In [104…

```
import pandas as pd
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [105…

```
df=pd.read_csv("TWO_CENTURIES_OF_UM_RACES.csv")
```

In [106…

```
df.head(5)
```

Out[106]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete club	Athle count
0	2018	06.01.2018	Selva Costera (CHI)	50km	22	4:51:39 h	Tnfrc	C
1	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:15:45 h	Roberto Echeverría	C
2	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:16:44 h	Puro Trail Osorno	C
3	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:34:13 h	Columbia	AF
4	2018	06.01.2018	Selva Costera (CHI)	50km	22	5:54:14 h	Baguales Trail	C

In [107…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7117634 entries, 0 to 7117633
Data columns (total 13 columns):
 #   Column                      Dtype  
---  ------                      -----  
 0   Year of event               int64  
 1   Event dates                 object 
 2   Event name                  object 
 3   Event distance/length       object 
 4   Event number of finishers   int64  
 5   Athlete performance         object 
 6   Athlete club                object 
 7   Athlete country             object 
 8   Athlete year of birth       float64
 9   Athlete gender              object 
 10  Athlete age category        object 
 11  Athlete average speed       object 
 12  Athlete ID                  int64  
dtypes: float64(1), int64(3), object(9)
memory usage: 705.9+ MB
```

In [108…

```
df.shape
```

Out[108]:

```
(7117634, 13)
```

In [109…

```
#Investigating how many different distance in UM in dataset
```

In [110…

```
df["Event distance/length"].value_counts()
```

Out[110]:

```
Event distance/length
50km                  1503475
100km                  883268
50mi                   333685
56km                   274234
24h                    172811
                        ...   
504km/7Etappen               1
303mi                        1
186mi                        1
101miles                     1
137.5km/3Etappen             1
Name: count, Length: 2131, dtype: int64
```

In [111…

```
#We will work on only the distance 50km and 50mil in 2020
#Now check the distance 50km and 50mil in 2020
```

In [112…

```
df[(df["Event distance/length"].isin(["50km","50mi"])) & (df["Year of eve
```

Out[112]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Ath
			Taipei 48hr Ultra				

2538571	2020	07.-09.02.2020	Marathon - 50mi (TPE)	50mi	38	7:34:19 h	
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50 h	
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40 h	
2538574	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:30:49 h	台灣
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47 h	
...	...	...	...	...	...	...	
2762404	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:25 h	AK V
2762405	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:36:27 h	*V
2762406	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	7:44:18 h	
2762407	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:04:50 h	P
2762408	2020	03.10.2020	Bison Ultra-Trail 50 (POL)	50km	271	8:11:43 h	Alek

63489 rows × 13 columns

```
In [113…: # Investigating UMs in USA
```

```
In [114…: df[df["Event name"].str.contains("USA",na=False)]
```

Out[114]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete
55	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	9:53:05 h	*Middle
56	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:09:35 h	*Wate
57	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:33:00 h	*Kitch
58	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:38:17 h	*Utic
59	2018	06.01.2018	Yankee Springs 50 Mile Winter Challenge (USA)	50mi	9	11:56:35 h	*Grass I
...	...	...	...	...	...	...	
7117228	2015	09.10.2015	West Virginia Trilogy 50 km (USA)	50km	79	9:40:15 h	*Pennsl
7117229	2015	09.10.2015	West Virginia Trilogy 50 km (USA)	50km	79	9:49:58 h	*Fento
			West Virginia				

7117230	2015	09.10.2015	Trilogy 50 km (USA)	50km	79	9:49:58 h	*Kimba
7117231	2015	09.10.2015	West Virginia Trilogy 50 km (USA)	50km	79	9:53:02 h	*Cumber
7117232	2015	09.10.2015	West Virginia Trilogy 50 km (USA)	50km	79	10:22:10 h	*Morgant

1365325 rows × 13 columns

In [115…

```
#Combining the filters which for the distance 50km and 50mi and for UMs i
```

In [116…

```
df[(df["Event distance/length"].isin(["50km","50mi"])) & (df["Year of eve
```

Out[116]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athle clu
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	3:17:55 h	*Normanc Park, W
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:02:32 h	*Gold Ba W
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:07:57 h	*Vasho W
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:22:02 h	*G Harbor, W

2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:27:34 h	*Bainbridg Island, W
...	...	...	...	...	...	...	
2760957	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:07:48 h	*Ea Lansing, I
2760958	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:27:22 h	*Travers City, I
2760959	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:27:24 h	*Travers City, I
2760960	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:38:30 h	*Mason, I
2760961	2020	03.10.2020	Yankee Springs Fall Trail Run Festival (USA)	50km	30	7:59:53 h	Na

26524 rows × 13 columns

In [117…
```
filt_df=df[(df["Event distance/length"].isin(["50km","50mi"])) & (df["Yea
```

In [118…
```
filt_df.head()
```

Out[118]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlet clu
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	3:17:55 h	*Normand Park, W
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:02:32 h	*Gold Ba W
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:07:57 h	*Vashor W
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:22:02 h	*Gi Harbor, W
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition (USA)	50km	20	4:27:34 h	*Bainbridg Island, W

In [119…

```
filt_df.shape
```

Out[119]:

```
(26524, 13)
```

In [120…

```
#Removing unnecessary (USA) substring from the event name because We filt
```

In [121…

```
filt_df["Event name"]=filt_df["Event name"].str.replace("(USA)","")
```

In [122…

```
filt_df.head(5)
```

Out[122]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlet clu
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55 h	*Normand Park, W
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32 h	*Gold Ba W
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57 h	*Vashor W
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02 h	*Gi Harbor, W
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34 h	*Bainbridg Island, W

