Seasonal and Gender-Based Analysis of Athlete Performance in Races

This notebook provides an in-depth analysis of athlete performance data across different seasons, genders, and race lengths. The objective is to uncover trends and insights that can assist race organizers, coaches, and athletes in understanding performance metrics.

Prepared by: ***Mugesh***

In [3]:
```
import pandas as pd
import plotly.express as px
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
df=pd.read_csv("/Users/mugesh/downloads/Marathon_dataset/TWO_CENTURIES_OF_UM_RACES.csv")
```

In [5]:
```
#The dataset has been successfully imported. Below is a preview of the data, showing key columns and initial rows f
```

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	At
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 [7]:
```
df.shape
```

Out[7]:
```
(7461195, 13)
```

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

Out[8]:
```
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 [9]:
```
#Filtering Data
```

In [10]:
```
# This step filters the dataset to display only the races where the event distance or length is exactly 50MI.
# The filtered data will provide insights specific to these races.
```

In [11]:
```
df[df['Event distance/length'] =='50mi']
```

Out[11]:

	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
55	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	9:53:05 h	*Middleville, MI	USA	1983.0	M	M23	
56	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:09:35 h	*Waterloo, ON	CAN	1977.0	F	W40	
57	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:33:00 h	*Kitchener, ON	CAN	1976.0	M	M40	
58	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:38:17 h	*Utica, MI	USA	1986.0	M	M23	
59	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:56:35 h	*Grass Lake, MI	USA	1988.0	M	M23	
...	...	...	...	...	...	...	...	...	...	...	...	...
7461181	1995	07.01.1995	Avalon Benefit	50mi	92	11:59:37 h	NaN	USA	1941.0	M	M50	

	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
			50-Mile Run (USA)									
7461182	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:01:41 h	NaN	USA	1932.0	M	M60	
7461183	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:03:26 h	NaN	USA	1934.0	F	W60	
7461184	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:03:26 h	NaN	USA	1951.0	F	W40	
7461185	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:05:59 h	NaN	USA	1947.0	F	W45	

352181 rows × 13 columns

In [12]:
```
# In this step, the dataset is filtered to include races with distances of either 50KM or 50MI and ensure that the
# This is achieved using the isin() method for distances and a condition for the year.
```

In [13]:
```
df[(df['Event distance/length'].isin(['50km','50mi']))  & (df['Year of event']==2020)]
```

Out[13]:

	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	Athl a categ
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	M
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50 h	NaN	AUS	1974.0	M	M
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40 h	NaN	TPE	1976.0	M	M
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	W
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47 h	NaN	TPE	1964.0	M	M
...	...	...	...	...	...	...	...	...	...	...	
2762404	2020	03.10.2020	Bison Ultra-	50km	271	7:36:25 h	AKS Polonia Warszawa	POL	1981.0	F	W

	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	Athl a categ
			Trail 50 (POL)								
2762405	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:27 h	*Warszawa	POL	1970.0	F	W
2762406	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:44:18 h	Outdoor Training	POL	1993.0	F	W
2762407	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:04:50 h	PH Bysewo Gdańsk	POL	1976.0	M	M
2762408	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:11:43 h	*Nowe Aleksandrowo	POL	1961.0	M	M

63489 rows × 13 columns

In [14]:
```
# This step isolates the dataset to show details of the event “Everglades 50 Mile Ultra Run (USA)”.
# This allows for a focused analysis of this particular race.
```

In [15]:
```
df[df['Event name']=='Everglades 50 Mile Ultra Run (USA)']
```

Out[15]:

	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
51923	2018	17.02.2018	Everglades 50 Mile Ultra Run (USA)	50mi	23	7:00:00 h	*Tadworth	GBR	1980.0	M	M35
51924	2018	17.02.2018	Everglades 50 Mile Ultra Run (USA)	50mi	23	8:29:48 h	*Fort Lauderdale, FL	USA	1989.0	M	M23
51925	2018	17.02.2018	Everglades 50 Mile Ultra Run (USA)	50mi	23	8:45:31 h	*Miami, FL	USA	1979.0	M	M35
51926	2018	17.02.2018	Everglades 50 Mile Ultra Run (USA)	50mi	23	9:01:52 h	*Fort Lauderdale, FL	USA	1967.0	M	M50
51927	2018	17.02.2018	Everglades 50 Mile Ultra Run (USA)	50mi	23	9:26:06 h	*Naples, FL	USA	1986.0	M	M23
...	...	...	...	...	...	...	...	...	...	...	...
6417091	2015	21.02.2015	Everglades 50 Mile Ultra Run (USA)	50mi	67	13:24:08 h	*Saarbrücken	GER	1968.0	M	M45
6417092	2015	21.02.2015	Everglades 50 Mile Ultra Run (USA)	50mi	67	13:24:08 h	*Saarbrücken	GER	1974.0	F	W40
6417093	2015	21.02.2015	Everglades 50 Mile	50mi	67	13:40:57 h	*Weston, FL	USA	1972.0	F	W40

	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
			Ultra Run (USA)								
6417094	2015	21.02.2015	Everglades 50 Mile Ultra Run (USA)	50mi	67	13:55:09 h	*Merritt Island, FL	USA	1960.0	M	M50
6417095	2015	21.02.2015	Everglades 50 Mile Ultra Run (USA)	50mi	67	13:55:14 h	*Miami, FL	USA	1983.0	M	M23

338 rows × 13 columns

In [16]:
```
# Extract Location from Event Name
# This step extracts the location of the event (e.g., USA) from the event name “Everglades 50 Mile Ultra Run (USA)”
# Using string manipulation, the portion within parentheses is isolated for further analysis.
```

In [17]:
```
df[df['Event name'] == 'Everglades 50 Mile Ultra Run (USA)']['Event name'].str.split('(').str.get(1).str.split(')')
```

Out[17]:
```
51923      USA
51924      USA
51925      USA
51926      USA
51927      USA
          ...
