1º Passo:

Analisar as ultra maratonas realizadas no ano de 2020 de 50km e 50mi

Descobrir país com o maior número de eventos

Buscar evento com o maior número de participantes

In [5]:
```
#Importando as bibliotecas
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
import numpy as np
```

In [6]:
```
#Importar e visualizar dataset
df = pd.read_csv('TWO_CENTURIES_OF_UM_RACES.csv', low_memory = False)

df.head()
```

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	

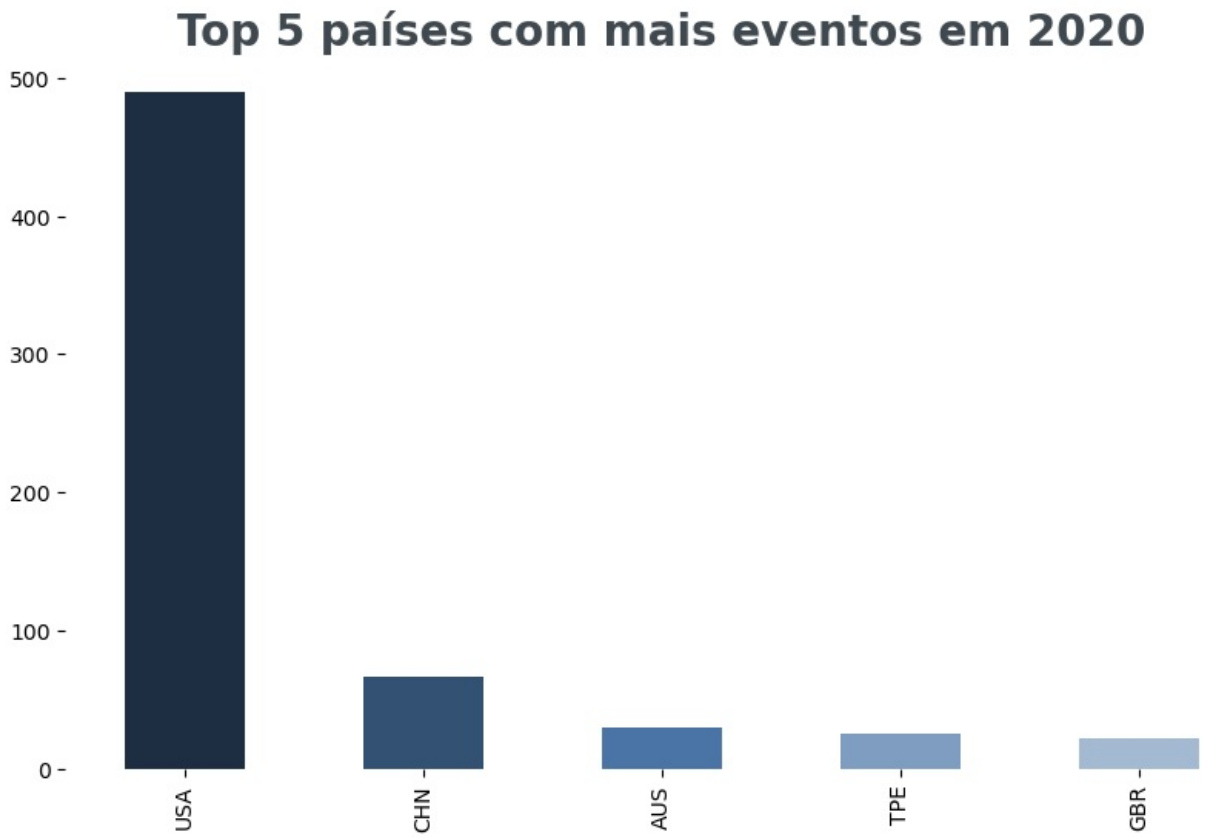
In [7]:
```
#Encontrar as ultra maratonas com 50km ou 50mi no ano de 2020
df1 = df[(df['Event distance/length'].isin(['50mi', '50km'])) & (df['Year of event']==2020)]

df1.head()
```

Out[7]:

	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	Ath aver sp
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	10
2538572	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	7:43:50 h	NaN	AUS	1974.0	M	M45	10
2538573	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:04:40 h	NaN	TPE	1976.0	M	M40	9
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	9
2538575	2020	07.-09.02.2020	Taipei 48hr Ultra Marathon - 50mi (TPE)	50mi	38	8:34:47 h	NaN	TPE	1964.0	M	M55	9