In [123…

```
#Adding a new column as Athlete age
```

In [124…

```
filt_df["Athlete age"]=(2020-filt_df["Athlete year of birth"])
```

In [125…

```
filt_df.head(5)
```

Out[125]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlet clu
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55 h	*Normand Park, W
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32 h	*Gold Ba W
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57 h	*Vashor W
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02 h	*Gi Harbor, W
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34 h	*Bainbridg Island, W

```
In [126…  #Removing "h" letter from Athlete performance column
```

```
In [127…  filt_df["Athlete performance"]=filt_df["Athlete performance"].str.replace
```

```
In [128…  #drop unnecessary columns:Athlete club, Athlete year of birth, Athlete ag
```

```
In [129…  filt_df.drop(["Athlete club","Athlete year of birth","Athlete age categor
```

```
In [130…  filt_df.head(5)
```

Out[130]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete country
2539945	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	USA
2539946	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	USA
2539947	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	USA
2539948	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	USA
2539949	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	USA

In [131…

```
filt_df.isna().sum()
```

Out[131]:

```
Year of event                  0
Event dates                    0
Event name                     0
Event distance/length          0
Event number of finishers      0
Athlete performance            0
Athlete country                0
Athlete gender                 0
Athlete average speed          0
Athlete ID                     0
Athlete age                  235
dtype: int64
```

In [132…

```
#Observing the null values
```

In [133…
```
#Drop the row which has null values
```

In [134…
```
filt_df.dropna(inplace=True)
```

In [135…
```
#Check the dataframe whether it has duplicates or not
```

In [136…
```
filt_df[filt_df.duplicated()]
```

Out[136]:

Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete country	Athlete gender	Athlet averag spee

In [137…
```
#In conclusion, there is no duplicated row in the dataframe.
```

In [138…
```
#Now, reset the index
```

In [139…
```
filt_df.reset_index(drop=True,inplace=True)
```

In [140…
```
filt_df.head()
```

Out[140]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete country	Athlete gender
0	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	USA	M
1	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	USA	M
2	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	USA	M
3	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	USA	M
4	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	USA	M

In [141…

```
#Convert the data type of Athlete age to integer
filt_df["Athlete age"]=filt_df["Athlete age"].astype(int)
```

In [142…

```
#Now, check the last version
```

In [143…

```
filt_df.head()
```

Out[143]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete country	Athlete gender
0	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	USA	M
1	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	USA	M
2	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	USA	M
3	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	USA	M
4	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	USA	M

In [144…

```
filt_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26289 entries, 0 to 26288
Data columns (total 11 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   Year of event              26289 non-null  int64 
 1   Event dates                26289 non-null  object
 2   Event name                 26289 non-null  object
 3   Event distance/length      26289 non-null  object
 4   Event number of finishers  26289 non-null  int64 
 5   Athlete performance        26289 non-null  object
 6   Athlete country            26289 non-null  object
 7   Athlete gender             26289 non-null  object
 8   Athlete average speed      26289 non-null  object
 9   Athlete ID                 26289 non-null  int64 
 10  Athlete age                26289 non-null  int64 
dtypes: int64(4), object(7)
memory usage: 2.2+ MB
```

In [145…

```
filt_df.shape
```

Out[145]:

```
(26289, 11)
```

In [146…

```
#Fixing the data types
```

In [147…

```
filt_df["Athlete average speed"]=filt_df["Athlete average speed"].astype(
```

In [148…

```
# Need to rename to columns to make them more functional
```

In [149…

```
filt_df.rename(columns= {"Year of event":"year",
                         "Event dates": "race_day",
                         "Event name":  "race_name",
                         "Event distance/length": "race_distance",
                         "Event number of finishers": "num_finishers",
                         "Athlete performance": "athl_performance",
                         "Athlete country": "athl_country",
                         "Athlete gender": "gender",
                         "Athlete average speed": "athl_avg_speed",
                         "Athlete ID": "athl_id",
                         "Athlete age": "athl_age"},inplace=True)
```

In [150…

```
filt_df.head()
```

Out[150]:

	year	race_day	race_name	race_distance	num_finishers	athl_performance	athl_
0	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	3:17:55	
1	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:02:32	
2	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:07:57	
3	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:22:02	
4	2020	02.02.2020	West Seattle Beach Run - Winter Edition	50km	20	4:27:34	

In [151…

```
#reorder columns
```

In [152…

```
df1=filt_df[["race_name",
             "race_day",
             "year",
             "race_distance",
             "num_finishers",
             "athl_id",
             "gender",
             "athl_age",
             "athl_country",
             "athl_performance",
             "athl_avg_speed"]]
```

In [153…

```
df1.head()
```

Out[153]:

	race_name	race_day	year	race_distance	num_finishers	athl_id	gender	athl_a
0	West Seattle Beach Run - Winter Edition	02.02.2020	2020	50km	20	71287	M	
1	West Seattle Beach Run - Winter Edition	02.02.2020	2020	50km	20	629508	M	
2	West Seattle Beach Run - Winter Edition	02.02.2020	2020	50km	20	64838	M	
3	West Seattle Beach Run - Winter Edition	02.02.2020	2020	50km	20	704450	M	
4	West Seattle Beach Run - Winter Edition	02.02.2020	2020	50km	20	810281	M	

In [154…

```
#year columns is unnecesary right now because we know that all the races
```