6417091    USA
6417092    USA
6417093    USA
6417094    USA
6417095    USA
Name: Event name, Length: 338, dtype: object
```

In [18]:
```
# This step filters the dataset to include only events where the location is specified as USA in the event name.
```

In [19]:
```
df[df['Event name'].str.split('(').str.get(1).str.split(')').str.get(0) == 'USA']
```

Out[19]:

	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
55	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	9:53:05 h	*Middleville, MI	USA	1983.0	M	M23	
56	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:09:35 h	*Waterloo, ON	CAN	1977.0	F	W40	
57	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:33:00 h	*Kitchener, ON	CAN	1976.0	M	M40	
58	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:38:17 h	*Utica, MI	USA	1986.0	M	M23	
59	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:56:35 h	*Grass Lake, MI	USA	1988.0	M	M23	
...	...	...	...	...	...	...	...	...	...	...	...	...
7461181	1995	07.01.1995	Avalon Benefit	50mi	92	11:59:37 h	NaN	USA	1941.0	M	M50	

	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
			50-Mile Run (USA)									
7461182	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:01:41 h	NaN	USA	1932.0	M	M60	
7461183	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:03:26 h	NaN	USA	1934.0	F	W60	
7461184	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:03:26 h	NaN	USA	1951.0	F	W40	
7461185	1995	07.01.1995	Avalon Benefit 50-Mile Run (USA)	50mi	92	12:05:59 h	NaN	USA	1947.0	F	W45	

1398540 rows × 13 columns

In [20]:
```
# Combine All Filters
# This step combines all the filters to narrow down the dataset to events that meet the following criteria:
#       •       Location: USA
#       •       Distance: 50KM or 50MI
#       •       Year: 2020
```

In [21]:
```
df[
(df['Event distance/length'].isin(['50km','50mi']))
&
(df['Year of event']==2020)
&
(df['Event name'].str.split('(').str.get(1).str.split(')').str.get(0) == 'USA')]
```

Out[21]:

	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 av s
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	3:17:55 h	*Normandy Park, WA	USA	1991.0	M	M23	1
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:02:32 h	*Gold Bar, WA	USA	1981.0	M	M35	1
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:07:57 h	*Vashon, WA	USA	1999.0	M	MU23	1
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:22:02 h	*Gig Harbor, WA	USA	1983.0	M	M35	1

	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 av s
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:27:34 h	*Bainbridge Island, WA	USA	1977.0	M	M40	
...	...	...	...	...	...	...	...	...	...	...	...	
2760957	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:07:48 h	*East Lansing, MI	USA	1958.0	F	W60	
2760958	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:27:22 h	*Traverse City, MI	USA	1977.0	F	W40	
2760959	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:27:24 h	*Traverse City, MI	USA	1962.0	F	W55	
2760960	2020	03.10.2020	Yankee Springs Fall Trail Run	50km	30	7:38:30 h	*Mason, MI	USA	1981.0	F	W35	

	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 av s
			Festival (USA)									
2760961	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:59:53 h	NaN	USA	1980.0	M	M35	

26090 rows × 13 columns

In [22]:
```
df2 = df[
(df['Event distance/length'].isin(['50km','50mi']))
&
(df['Year of event']==2020)
&
(df['Event name'].str.split('(').str.get(1).str.split(')').str.get(0) == 'USA')]
```

In [23]:
```
df2.shape
```

Out[23]:
```
(26090, 13)
```

In [24]:
```
# Remove (USA) from Event Names
# This step cleans the event names by removing (USA) from them to make the dataset more consistent and readable.
```

In [25]:
```
df2['Event name'].str.split('(').str.get(0)
```

Out[25]:
```
2539945    West Seattle Beach Run - Winter Edition
2539946    West Seattle Beach Run - Winter Edition
2539947    West Seattle Beach Run - Winter Edition
2539948    West Seattle Beach Run - Winter Edition
2539949    West Seattle Beach Run - Winter Edition
                          ...
2760957     Yankee Springs Fall Trail Run Festival
2760958     Yankee Springs Fall Trail Run Festival
2760959     Yankee Springs Fall Trail Run Festival
2760960     Yankee Springs Fall Trail Run Festival
2760961     Yankee Springs Fall Trail Run Festival
Name: Event name, Length: 26090, dtype: object
```

In [26]:
```
# Extract Clean Event Names
# This step extracts the clean event names by removing any text within parentheses, such as (USA), leaving only the
```

In [ ]:
```
df2['Event name'] = df2['Event name'].str.split('(').str.get(0)
```

In [28]:
```
df2.head()
```

Out[28]:

	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	At ave s
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55 h	*Normandy Park, WA	USA	1991.0	M	M23	1
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32 h	*Gold Bar, WA	USA	1981.0	M	M35	1
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57 h	*Vashon, WA	USA	1999.0	M	MU23	1
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02 h	*Gig Harbor, WA	USA	1983.0	M	M35	1
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34 h	*Bainbridge Island, WA	USA	1977.0	M	M40	1

In [29]:
```
# Calculate Athlete Age
# This step calculates the athlete_age by subtracting the athlete's year of birth from 2020,providing a consistent
```

In [ ]:
```
df2['athlete age'] = 2020 - df2['Athlete year of birth']
```

In [31]:
```
# Remove Hours (h) from Athlete Performance
# This step extracts the numerical part of the Athlete performance column by removing the h (hours),
#ensuring the data is in a clean, usable format for analysis.
```

In [32]:
```
df2['Athlete performance'].str.split(' ').str.get(0)
```

Out[32]:
```
2539945    3:17:55
2539946    4:02:32
2539947    4:07:57
2539948    4:22:02
2539949    4:27:34
            ...