In [8]:

```
#Top 5 países com a maior quantiade de maratonas em 2020

#Plotar gráfico de barras
plt.figure(figsize = (10,6))
colors = ['#1D2E42', '#335173', '#4974A5', '#7F9DC0', '#A4B9D2']
df1.drop_duplicates(subset=['Event name'])['Event name'].str.split('(').str.get(1).str.split(')').str.get(0).va

#Formatar gráfico
plt.xlabel('')
plt.title('Top 5 países com mais eventos em 2020', fontsize = 20, color = '#414950', fontweight = 'bold')
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['left'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
```

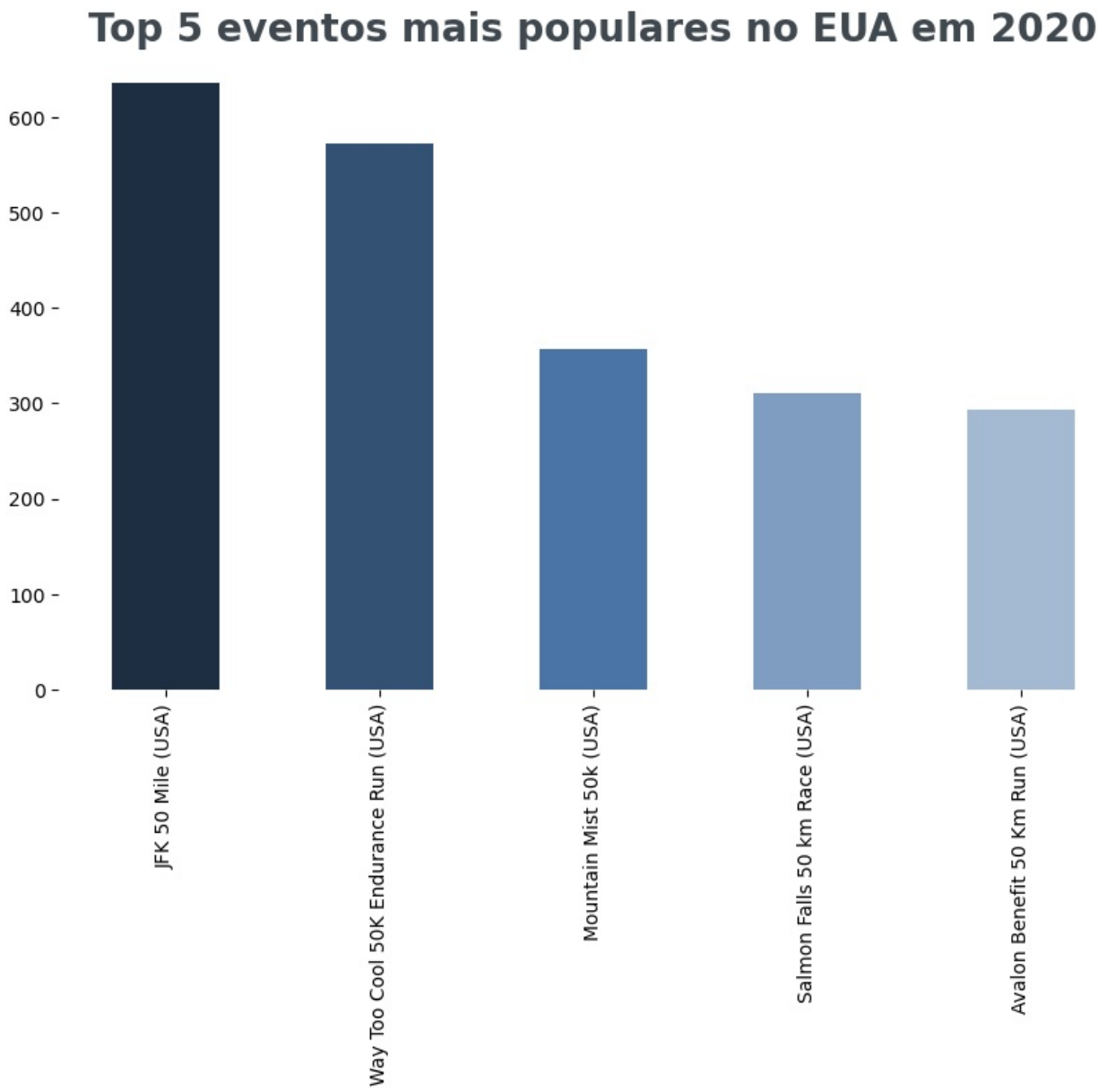


Dentre os top 5 o Estados Unidos lidera com a maior quantidade de ultra maratonas com quase 500 eventos em 2020.

