DataSet

https://www.kaggle.com/datasets/aiaiaidavid/the-big-dataset-of-ultra-marathon-running?resource=download (https://www.kaggle.com/datasets/aiaiaidavid/the-big-dataset-of-ultra-marathon-running?resource=download)

In [2]:
```
import pandas as pd
import seaborn as sns
import numpy as np
```

In [3]:
```
df = pd.read_csv("two_cen_race.csv")
```

```
C:\Users\lahir\AppData\Local\Temp\ipykernel_1416\1340791803.py:1: DtypeWarning: Columns (11) have mixed
types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv("two_cen_race.csv")
```

In [4]:
```
#pandas detect inconsistent data type in column 11
```

In [5]:
```
df = pd.read_csv("two_cen_race.csv",low_memory = False)
```

In [6]:
```
df.head(10)
```

Out[6]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete club	Athlete country	Athlete year of birth	Athlete gender	Athlete age category	Athlete average speed	Ath
0	2018	06.01.2018	Selva Costera (CHI)	50km	22	4:51:39 h	Tnfrc	CHI	1978.0	M	M35	10.286	
1	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:15:45 h	Roberto Echeverría	CHI	1981.0	M	M35	9.501	
2	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:16:44 h	Puro Trail Osorno	CHI	1987.0	M	M23	9.472	
3	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:34:13 h	Columbia	ARG	1976.0	M	M40	8.976	
4	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:54:14 h	Baguales Trail	CHI	1992.0	M	M23	8.469	
5	2018	06.01.2018	Selva Costera (CHI)	50km	22	6:25:01 h	NaN	ARG	1974.0	M	M40	7.792	
6	2018	06.01.2018	Selva Costera (CHI)	50km	22	6:28:00 h	Los Patagones	ARG	1979.0	F	W35	7.732	
7	2018	06.01.2018	Selva Costera (CHI)	50km	22	6:32:24 h	Reaktiva Chile	CHI	1967.0	F	W50	7.645	
8	2018	06.01.2018	Selva Costera (CHI)	50km	22	6:39:08 h	Puro Trail Osorno	CHI	1985.0	M	M23	7.516	
9	2018	06.01.2018	Selva Costera (CHI)	50km	22	6:45:11 h	Marlene Flores Team	CHI	1976.0	M	M40	7.404	

In [8]:
```
df.shape
```

Out[8]:
```
(7461195, 13)
```

In [9]:
```
df.dtypes
```

Out[9]:
```
Year of event                int64
Event dates                 object
Event name                  object
Event distance/length       object
Event number of finishers    int64
Athlete performance         object
Athlete club                object
Athlete country             object
Athlete year of birth      float64
Athlete gender              object
Athlete age category        object
Athlete average speed       object
Athlete ID                   int64
dtype: object
```

In [10]:
```
# first We have to Clean the Data set
```

In [17]:
```
df
```

Out[17]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete club	Athlete country	Athlete year of birth	Athlete gender	Athlete age category	A a
0	2018	06.01.2018	Selva Costera (CHI)	50km	22	4:51:39 h	Tnfrc	CHI	1978.0	M	M35	
1	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:15:45 h	Roberto Echeverría	CHI	1981.0	M	M35	
2	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:16:44 h	Puro Trail Osorno	CHI	1987.0	M	M23	
3	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:34:13 h	Columbia	ARG	1976.0	M	M40	
4	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:54:14 h	Baguales Trail	CHI	1992.0	M	M23	
...	...	...	...	...	...	...	...	...	...	...	...	
7461190	1995	00.00.1995	La SainteLyon 65 km (FRA)	65km	2	4:33:20 h	NaN	FRA	NaN	M	NaN	1
7461191	1995	00.00.1995	La SainteLyon 65 km (FRA)	65km	2	6:05:15 h	NaN	FRA	NaN	F	NaN	1
7461192	1995	00.00.1995	Szombathely 24 hours running Race (HUN)	24h	3	241.000 km	*Budapest	HUN	1950.0	M	M40	1
7461193	1995	00.00.1995	Szombathely 24 hours running Race (HUN)	24h	3	228.000 km	*Szeged	HUN	1959.0	M	M35	
7461194	1995	00.00.1995	Szombathely 24 hours running Race (HUN)	24h	3	224.000 km	*Pecs	HUN	1958.0	M	M35	