2760957    7:07:48
2760958    7:27:22
2760959    7:27:24
2760960    7:38:30
2760961    7:59:53
Name: Athlete performance, Length: 26090, dtype: object
```

In [ ]:
```
df2['Athlete performance']=df2['Athlete performance'].str.split(' ').str.get(0)
```

In [34]:
```
df2.head()
```

Out[34]:

	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	At ave s
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	*Normandy Park, WA	USA	1991.0	M	M23	1
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	*Gold Bar, WA	USA	1981.0	M	M35	1
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	*Vashon, WA	USA	1999.0	M	MU23	1
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	*Gig Harbor, WA	USA	1983.0	M	M35	1
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	*Bainbridge Island, WA	USA	1977.0	M	M40	1

In [35]:
```
# Drop noisy Columns
# This step removes redundant columns (Athlete club,
```

```
#                                     Athlete country,
#                                     Athlete year of birth,
#                                     Athlete age category)
# to streamline the dataset and focus on relevant fields for the analysis.
```

In [36]:
```
df2 = df2.drop(['Athlete club',
                'Athlete country',
                'Athlete year of birth',
                'Athlete age category'],
               axis =1
              )
```

In [37]:
```
df2.head()
```

Out[37]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	athlete age
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	M	15.158	71287	29.0
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	M	12.369	629508	39.0
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	M	12.099	64838	21.0
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	M	11.449	704450	37.0
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	M	11.212	810281	43.0

In [38]:
```
#clean up null values
```

In [39]:
```
df2.isna().sum()
```

Out[39]:
```
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
athlete age                 233
dtype: int64
```

In [40]:
```
df2[df2['athlete age'].isna()==1]
```

Out[40]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	athlete age
2547794	2020	25.01.2020	North Carolina Fat Ass 50 Km	50km	57	6:10:30	M	8.097	811923	NaN
2551331	2020	19.01.2020	Big Bend 50 - Fresno Creek 50K	50km	54	4:46:34	M	10.469	812656	NaN
2551336	2020	19.01.2020	Big Bend 50 - Fresno Creek 50K	50km	54	5:08:36	M	9.721	812657	NaN
2551344	2020	19.01.2020	Big Bend 50 - Fresno Creek 50K	50km	54	5:54:04	F	8.473	658221	NaN
2551348	2020	19.01.2020	Big Bend 50 - Fresno Creek 50K	50km	54	6:07:11	M	8.17	812660	NaN
...	...	...	...	...	...	...	...	...	...	...
2746543	2020	17.10.2020	Black River Trail Classic 50 Km	50km	8	8:31:26	F	5.866	857251	NaN
2749869	2020	17.10.2020	MuleSkinner Endurance 50 Mile Race	50mi	27	11:55:05	M	6.752	857957	NaN
2755985	2020	10.10.2020	Man Against Horse 50 Mile Race	50mi	23	9:03:25	M	8.885	859462	NaN
2755994	2020	10.10.2020	Man Against Horse 50 Mile Race	50mi	23	10:37:00	M	7.579	398583	NaN
2755997	2020	10.10.2020	Man Against Horse 50 Mile Race	50mi	23	12:30:00	M	6.437	859465	NaN

233 rows × 10 columns

In [41]:
```
df2 = df2.dropna()
```

In [42]:
```
df2.shape
```

Out[42]: (25857, 10)

In [43]:
```
#checking for duplicates
```

In [44]:
```
df2[df2.duplicated()== True]
```

Out[44]:

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

In [45]:
```
#reset index
```

In [46]:
```
df2.reset_index(drop = True)
```

Out[46]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	athlete age
0	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	M	15.158	71287	29.0
1	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	M	12.369	629508	39.0
2	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	M	12.099	64838	21.0
3	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	M	11.449	704450	37.0
4	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	M	11.212	810281	43.0
...	...	...	...	...	...	...	...	...	...	...
25852	2020	03.10.2020	Yankee Springs Fall Trail Run Festival	50km	30	7:07:48	F	7.013	816361	62.0
25853	2020	03.10.2020	Yankee Springs Fall Trail Run Festival	50km	30	7:27:22	F	6.706	326469	43.0
25854	2020	03.10.2020	Yankee Springs Fall	50km	30	7:27:24	F	6.705	372174	58.0

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	athlete age
			Trail Run Festival							
25855	2020	03.10.2020	Yankee Springs Fall Trail Run Festival	50km	30	7:38:30	F	6.543	860349	39.0
25856	2020	03.10.2020	Yankee Springs Fall Trail Run Festival	50km	30	7:59:53	M	6.252	770097	40.0

25857 rows × 10 columns

In [47]:
```
# Convert Data Types for Consistency
# This step ensures data type consistency by:
#       •       Converting athlete age to an integer for accurate numerical analysis.
#       •       Converting Athlete average speed to a float for precise calculations.
```

In [48]:
```
df2.dtypes
```

Out[48]:
```
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
athlete age                float64
dtype: object
```

In [49]:
```
df2['athlete age']= df2['athlete age'].astype(int)
```

In [50]:
```
df2['Athlete average speed']= df2['Athlete average speed'].astype(float)
```

In [51]:
```
df2.dtypes
```

Out[51]:
```
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       float64
Athlete ID                    int64
athlete age                   int64
dtype: object
```

In [52]:
```
df2.head()
```

Out[52]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete average speed	Athlete ID	athlete age
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	M	15.158	71287	29
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	M	12.369	629508	39
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	M	12.099	64838	21
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	M	11.449	704450	37
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	M	11.212	810281	43

In [53]:
```
# Rename Columns for Clarity
# This step renames the columns to make them more descriptive, consistent, and easier to interpret during analysis.
```

In [54]:
```
# 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        float64
# Athlete ID                     int64
# athlete age                    int64
# dtype: object
```

In [55]:
```
df2=df2.rename(columns = {'Year of event ':'year',
                          'Event dates':'race_day',
                          'Event name':'race_name',
                          'Event distance/length':'race_length',
                          'Event number of finishers':'race_number_of_finishers',
                          'Athlete performance':'athlete_performance',
                          'Athlete gender':'athlete_gender',
                          'Athlete average speed':'athlete_avg_speed',
                          'Athlete ID':'athlete_id',
                          'athlete age':'athlete_age'
})
```

In [56]:
```
df2.columns
```

Out[56]:
```
Index(['Year of event', 'race_day', 'race_name', 'race_length',
       'race_number_of_finishers', 'athlete_performance', 'athlete_gender',
       'athlete_avg_speed', 'athlete_id', 'athlete_age'],
      dtype='object')
```

In [57]:
```
# Rearrange Column Order
# This step rearranges the columns in a logical order to improve readability and make the dataset more intuitive fo
```

In [58]:
```
df3 =df3=df2[['race_day','race_name','race_length','Year of event','race_number_of_finishers','athlete_id',
       'athlete_gender','athlete_age','athlete_performance',
      'athlete_avg_speed']]
```

In [59]:
```
df3.head(10)
```

Out[59]:

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
2539945	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	71287	M	29	
2539946	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	629508	M	39	
2539947	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	64838	M	21	
2539948	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	704450	M	37	
2539949	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810281	M	43	
2539950	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810282	F	35	
2539951	02.02.2020	West Seattle Beach Run	50km	2020	20	11739	M	59	

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
		- Winter Edition							
2539952	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	80394	M	50	
2539953	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	140909	F	45	
2539954	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	753889	M	41	

In [60]:
```
# Which Races Did I Run in 2020?
# This step answers the question by filtering the dataset to find the two races—Sarasota
#                           and Everglades—that I participated in during the year 2020.
```

In [61]:
```
df3[df3['athlete_id'].duplicated(keep=False)].sort_values(by='athlete_id')
```

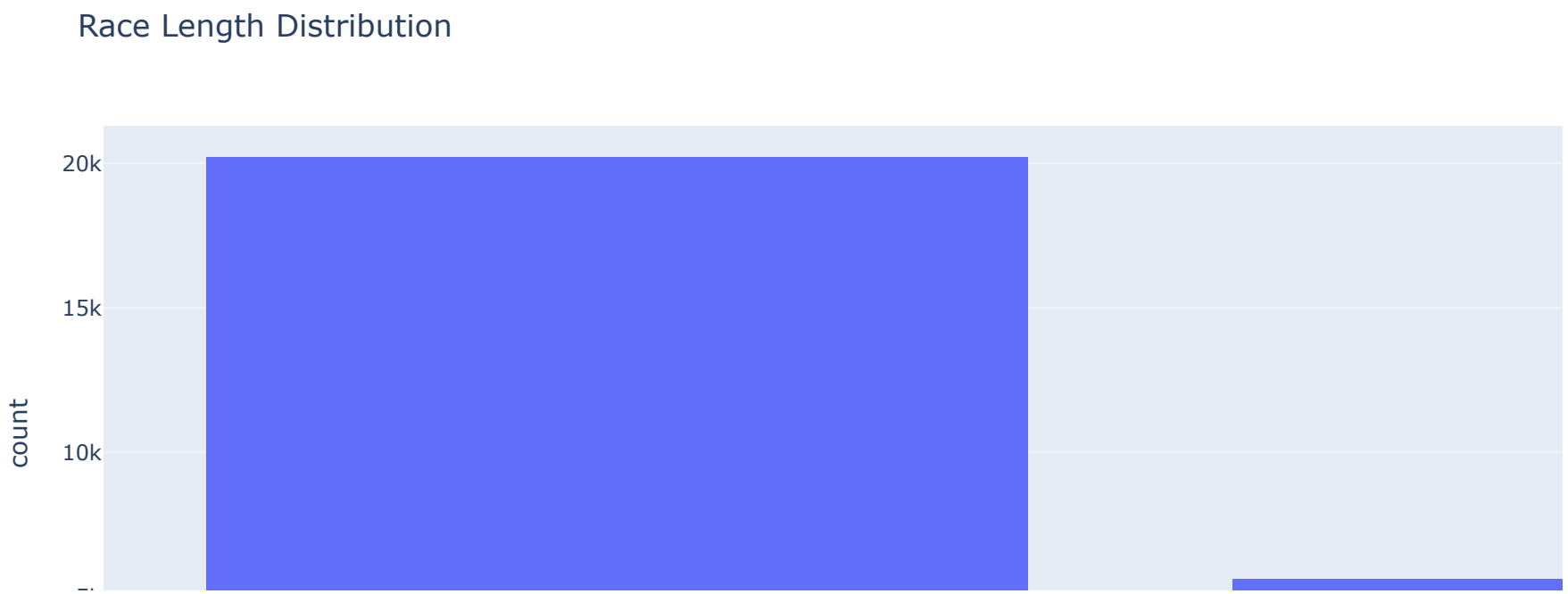
Out[61]:

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athle
2555844	18.01.2020	Capitol Peak MegaFatAss	50km	2020	85	55	M	37	
2612803	29.02.2020	Fragrance Lake 50km Race	50km	2020	53	55	M	37	
2733216	25.10.2020	Cougar Mountain Trail Run	50km	2020	25	55	M	37	
2572693	25.07.2020	Bunk House Trail Runs	50km	2020	23	58	M	34	
2563950	11.01.2020	Frozen Gnome 50K	50km	2020	100	58	M	34	
...	...	...	...	...	...	...	...	...	