In [155…

```
df1.drop("year",inplace=True,axis=1)
```

In [156…

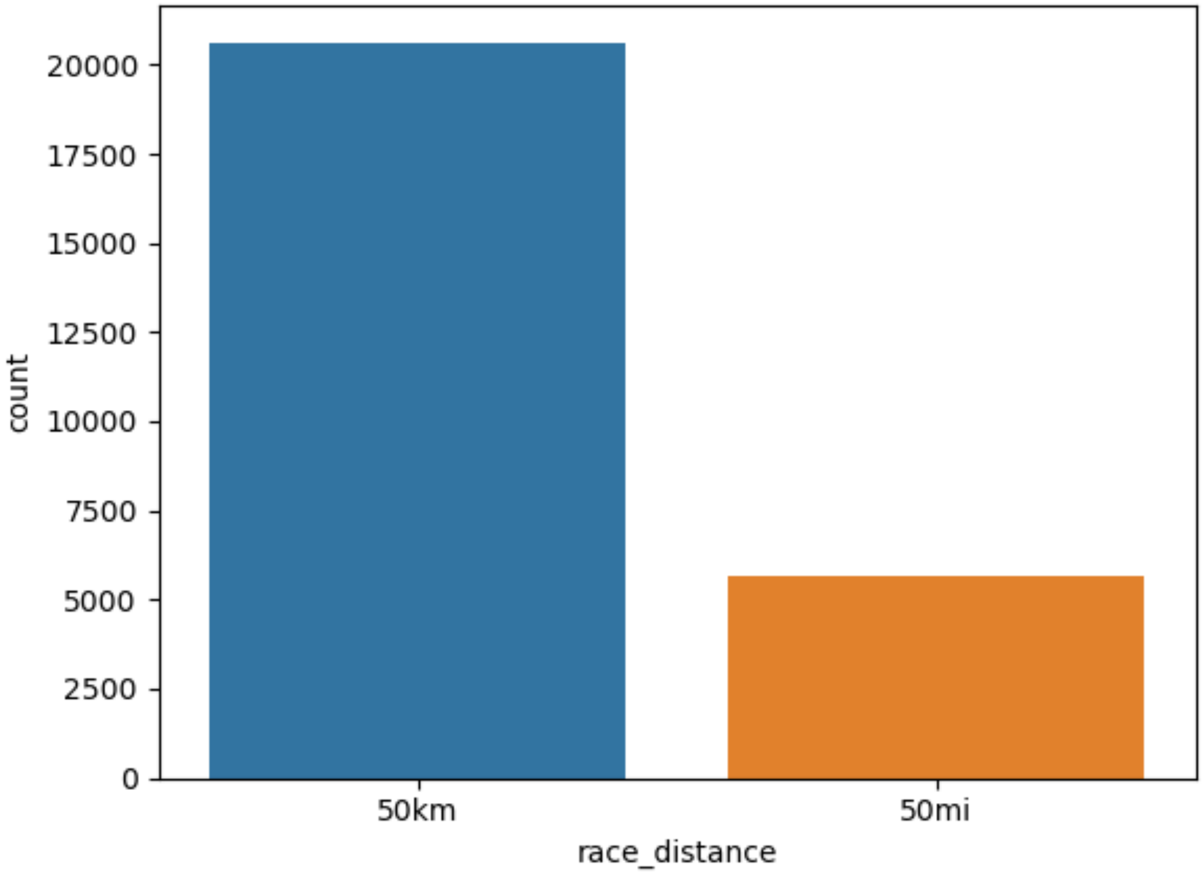
```
df1.head()
```

Out[156]:

	race_name	race_day	race_distance	num_finishers	athl_id	gender	athl_age	atl
0	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	71287	M	29	
1	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	629508	M	39	
2	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	64838	M	21	
3	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	704450	M	37	
4	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	810281	M	43	

In [157…

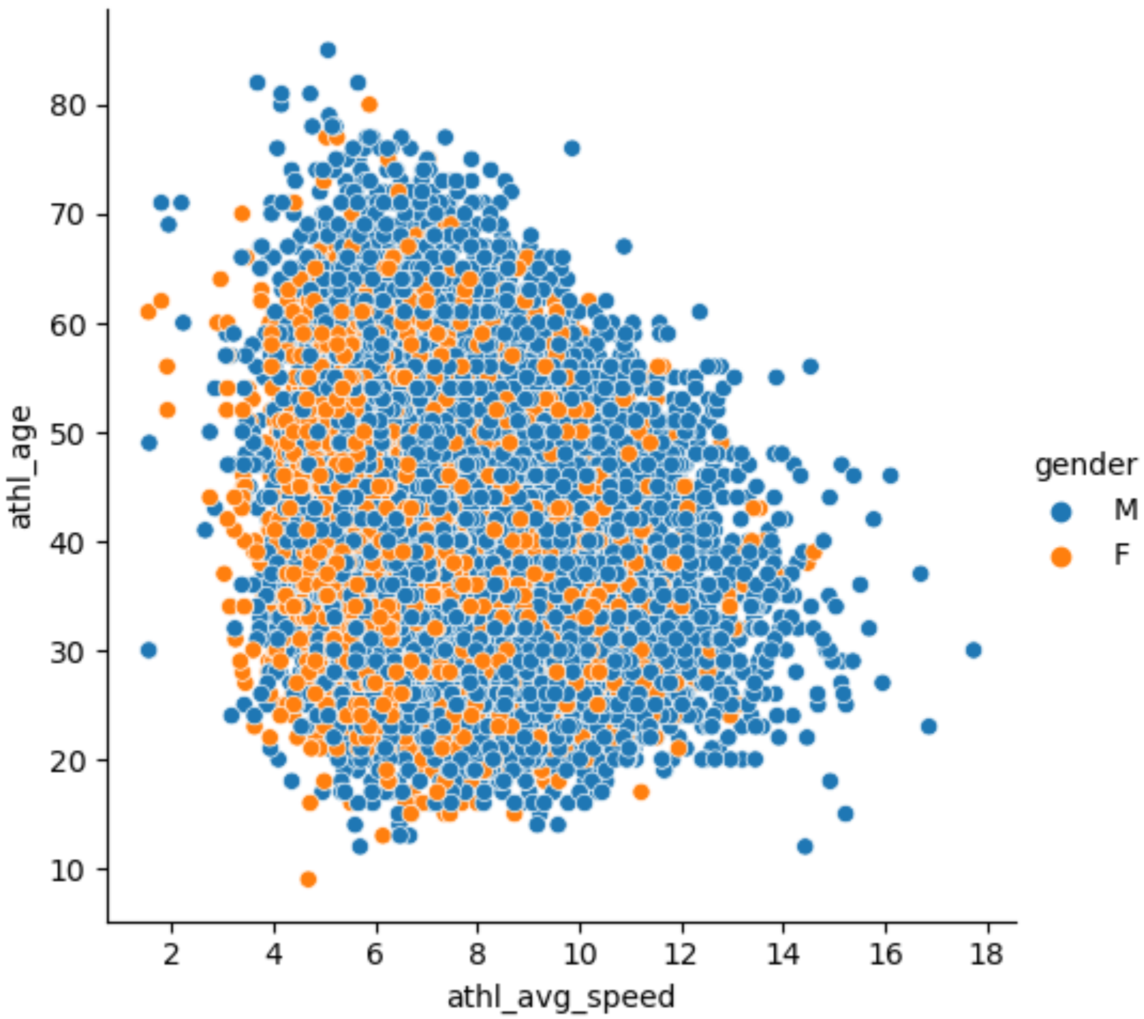
```
sns.countplot(data=df1,x="race_distance");
```



```
In [158… #Question1: Difference in speed for the 50k and 50mi male to female
```