In [10]:
```
# Filtrar somente maratonas nos EUA
df1 = df1[df1['Event name'].str.split('(').str.get(1).str.split(')').str.get(0)=='USA']

#Exibir quantidade de linhas e colunas
df1.shape
```

Out[10]: (26090, 13)

In [11]:
```
#Top 5 eventos mais populares no EUA em 2020

#Plotar gráfico
plt.figure(figsize = (10,6))
colors = ['#1D2E42', '#335173', '#4974A5', '#7F9DC0', '#A4B9D2']
df1['Event name'].value_counts().head(5).plot(kind = 'bar', color= colors)

#formatar gráfico
plt.xlabel('')
plt.title('Top 5 eventos mais populares no EUA em 2020', fontsize = 20, color = '#414950', fontweight = 'bold')
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['left'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
```



Dos eventos que ocorreram no EUA no ano de 2020, o JFK 50 Mile foi o mais popular dentre eles.

2º Passo

Limpeza e organização dos dados

In [14]:
```
#Filtrar apenas JFK
jfk_event = df1[df1['Event name'] == 'JFK 50 Mile (USA)']

#Visualizar DF
jfk_event.head()
```

Out[14]:

	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
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:18:42 h	*Cedar City, UT	USA	1991.0	M	M23	15.149
2713362	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:27:07 h	*Flagstaff, AZ	USA	1991.0	M	M23	14.759
2713363	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:37:16 h	*Colorado Springs, CO	USA	1991.0	M	M23	14.315
2713364	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:45:33 h	*Durango, CO	USA	1992.0	M	M23	13.972
2713365	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:52:54 h	*Ann Arbor, MI	USA	1990.0	M	M23	13.681

In [15]:
```
#Verificar se existem valores nulos
jfk_event.isnull().sum()
```

Out[15]:
```
Year of event                0
Event dates                  0
Event name                   0
Event distance/length        0
Event number of finishers    0
Athlete performance          0
Athlete club                 4
Athlete country              0
Athlete year of birth        4
Athlete gender               0
Athlete age category         0
Athlete average speed        0
Athlete ID                   0
dtype: int64
```

In [16]:
```
# Verificar tamanho do DF
jfk_event.shape
```

Out[16]:
```
(636, 13)
```

In [17]:
```
# Visualizar os dados nulos
jfk_event[jfk_event.isnull().any(axis=1)]
```

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	Athlete average speed	A
2713466	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	8:59:27 h	NaN	XXX	NaN	M	M35	8.95	
2713524	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	9:50:08 h	NaN	XXX	NaN	M	M35	8.181	
2713744	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	11:38:53 h	NaN	XXX	NaN	M	M35	6.908	
2713948	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	13:12:24 h	NaN	XXX	NaN	M	M35	6.093	

Houve registro do mesmo atleta em diferentes linhas.

In [19]:
```
#Remove os valores nulos
jfk_event = jfk_event.dropna()
```

In [20]:
```
# Remove a coluna 'Athlete year of birth', 'Athlete club' e 'Athlete Country'
jfk_event = jfk_event.drop(['Athlete year of birth', 'Athlete club', 'Athlete country'], axis=1)
```

In [21]:
```
# Remove "h" da coluna 'Athlete performance'
jfk_event['Athlete performance'] = jfk_event['Athlete performance'].str.replace(' h','')
```

In [22]:
```
# Verifica os padrões dos caracteres que antecedem os valores das idades.
jfk_event['Athlete age category'].unique()
```

Out[22]:
```
array(['M23', 'M35', 'MU23', 'W35', 'M40', 'W23', 'M55', 'M45', 'W40',
       'M50', 'W50', 'W45', 'M60', 'WU23', 'W55', 'M65', 'M70', 'W65',
       'W60'], dtype=object)
```

In [23]:
```
# Remove os caracteres que antecedem as idades
jfk_event['Athlete age category'] = (jfk_event['Athlete age category'].str.replace('F','').str.replace('M','').
```

