11.04	Inge Helten	West Germany	Frth, West Germany	1976	10.880000

In [43]:

```
# Number of Athelete who have their record more than the average of the respective events
app_data[app_data['Record'] > app_data['mean']].groupby('Event')['Athlete'].agg(['count'])
```

Out[43]:

	count
Event	
Mens 100m	9
Mens 800m	10
Mens Mile	15
Mens Polevault	31
Mens Shotput	16
Mens TripleJump	12
Womens 100m	4
Womens 800m	13
Womens Mile	5
Womens Shotput	21

Q7. Select the athlete who took most time in men’s 100m and women’s event.

In [44]:

```
app_data.groupby('Event')['Record', 'Athlete'].max().loc['Mens 100m']['Athlete']
```

Out[44]:

```
'Usain Bolt'
```

In [45]:

```
app_data.groupby('Event')['Record', 'Athlete'].max().loc['Womens 100m']['Athlete']
```

Out[45]:

```
'Wyomia Tyus'
```

Q8. Which country won maximum times of men’s 100m event?

In [46]:

```
countries_with_win = app_data.groupby(['Event'])['Nationality'].max().loc['Mens 100m']
countries_with_win
```

Out[46]:

```
'United States'
```

Q9. How many athletes are there in each event?

In [47]:

```
app_data.groupby(['Event'])['Athlete'].count()
```

Out[47]:

```
Event
Mens 100m          17
Mens 800m          24
Mens Mile          32
Mens Polevault     55
Mens Shotput       39
Mens TripleJump    25
Womens 100m        10
Womens 800m        29
Womens Mile        13
Womens Shotput     41
Name: Athlete, dtype: int64
```

Q10. Which country has maximum wins?

In [48]:

```
print(app_data['Nationality'].value_counts().argmax())
```

```
United States
```

In [49]:

```
print(app_data['Nationality'].mode())
```

```
0    United States
dtype: object
```

In [50]:

```
app_data['Nationality'].value_counts()[:1].index.tolist()
```

Out[50]:

```
['United States']
```