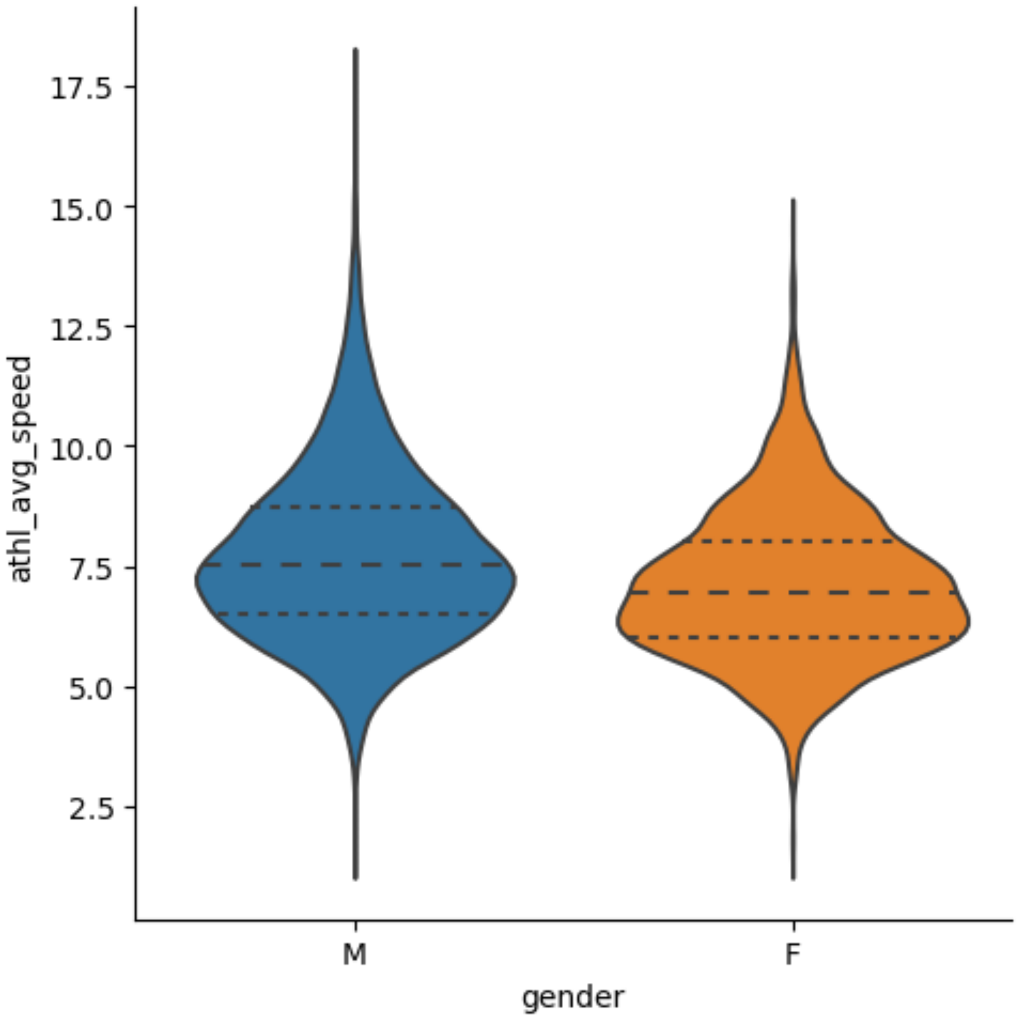
```
In [159… df_50km=df1[df1["race_distance"]=="50km"]
```

```
In [160… sns.relplot(data=df_50km,x="athl_avg_speed",y="athl_age",hue="gender");
```



In [161…

```
sns.catplot(data=df_50km,y="athl_avg_speed",x="gender",kind="violin",inne
```



In [162…

```
df_50km.groupby("gender")["athl_avg_speed"].agg([("mean", np.mean), ("med
```

Out[162]:

	mean	median	max_speed	min_speed
gender				
F	7.092015	6.927	14.614	1.541
M	7.743376	7.508	17.746	1.547