7461195 rows × 13 columns

Filter Data

In [28]:
```
df2 = df[(df["Event distance/length"].isin(["50km","50mi"])) & (df["Year of event"] == 2020)]
```

In [29]:
```
df2 #filtered data
```

Out[29]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete club	Athlete country	Athlete year of birth	Athlete gender	Athlete age category
2538571	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:34:19 h	日本隊	JPN	1965.0	M	M50
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50 h	NaN	AUS	1974.0	M	M45
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40 h	NaN	TPE	1976.0	M	M40
2538574	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:30:49 h	台灣大腳丫長跑協會	TPE	1969.0	F	W50
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47 h	NaN	TPE	1964.0	M	M55
...	...	...	...	...	...	...	...	...	...	...	...
2762404	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:25 h	AKS Polonia Warszawa	POL	1981.0	F	W35
2762405	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:27 h	*Warszawa	POL	1970.0	F	W45
2762406	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:44:18 h	Outdoor Training	POL	1993.0	F	W23
2762407	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:04:50 h	PH Bysewo Gdańsk	POL	1976.0	M	M40
2762408	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:11:43 h	*Nowe Aleksandrowo	POL	1961.0	M	M55

63489 rows × 13 columns

In [34]:
```
df2["Athlete performance"] = df2["Athlete performance"].str.replace("h"," ")
```

```
C:\Users\lahir\AppData\Local\Temp\ipykernel_1416\451958244.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df2["Athlete performance"] = df2["Athlete performance"].str.replace("h"," ")
```

In [35]:
```
df2
```

Out[35]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete club	Athlete country	Athlete year of birth	Athlete gender	Athlete age category
2538571	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:34:19	日本隊	JPN	1965.0	M	M50
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50	NaN	AUS	1974.0	M	M45
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40	NaN	TPE	1976.0	M	M40
2538574	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:30:49	台灣大腳丫長跑協會	TPE	1969.0	F	W50
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47	NaN	TPE	1964.0	M	M55
...	...	...	...	...	...	...	...	...	...	...	...
2762404	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:25	AKS Polonia Warszawa	POL	1981.0	F	W35
2762405	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:27	*Warszawa	POL	1970.0	F	W45
2762406	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:44:18	Outdoor Training	POL	1993.0	F	W23
2762407	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:04:50	PH Bysewo Gdańsk	POL	1976.0	M	M40
2762408	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:11:43	*Nowe Aleksandrowo	POL	1961.0	M	M55

63489 rows × 13 columns

In [59]:
```
df2["Age"] = df["Year of event"] - df["Athlete year of birth"]
```

In [62]:
```
df2["Age"].dropna()
```

Out[62]:
```
2538571    55.0
2538572    46.0
2538573    44.0
2538574    51.0
2538575    56.0
           ...
2762404    39.0
2762405    50.0
2762406    27.0
2762407    44.0
2762408    59.0
Name: Age, Length: 60025, dtype: float64
```

In [82]:
```
df2["Age"].replace(np.inf,0)
```

Out[82]:
```
2538571    55.0
2538572    46.0
2538573    44.0
2538574    51.0
2538575    56.0
           ...
2762404    39.0
2762405    50.0
2762406    27.0
2762407    44.0
2762408    59.0
Name: Age, Length: 63489, dtype: float64
```

In [88]:
```
df2["Age"].fillna(0)
```

Out[88]:
```
2538571    55.0
2538572    46.0
2538573    44.0
2538574    51.0
2538575    56.0
           ...
2762404    39.0
2762405    50.0
2762406    27.0
2762407    44.0
2762408    59.0
Name: Age, Length: 63489, dtype: float64
```

In [171]:
```
df2= df2.drop(["Athlete club","Athlete country","Athlete age category","Athlete year of birth"], axis = 1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[171], line 1
----> 1 df2= df2.drop(["Athlete club","Athlete country","Athlete age category","Athlete year of birth"],
axis = 1)

File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:5258, in DataFrame.drop(self, labels, axis, inde
x, columns, level, inplace, errors)
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (...)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123
   (...)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4549, in NDFrame.drop(self, labels, axis, inde
x, columns, level, inplace, errors)
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File ~\anaconda3\Lib\site-packages\pandas\core\generic.py:4591, in NDFrame._drop_axis(self, labels, axi
s, level, errors, only_slice)
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:6699, in Index.drop(self, labels, errors)
   6697 if mask.any():
   6698     if errors != "ignore":
-> 6699         raise KeyError(f"{list(labels[mask])} not found in axis")
   6700     indexer = indexer[~mask]
   6701 return self.delete(indexer)

KeyError: "['Athlete club', 'Athlete country', 'Athlete age category', 'Athlete year of birth'] not foun
d in axis"
```