In [24]:
```
jfk_event.head()
```

Out[24]:

	Year of event	Event dates	Event name	Event distance/length	Event number of finishers	Athlete performance	Athlete gender	Athlete age category	Athlete average speed	Athlete ID
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:18:42	M	23	15.149	52105
2713362	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:27:07	M	23	14.759	168122
2713363	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:37:16	M	23	14.315	848700
2713364	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:45:33	M	23	13.972	37196
2713365	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:52:54	M	23	13.681	153351

In [25]:
```
# Verificar colunas
jfk_event.columns
```

Out[25]:
```
Index(['Year of event', 'Event dates', 'Event name', 'Event distance/length',
       'Event number of finishers', 'Athlete performance', 'Athlete gender',
       'Athlete age category', 'Athlete average speed', 'Athlete ID'],
      dtype='object')
```

In [26]:
```
# Renomear Colunas
jfk_event.rename(columns={
    'Year of event': 'year',
    'Event dates': 'event_date',
    'Event name': 'event_name',
    'Event distance/length': 'distance',
    'Event number of finishers': 'finishers',
    'Athlete performance': 'performance',
    'Athlete gender': 'gender',
    'Athlete age category': 'age',
    'Athlete average speed': 'avg_speed',
    'Athlete ID': 'id'
}, inplace=True)
```

In [27]:
```
#Ver as colunas do df
jfk_event.columns
```

Out[27]:
```
Index(['year', 'event_date', 'event_name', 'distance', 'finishers',
       'performance', 'gender', 'age', 'avg_speed', 'id'],
      dtype='object')
```

In [28]:
```
#verificar o tipo de dado de cada coluna
jfk_event.dtypes
```

Out[28]:
```
year           int64
event_date    object
event_name    object
distance      object
finishers      int64
performance   object
gender        object
age           object
avg_speed     object
id             int64
dtype: object
```

In [29]:
```
jfk_event.head(1)
```

Out[29]:

	year	event_date	event_name	distance	finishers	performance	gender	age	avg_speed	id
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	5:18:42	M	23	15.149	52105

In [30]:
```
# Average speed para float
jfk_event['avg_speed'] = jfk_event['avg_speed'].astype('float')

# Age para int
jfk_event['age'] = jfk_event['age'].astype('int64')
```

In [31]:
```
#Performance para datetime
jfk_event ['performance'] = pd.to_datetime('2020-11-21 ' + jfk_event['performance'])
```

3º Passo

Distribuir participantes por gênero

Classificar esses participantes por idade

In [33]:
```
jfk_event.head()
```

Out[33]:

	year	event_date	event_name	distance	finishers	performance	gender	age	avg_speed	id
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:18:42	M	23	15.149	52105
2713362	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:27:07	M	23	14.759	168122
2713363	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:37:16	M	23	14.315	848700
2713364	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:45:33	M	23	13.972	37196
2713365	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:52:54	M	23	13.681	153351