In [163…

```
df_50mi=df1[df1["race_distance"]=="50mi"]
```

In [164…

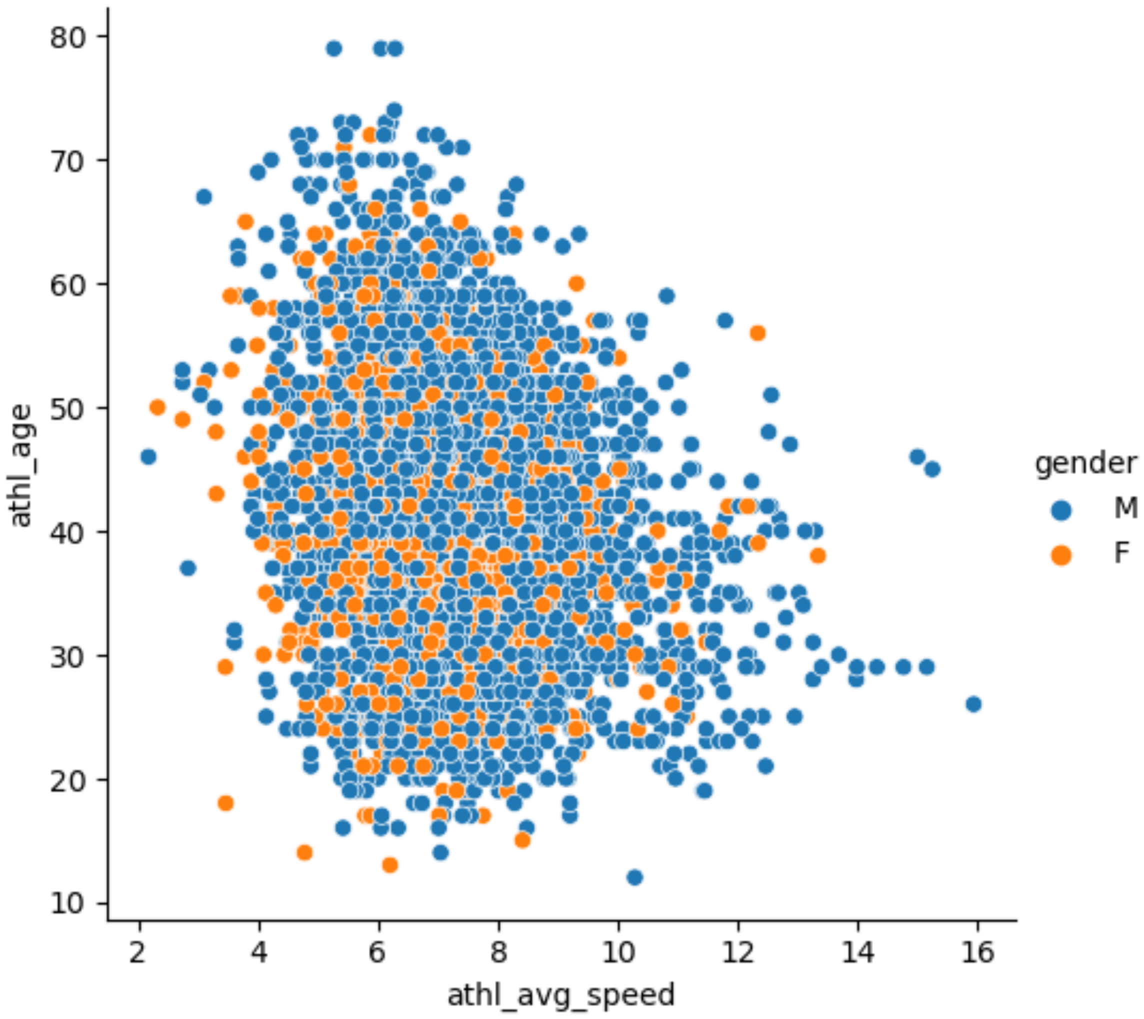
```
df_50mi.head()
```

Out[164]:

	race_name	race_day	race_distance	num_finishers	athl_id	gender	athl_age
433	Elephant Mountain 50 Mile	01.02.2020	50mi	10	86674	M	33
434	Elephant Mountain 50 Mile	01.02.2020	50mi	10	53268	M	35
435	Elephant Mountain 50 Mile	01.02.2020	50mi	10	778567	M	32
436	Elephant Mountain 50 Mile	01.02.2020	50mi	10	209242	M	41
437	Elephant Mountain 50 Mile	01.02.2020	50mi	10	810742	M	23

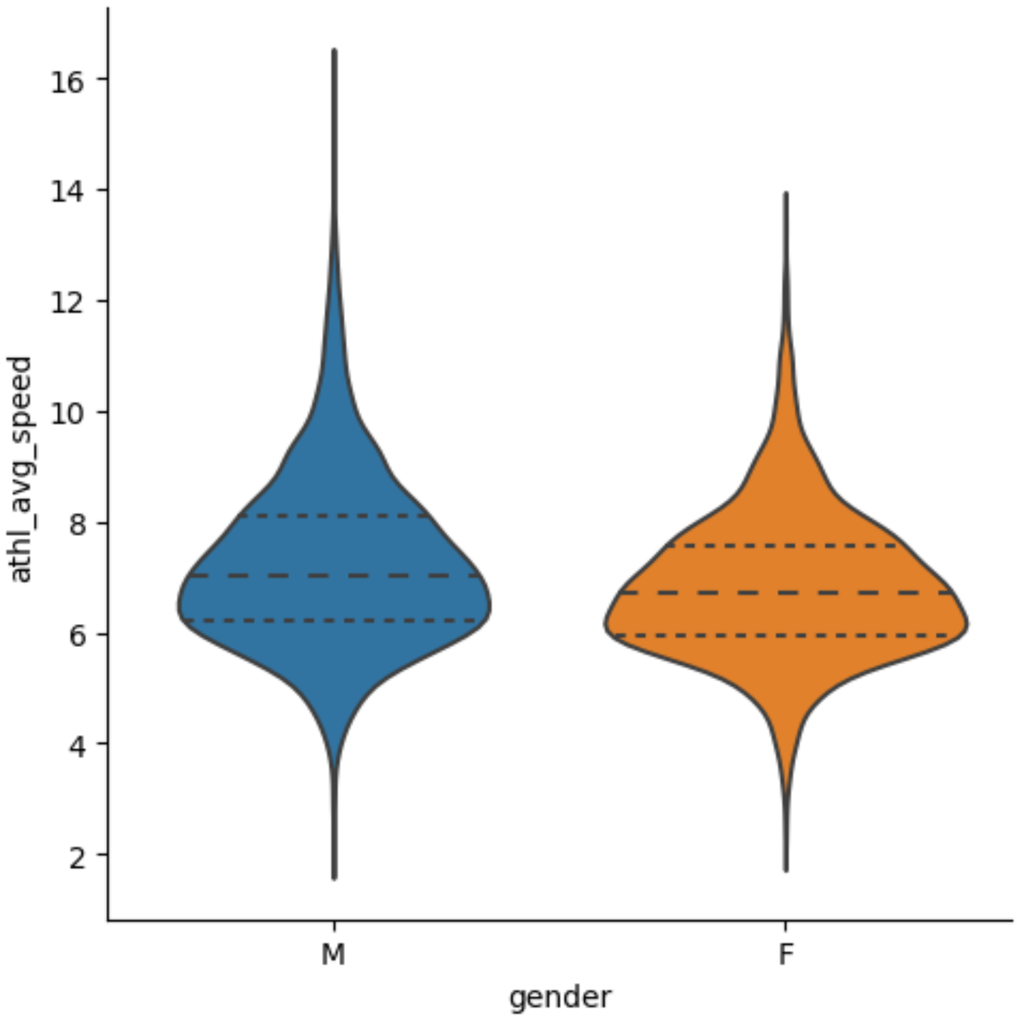
In [165…

```
sns.relplot(data=df_50mi,x="athl_avg_speed",y="athl_age",hue="gender");
```