In [90]:
```
df2.head()
```

Out[90]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	Age
2538571	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:34:19	M	10.627	53107	55.0
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50	M	10.409	8785	46.0
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40	M	9.962	4502	44.0
2538574	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:30:49	F	9.452	63964	51.0
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47	M	9.379	4485	56.0

In [91]:
```
df2.shape
```

Out[91]:
```
(63489, 10)
```

In [92]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 63489 entries, 2538571 to 2762408
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Year of event             63489 non-null  int64  
 1   Event dates               63489 non-null  object 
 2   Event name                63489 non-null  object 
 3   Event distance/length     63489 non-null  object 
 4   Event number of finishers  63489 non-null  int64  
 5   Athlete performance       63489 non-null  object 
 6   Athlete gender            63489 non-null  object 
 7   Athlete average speed     63489 non-null  object 
 8   Athlete ID                63489 non-null  int64  
 9   Age                       60025 non-null  float64
dtypes: float64(1), int64(3), object(6)
memory usage: 7.3+ MB
```

Remove Null Values

In [93]:
```
df2.isnull().sum()
```

Out[93]:
```
Year of event                  0
Event dates                    0
Event name                     0
Event distance/length          0
Event number of finishers      0
Athlete performance            0
Athlete gender                 0
Athlete average speed          0
Athlete ID                     0
Age                         3464
dtype: int64
```

Remove Duplicated

In [125]:
```
df2[df2.duplicated()== True] # No Dups
```

Out[125]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	Age

Fix dtypes

In [126]:
```
df2.dtypes
```

Out[126]:
```
Year of event                int64
Event dates                 object
Event name                  object
Event distance/length       object
Event number of finishers    int64
Athlete performance         object
Athlete gender              object
Athlete average speed       object
Athlete ID                   int64
Age                          int32
dtype: object
```

In [123]:
```
df2['Age'] = df2['Age'].replace([np.inf, -np.inf], np.nan)
df2['Age'] = df2['Age'].fillna(0)
df2['Age'] = df2['Age'].astype(int)