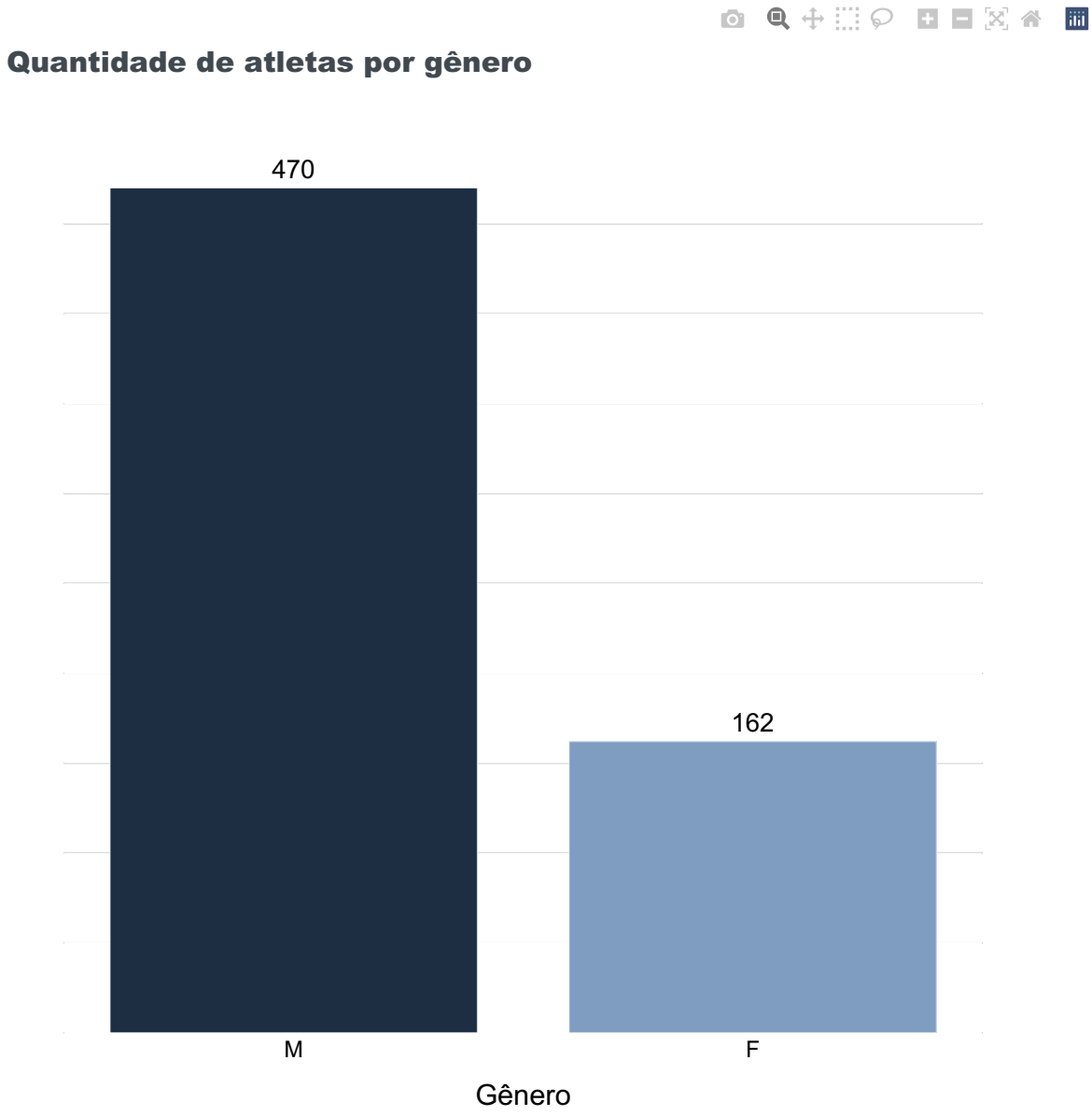
In [34]:
```
#Conta o número de linhas
jfk_event.shape
```

Out[34]:
```
(632, 10)
```

Podemos notar que a prova não houve nenhum desistente. Foram 636 participantes (Atualmente são 632 linhas por conta da remoção dos valores nulos) e nenhum deles quebrou. "Quebrar" é um termo utilizado para quando um atleta em determinado momento da prova e não consegue completar a corrida (ou precisa diminuir muito o ritmo para chegar até o final).

Uma das hipóteses é devido à estação do ano que a prova foi realizada. Na data que a ultra maratona foi realizada era outono, a temperatura é mais amena se comparada ao verão, por exemplo.

In [36]:
```
# Contar Atletas por gênero
contagem_genero = jfk_event['gender'].value_counts()

#Plotar gráfico
fig = go.Figure(go.Bar(
    x=contagem_genero.index,
    y=contagem_genero.values,
    text=contagem_genero.values,
    textposition='outside',
    marker=dict(color=['#1D2E42', '#7F9DC0']),
    textfont=dict(size=18, color='black', family='Arial')
))

# Formatar gráfico
fig.update_layout(
    title='Quantidade de atletas por gênero',
    title_font=dict(size=20, color='#414950', family='Arial Black'),
    yaxis=dict(title='', showticklabels=False),
    xaxis=dict(title="Gênero", title_font=dict(size=20, color="black",family="Arial"),
    tickfont=dict(size=16, color="black", family="Arial")
            ),
    width=800,
    height=800,
    legend=dict(font=dict(size=20)),
    plot_bgcolor='white',
    paper_bgcolor='white'
)

fig.show()
```



Podemos notar que a maioria dos atletas são do sexo masculino.