In [166…

```
sns.catplot(data=df_50mi,y="athl_avg_speed",x="gender",kind="violin",inne
```



In [167…

```
df_50mi.groupby("gender")["athl_avg_speed"].agg([("mean", np.mean), ("med
```

Out[167]:

gender	mean	median	max_speed	min_speed
F	6.830764	6.6940	13.335	2.323
M	7.249766	7.0045	15.930	2.170

In [168…

```
#What age group are the best in the 50mi and 50 km race ?
```

In [169…

```
df_50mi["athl_age"].sort_values()
```

Out[169]:

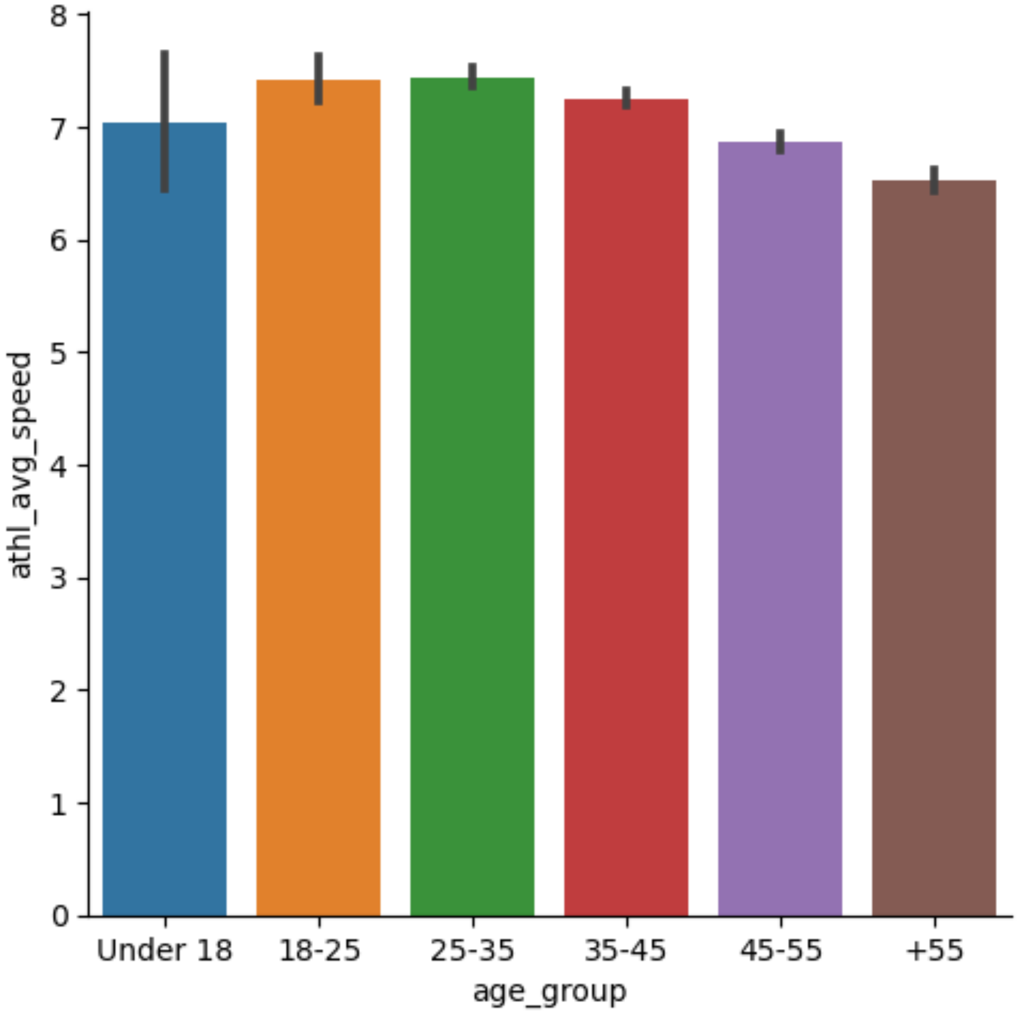
```
5756      12
7607      13
6784      14
22096     14
25222     15
          ..
25457     74
23623     79
15486     79
21709     79
23028     79
Name: athl_age, Length: 5657, dtype: int64
```

In [170…

```
df_50mi["age_group"]=df_50mi["athl_age"].apply(lambda x: "Under 18" if x
                                                         "18-25" if 18 <=
                                                         "25-35" if 25 <=
                                                         "35-45" if 35 <=
                                                         "45-55" if 45 <=
                                                         "+55")
```

In [171…

```
sns.catplot(data=df_50mi,x="age_group",y="athl_avg_speed",kind="bar",orde
```



In [172…

```
df_50mi.groupby("age_group")["athl_avg_speed"].mean().reset_index(name="m
```

Out[172]:

	age_group	mean
2	25-35	7.441253
1	18-25	7.424739
3	35-45	7.255109
5	Under 18	7.036316
4	45-55	6.872478
0	+55	6.518624