```

In [124]:
```
df2.dtypes
```

Out[124]:
```
Year of event                int64
Event dates                 object
Event name                  object
Event distance/length       object
Event number of finishers    int64
Athlete performance         object
Athlete gender              object
Athlete average speed       object
Athlete ID                   int64
Age                          int32
dtype: object
```

In [127]:
```
df2 = df2.rename(columns = { "Year of event":"Year",
                            "Event dates":"E_Date",
                            "Event name" : "E_name",
                            "Event distance/length": "E_Distance",
                            "Event number of finishers":"E_num_of_Finishers",
                            "Athlete performance":"Athlete_performance",
                            "Athlete gender":"Athlete_gender",
                            "Athlete average speed" : "Athlete_average_speed",
                            "Athlete ID":"Athlete_ID",
                            "Age":"Age"
                            })
```

In [128]:
```
df2
```

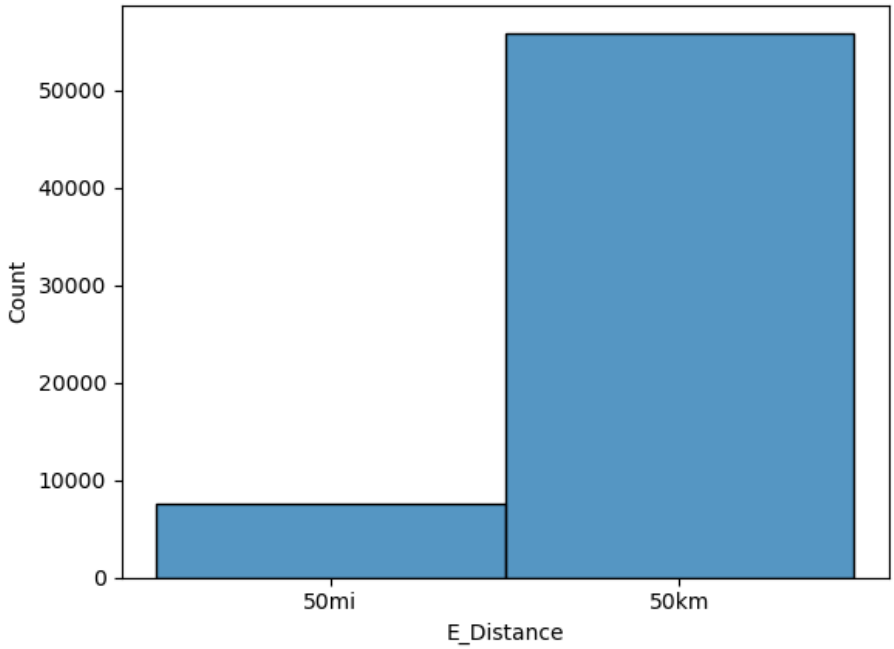
Out[128]:

	Year	E_Date	E_name	E_Distance	E_num_of_Finishers	Athlete_performance	Athlete_gender	Athlete_average_sp
2538571	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:34:19	M	10
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50	M	10
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40	M	9
2538574	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:30:49	F	9
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47	M	9
...	...	...	...	...	...	...	...	
2762404	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:25	F	6
2762405	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:27	F	6
2762406	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:44:18	F	6
2762407	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:04:50	M	6
2762408	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:11:43	M	6

63489 rows × 10 columns

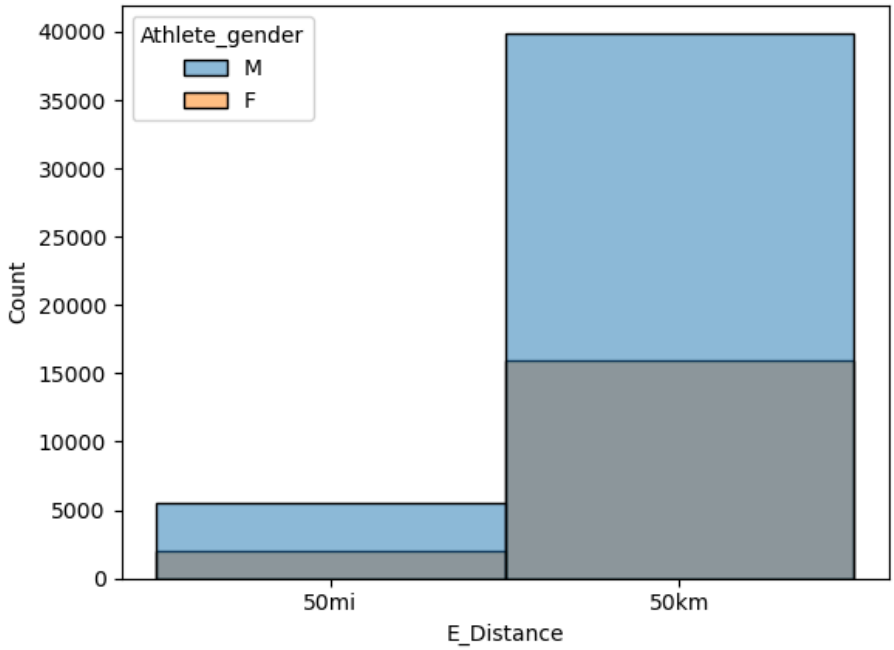
In [132]:
```
sns.histplot(df2["E_Distance"])
```

Out[132]: <Axes: xlabel='E_Distance', ylabel='Count'>



In [133]:
```
sns.histplot(df2,x="E_Distance",hue = "Athlete_gender")
```

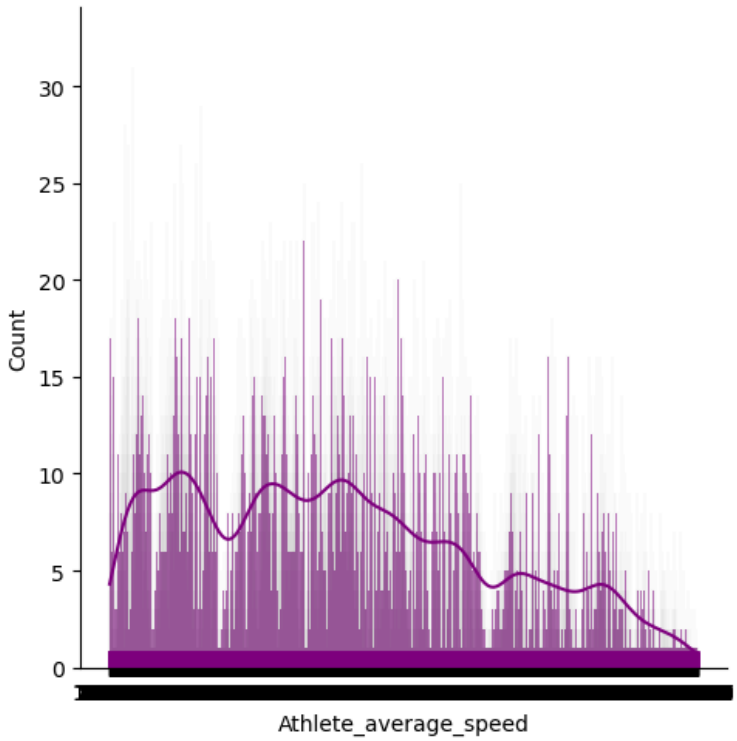
Out[133]: <Axes: xlabel='E_Distance', ylabel='Count'>



In [138]:
```
sns.displot(data=df2['Athlete_average_speed'], kde=True, color='purple', rug=True)
```