2728055	07.11.2020	Barrier Island 50 Mile Ultra Race	50mi	2020	17	853139	M	20	
2728067	07.11.2020	Barrier Island 50 Mile Ultra Race	50mi	2020	17	853142	M	47	
2747823	17.10.2020	Oktoberfest Trail Run Festival	50km	2020	36	853142	M	47	
2755643	10.10.2020	The Remix 50 km Race	50km	2020	47	853149	F	34	
2728111	07.-08.11.2020	Jalapeno Hundred 50 km Race	50km	2020	28	853149	F	34	

8251 rows × 10 columns

In [62]:
```
#Race Length Distribution
# This step uses a histogram to visualize the distribution of race lengths, providing insights into the frequency o
# The chart highlights how races are grouped across various lengths.
```

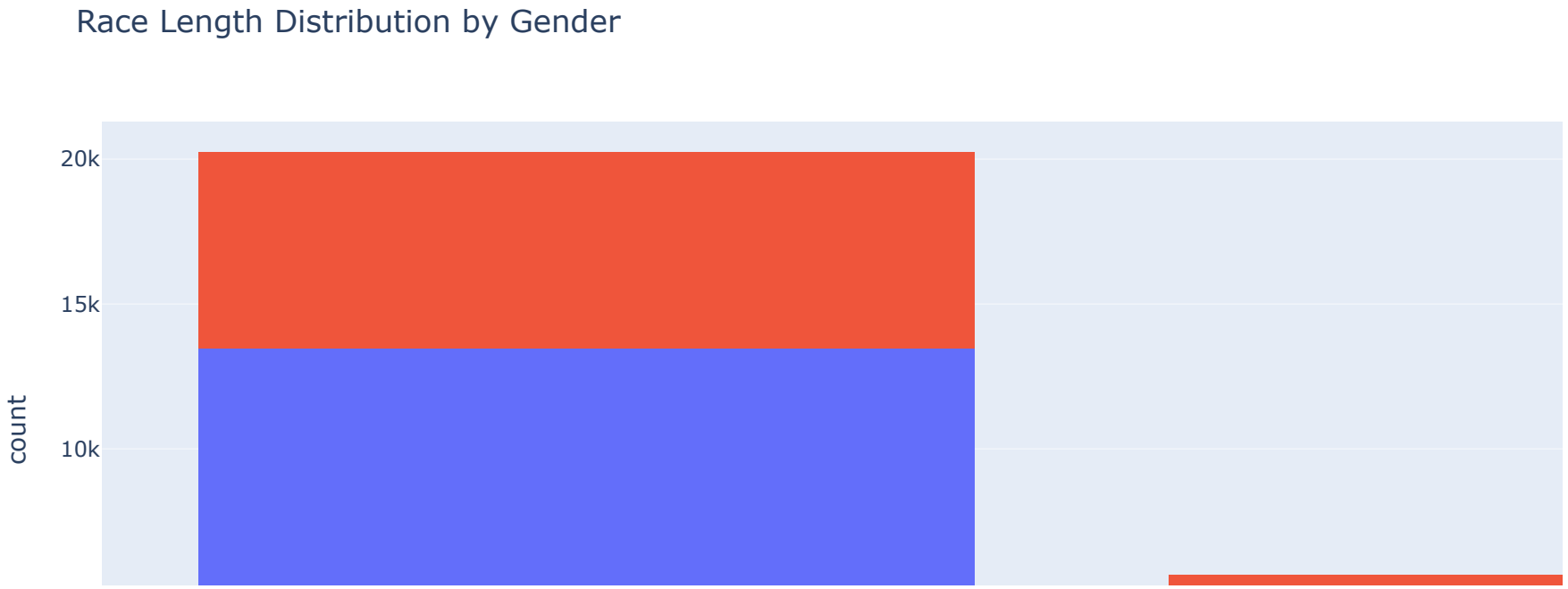
In [63]:
```
px.histogram(df3, x='race_length', nbins=10, title="Race Length Distribution").show()
```



In [64]:
```
# Race Length Distribution by Gender
# This histogram visualizes the distribution of race lengths, differentiated by athlete gender.
```