In [173…

```python
df_50km["athl_age"].sort_values()
```

Out[173]:

```
24707     9
7100     12
24273    12
12515    13
14957    13
         ..
3981     81
12773    81
13531    82
6991     82
808      85
Name: athl_age, Length: 20632, dtype: int64
```

In [174…

```python
df_50km["age_group"]=df_50km["athl_age"].apply(lambda x: "Under 18" if x
                                                          "18-25" if 18 <=
                                                          "25-35" if 25 <=
                                                          "35-45" if 35 <=
                                                          "45-55" if 45 <=
                                                          "+55")
```

In [175…

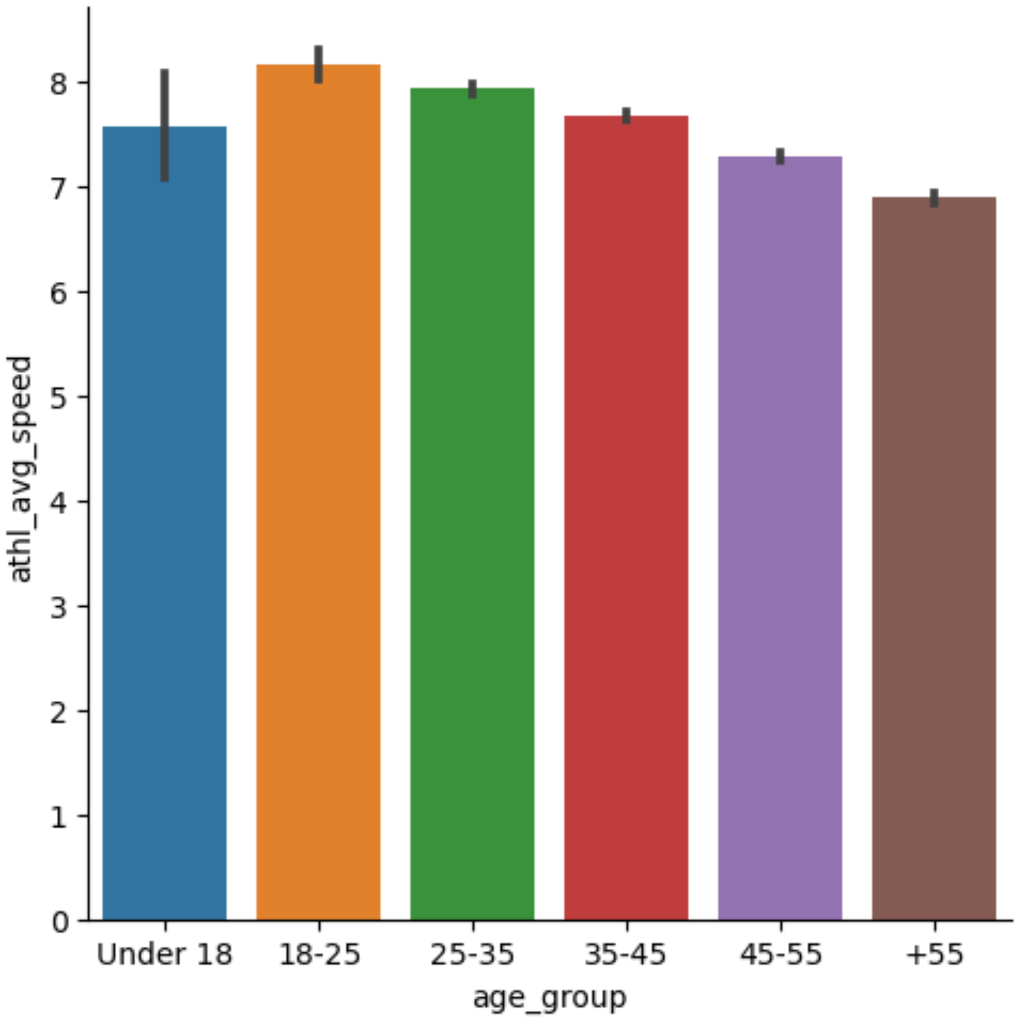
```python
df_50km.head()
```

Out[175]:

	race_name	race_day	race_distance	num_finishers	athl_id	gender	athl_age	atl
0	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	71287	M	29	
1	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	629508	M	39	
2	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	64838	M	21	
3	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	704450	M	37	
4	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	810281	M	43	

In [176…

```
sns.catplot(data=df_50km,x="age_group",y="athl_avg_speed",kind="bar",orde
```



In [177…

```
df_50km.groupby("age_group")["athl_avg_speed"].mean().reset_index(name="m
```

Out[177]:

	age_group	mean
1	18-25	8.149263
2	25-35	7.922358
3	35-45	7.657472
5	Under 18	7.556508
4	45-55	7.270246
0	+55	6.885121

In [178…

```
df1["age_group"]=df1["athl_age"].apply(lambda x: "Under 18" if x < 18 els
                                                         "18-25" if 18 <=
                                                         "25-35" if 25 <=
                                                         "35-45" if 35 <=
                                                         "45-55" if 45 <=
                                                         "+55")
```

In [179…

```
df1.head()
```

Out[179]:

	race_name	race_day	race_distance	num_finishers	athl_id	gender	athl_age	atl
0	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	71287	M	29	
1	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	629508	M	39	
2	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	64838	M	21	
3	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	704450	M	37	
4	West Seattle Beach Run - Winter Edition	02.02.2020	50km	20	810281	M	43	

In [180…

```
#Fixing the dates
```

In [181…

```
df1[df1["race_day"].str.contains(r"\.\.")]
```

Out[181]:

race_name	race_day	race_distance	num_finishers	athl_id	gender	athl_age	athl_c

In [182…

```
df1.loc[df1["race_day"].str.contains(r"\.\.", na=False), "race_day"] = df
```

In [183…

```
df1["race_day"].value_counts()
```

Out[183]:
```
race_day
07.03.2020         1660
25.01.2020         1167
21.11.2020         1165
11.01.2020         1138
29.02.2020         1052
                   ... 
28.-29.11.2020       11
05.07.2020            9
12.03.2020            5
05.-07.09.2020        4
11.03.2020            3
Name: count, Length: 108, dtype: int64
```