```
C:\Users\lahir\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has c
hanged to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[138]: <seaborn.axisgrid.FacetGrid at 0x1cb238b5250>

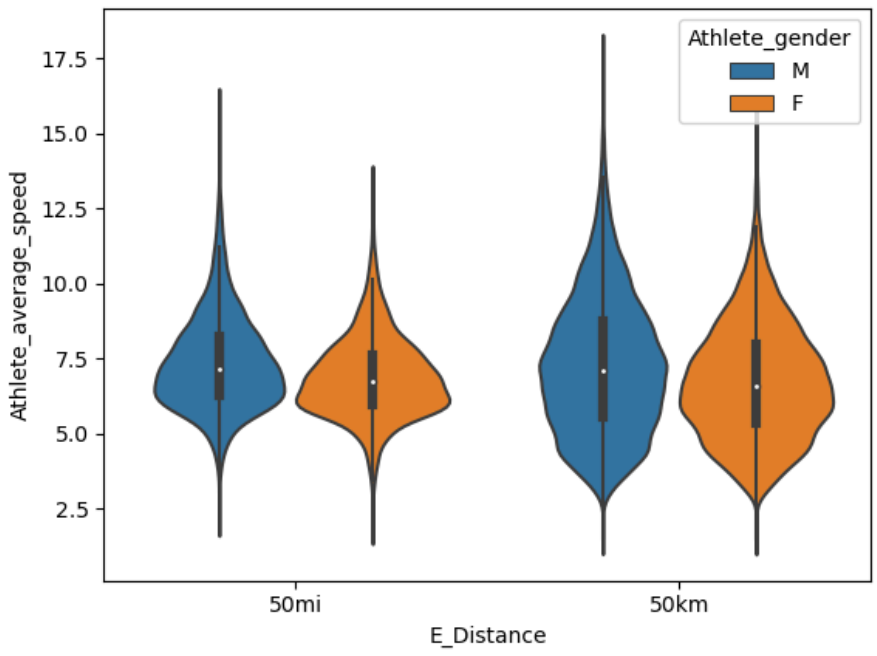


I have an error. because the AVG speed column is not in numeric type.