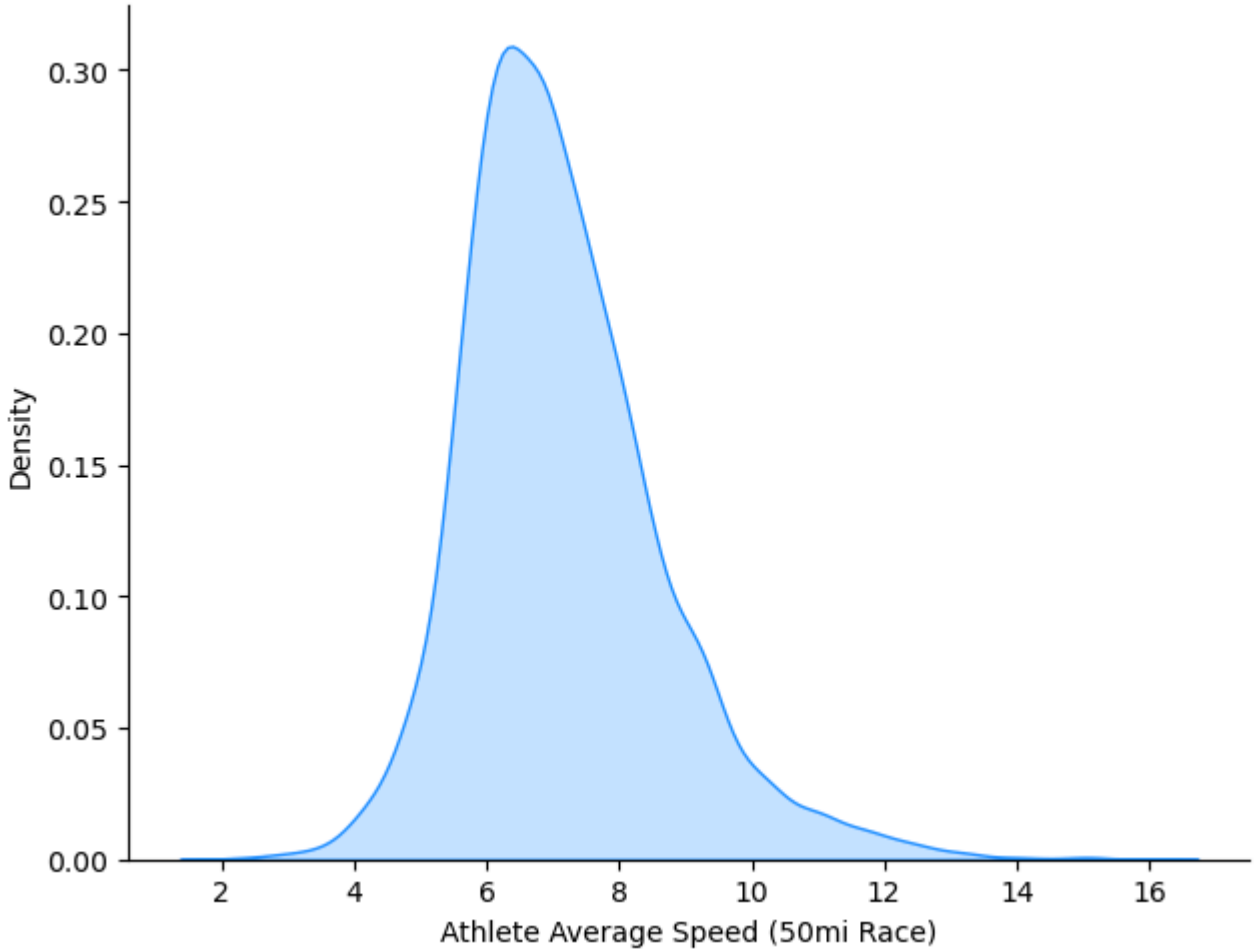
In [65]:
```
fig = px.histogram(
    df3,
    x='race_length',
    color='athlete_gender',  # Differentiate by gender
    nbins=10,
    title="Race Length Distribution by Gender"
)

fig.show()
```



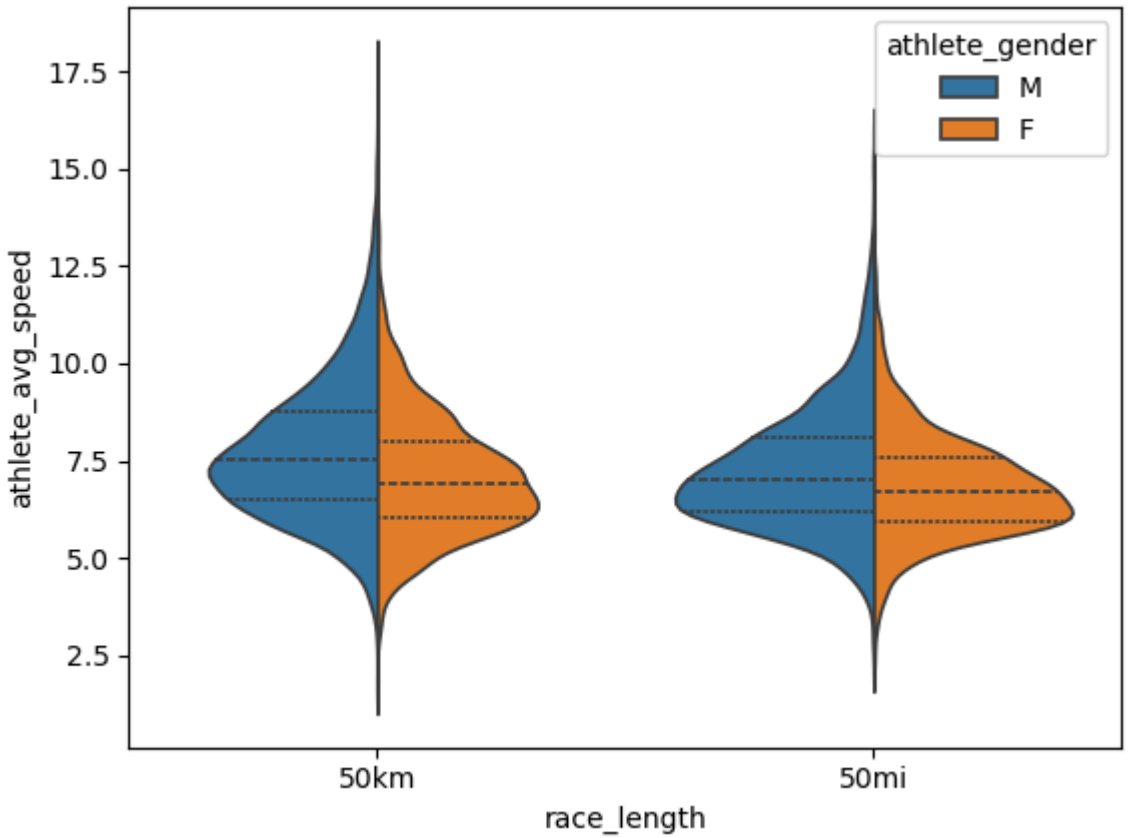
In [66]:
```
# Athlete Average Speed Distribution (50mi Race)
# This density plot illustrates the distribution of athlete average speeds specifically for 50-mile races,
# offering a smooth curve to highlight performance trends within this distance category.
```

In [67]:
```
sns.displot(df3[df3['race_length'] == '50mi'],
            x='athlete_avg_speed', kind="kde", fill=True, color="dodgerblue",
            height=5, aspect=1.3).set_axis_labels("Athlete Average Speed (50mi Race)", "Density")
plt.show()
```



In [68]:
```
# Athlete Speed by Race Length and Gender
# This violin plot shows the distribution of athlete average speeds across different race lengths, split by gender.
# Quartile lines are included to provide additional statistical insights.
```

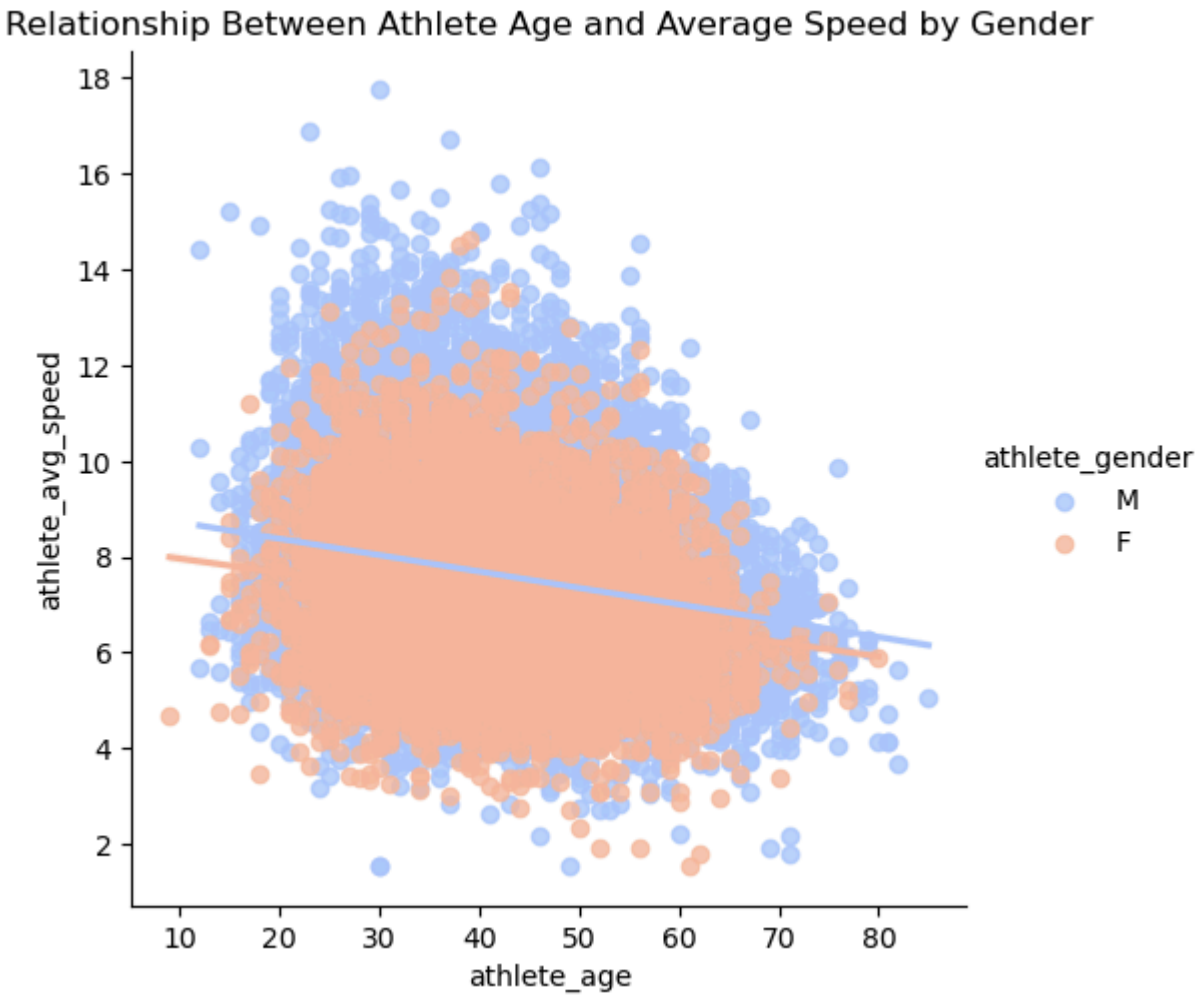
In [69]:
```
sns.violinplot(data=df3, x='race_length', y='athlete_avg_speed', hue='athlete_gender', split=True, inner='quart')
plt.show()
```



In [70]:
```
# Relationship Between Athlete Age and Average Speed by Gender
# This scatter plot with regression lines illustrates the relationship between athlete age and average speed, diffe
# The trend lines provide insights into how performance varies across different age groups for males and females.
```

In [71]:
```
g = sns.lmplot(
    data=df3,
    x='athlete_age',
    y='athlete_avg_speed',
    hue='athlete_gender',
    palette='coolwarm'
)
g.set(title="Relationship Between Athlete Age and Average Speed by Gender")
```

```
plt.show()
```



In [72]:
```
# Difference in Speed for 50KM and 50MI: Male vs. Female
# This step analyzes the average speed difference between male and female athletes for 50KM and 50MI races,
# providing insights into gender-based performance variations.
```

In [73]:
```
df3.groupby(['race_length','athlete_gender'])['athlete_avg_speed'].mean()
```

Out[73]:
```
race_length  athlete_gender
50km         F                 7.083011
             M                 7.738985
50mi         F                 6.834371
             M                 7.257633
Name: athlete_avg_speed, dtype: float64
```

In [74]:
```
# 50MI Race: Athlete Speed by Age Group (Count > 19)
# This step analyzes athlete performance for 50-mile races, grouping by age and filtering for age groups with more
# It calculates the mean speed and total count for each age group, sorting them by average speed.
```

In [75]:
```
df3.query('race_length == "50mi" ').groupby('athlete_age')['athlete_avg_speed'].agg(['mean','count']).sort_values('
```

Out[75]:

athlete_age	mean	count
29	7.902252	135
23	7.779800	55
28	7.575252	107
30	7.569204	157
25	7.540923	91
31	7.451638	138
38	7.430022	231
35	7.422359	195
36	7.403854	185
26	7.379800	75
33	7.379188	149
22	7.367902	41
24	7.354274	73
42	7.327656	209
34	7.319011	182
40	7.310669	236
21	7.304658	38
20	7.293778	27
32	7.287969	162
39	7.283648	227
37	7.220493	221
27	7.201748	119

athlete_age	mean	count
41	7.197017	179
45	7.130667	156
47	7.085410	156
44	7.052642	179
51	7.019379	124
46	6.897236	174
43	6.884225	187
52	6.847400	130
55	6.828623	69
54	6.805506	83
49	6.789553	159
57	6.783205	78
53	6.736969	96
56	6.708373	67
48	6.696853	136
59	6.672072	83
50	6.671541	172
64	6.620727	22
58	6.582328	67
63	6.514806	31
61	6.358355	31
62	6.272730	37

	mean	count
athlete_age		
60	6.261788	33

In [76]:
```
# 50MI Race: Athlete Speed by Age Group (Count > 10)
# This step evaluates athlete performance in 50-mile races by grouping data based on age. It calculates the mean sp
# sorts them by average speed in ascending order.
```

In [77]:
```
df3.query('race_length == "50mi" ').groupby('athlete_age')['athlete_avg_speed'].agg(['mean','count']).sort_values('
```

Out[77]:

athlete_age	mean	count
70	5.470667	12
65	5.934786	14
67	6.114909	11
60	6.261788	33
62	6.272730	37
61	6.358355	31
63	6.514806	31
58	6.582328	67
64	6.620727	22
50	6.671541	172
59	6.672072	83
48	6.696853	136
56	6.708373	67
53	6.736969	96
57	6.783205	78
49	6.789553	159
54	6.805506	83
55	6.828623	69
52	6.847400	130
43	6.884225	187
46	6.897236	174
51	7.019379	124

athlete_age	mean	count
44	7.052642	179
47	7.085410	156
45	7.130667	156
41	7.197017	179
27	7.201748	119
37	7.220493	221
39	7.283648	227
32	7.287969	162
20	7.293778	27
21	7.304658	38
40	7.310669	236
34	7.319011	182
42	7.327656	209
24	7.354274	73
22	7.367902	41
33	7.379188	149
26	7.379800	75
36	7.403854	185
35	7.422359	195
38	7.430022	231
31	7.451638	138
19	7.506133	15

athlete_age	mean	count
25	7.540923	91
30	7.569204	157
28	7.575252	107
23	7.779800	55
29	7.902252	135

In [78]:
```
# Seasonal Analysis: Is It Slower in Summer Compared to Winter?
# This step examines the dataset by categorizing races into seasons:
#       •       Spring: March (3) to May (5)
#       •       Summer: June (6) to August (8)
#       •       Fall: September (9) to November (11)
#       •       Winter: December (12) to February (2)

# The analysis compares the average speeds of athletes across seasons to determine if performance is slower in summ
```

In [ ]:
```
df3['race_month']= df3['race_day'].str.split('.').str.get(1).astype(int)
```

In [ ]:
```
df3['race_season']=df3['race_month'].apply(lambda x:'winter'if x>11 else'Fall' if x>8 else 'summer' if x>5 else 'sp
```

In [81]:
```
df3.head(25)
```

Out[81]:

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
2539945	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	71287	M	29	
2539946	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	629508	M	39	
2539947	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	64838	M	21	
2539948	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	704450	M	37	
2539949	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810281	M	43	
2539950	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810282	F	35	
2539951	02.02.2020	West Seattle Beach Run	50km	2020	20	11739	M	59	

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
		- Winter Edition							
2539952	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	80394	M	50	
2539953	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	140909	F	45	
2539954	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	753889	M	41	
2539955	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810283	F	23	
2539956	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	810284	F	55	
2539957	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	40726	M	25	
2539958	02.02.2020	West Seattle	50km	2020	20	708195	F	45	

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
		Beach Run - Winter Edition							
2539959	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	73327	F	52	
2539960	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	48473	F	58	
2539961	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	150288	M	46	
2539962	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	53463	M	44	
2539963	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	11777	F	61	
2539964	02.02.2020	West Seattle Beach Run - Winter Edition	50km	2020	20	11780	M	60	

	race_day	race_name	race_length	Year of event	race_number_of_finishers	athlete_id	athlete_gender	athlete_age	athlete_
2541271	01.02.2020	White Rock Classic 50K	50km	2020	63	364678	M	54	
2541272	01.02.2020	White Rock Classic 50K	50km	2020	63	347890	M	44	
2541273	01.02.2020	White Rock Classic 50K	50km	2020	63	48096	M	27	
2541274	01.02.2020	White Rock Classic 50K	50km	2020	63	54241	F	47	
2541275	01.02.2020	White Rock Classic 50K	50km	2020	63	67094	M	35	

In [82]:
```
# Seasonal Athlete Performance Analysis
# This step groups the data by race season and calculates the mean and count of athlete average speeds for each sea
# The results are sorted by the average speed in descending order,
# allowing us to determine which season has the highest average athlete speed.
```

In [83]:
```
df3.groupby('race_season')['athlete_avg_speed'].agg(['mean','count']).sort_values('mean',ascending = False)
```

Out[83]:

race_season	mean	count
spring	7.684430	3294
winter	7.518187	11595
Fall	7.406619	8315
summer	6.869336	2653

In [84]:
```
# 50-Mile Race Performance by Season
# This step filters the data to include only 50-mile races, then groups it by race season to calculate the mean and
# The results are sorted by the average speed in descending order, revealing the seasonal trends in athlete perform
```

In [85]:
```
df3.query('race_length =="50mi"').groupby('race_season')['athlete_avg_speed'].agg(['mean','count']).sort_values('me
```

Out[85]:

race_season	mean	count
Fall	7.511585	1997
spring	7.082557	836
winter	7.048442	1977
summer	6.505776	817

END