In [184…
```
df1.loc[df1["race_day"].str.contains(r"\.\-", na=False), "race_day"]=df1[
```

In [185…
```
df1["race_day"].value_counts()
```

Out[185]:
```
race_day
07.03.2020    1660
25.01.2020    1167
21.11.2020    1165
11.01.2020    1138
29.02.2020    1052
              ... 
29.11.2020      11
05.07.2020       9
12.03.2020       5
07.09.2020       4
11.03.2020       3
Name: count, Length: 95, dtype: int64
```

In [186…
```
df1["race_month"]=df1["race_day"].str.split(".").str.get(1).astype(int)
```

In [187…
```
df1["race_season"]=df1["race_month"].apply(lambda x: "Spring" if 2<x<=5 e
                                                      "Summer" if 5<x<=8 e
                                                      "Fall"   if 8<x<=11
                                                      "Winter")
```

In [188…
```
df1.drop("race_month",axis=1,inplace=True)
```

In [189…
```
df1.groupby(["race_season","age_group"])["athl_avg_speed"].mean()
```

Out[189]:

```
race_season  age_group
Fall         +55          6.668663
             18-25        7.798949
             25-35        7.776901
             35-45        7.481156
             45-55        7.054447
             Under 18     7.228088
Spring       +55          6.983531
             18-25        8.440274
             25-35        7.951876
             35-45        7.912074
             45-55        7.466796
             Under 18     9.746375
Summer       +55          6.304400
             18-25        7.167944
             25-35        7.099191
             35-45        6.967286
             45-55        6.567397
             Under 18     6.692154
Winter       +55          6.974438
             18-25        8.230388
             25-35        8.025848
             35-45        7.696232
             45-55        7.372479
             Under 18     7.384103
Name: athl_avg_speed, dtype: float64
```

In [190…

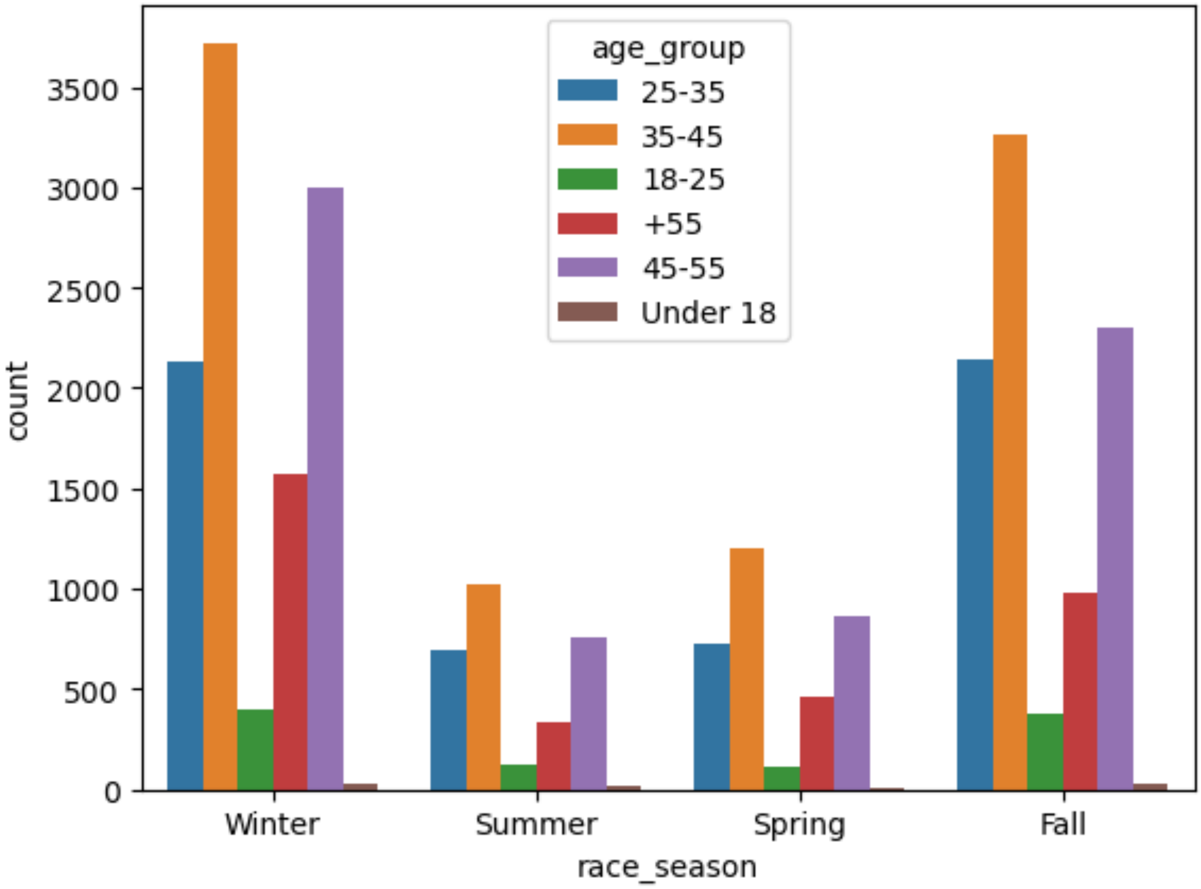
```
df1.groupby("race_season")["athl_avg_speed"].agg(["mean","count"]).sort_v
```

Out[190]:

	mean	count
race_season		
Spring	7.703542	3385
Winter	7.585842	10853
Fall	7.367121	9112
Summer	6.826808	2939

In [191…

```
sns.countplot(data=df1,x="race_season",hue="age_group");
```


age_group
25-35
35-45
18-25
+55
45-55
Under 18
count
0
500
1000
1500
2000
2500
3000
3500
Winter
Summer
Spring
Fall
race_season

```
In [ ]:
```