In [145]:
```
df2['Athlete_average_speed'] = pd.to_numeric(df2['Athlete_average_speed'])
```

In [147]:
```
sns.violinplot(data = df2, x = 'E_Distance',y = 'Athlete_average_speed', hue = 'Athlete_gender')
```

Out[147]: <Axes: xlabel='E_Distance', ylabel='Athlete_average_speed'>



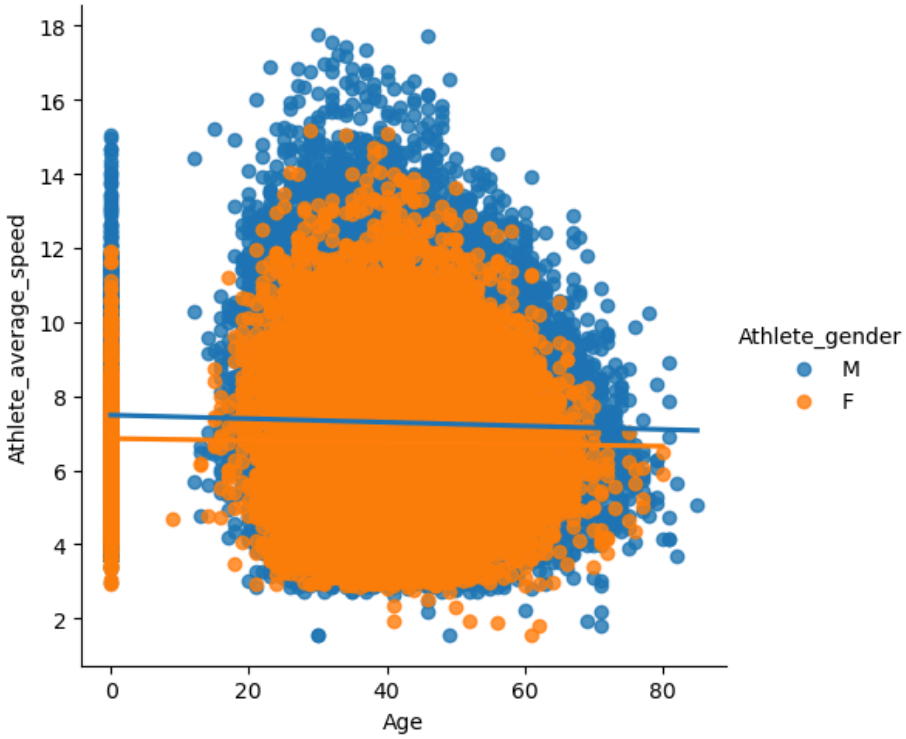
In [148]:
```
df2.dtypes
```

Out[148]:
```
Year                     int64
E_Date                  object
E_name                  object
E_Distance              object
E_num_of_Finishers       int64
Athlete_performance     object
Athlete_gender          object
Athlete_average_speed  float64
Athlete_ID               int64
Age                      int32
dtype: object
```

In [149]:
```
sns.lmplot(data = df2, x='Age', y = 'Athlete_average_speed', hue = 'Athlete_gender')
```

```
C:\Users\lahir\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has c
hanged to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[149]:
```
<seaborn.axisgrid.FacetGrid at 0x1cb37f49550>
```



Questions

In [150]:
```
# difference between male and female via Speed
```

In [152]:
```
df2.groupby(['E_Distance','Athlete_gender'])['Athlete_average_speed'].mean()
```

Out[152]:
```
E_Distance  Athlete_gender
50km        F                6.737871
            M                7.285058
50mi        F                6.882499
            M                7.367648
Name: Athlete_average_speed, dtype: float64
```

In [155]:
```
df2.groupby(['Age','E_Distance'])['Athlete_average_speed'].agg(['mean','sum'])
```

Out[155]:

		mean	sum
Age	**E_Distance**		
0	**50km**	6.985746	23541.964
	50mi	6.815468	640.654
9	**50km**	4.681000	4.681
12	**50km**	10.067000	20.134
	50mi	10.280000	10.280
...	**...**	...	...
79	**50mi**	5.969000	23.876
80	**50km**	5.502333	16.507
81	**50km**	6.120333	36.722
82	**50km**	4.670000	9.340
85	**50km**	5.068000	5.068

140 rows × 2 columns

In [166]:
```
df2.groupby(['Age'])['Athlete_average_speed'].agg(['mean','count']).sort_values('mean')
```

Out[166]:

	mean	count
Age		
82	4.670000	2
9	4.681000	1
85	5.068000	1
80	5.502333	3
77	5.859500	12
...	...	...
22	8.106074	258
20	8.132565	168
19	8.155851	87
15	8.293700	10
12	10.138000	3

74 rows × 2 columns

Thank You

By Lahiru Sadakelum