Para analisar a separação dos gêneros por idade, vamos criar uma coluna com a classificação da categoria por idade, sendo elas:

- Atletas com menos de 20: 20-
- Atletas entre 20 e 30: 20 a 30
- Atletas entre 31 e 40: 31 a 40
- Atletas entre 41 e 50: 41 a 50
- Atletas maiores que 50: 50+

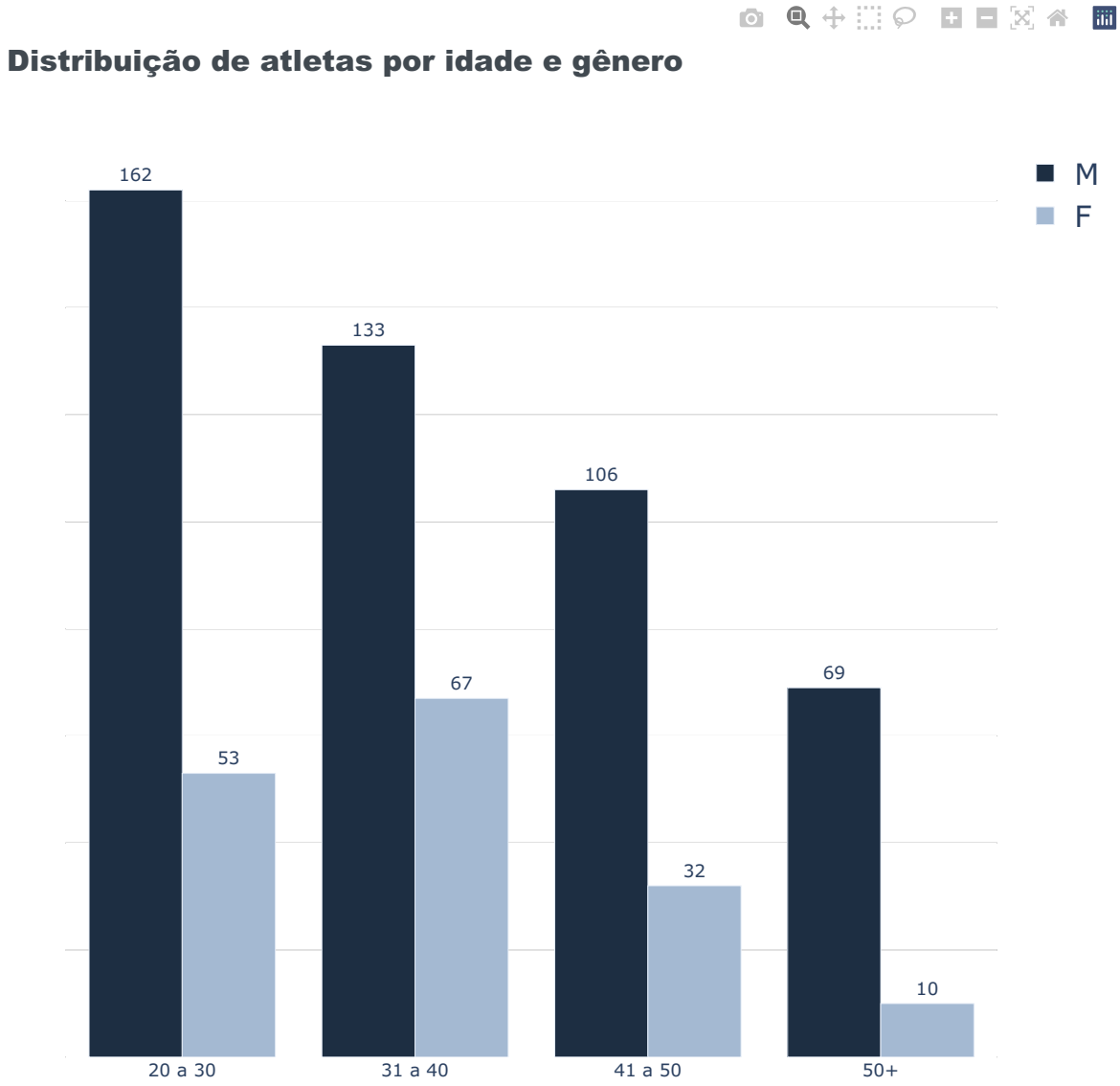
In [39]:
```
# Criar função de categoria por idades
def age_class(x):
    if(x<20):
        return '20-'
    elif (x>=20 and x<=30):
        return '20 a 30'
    elif (x>30 and x<=40):
        return '31 a 40'
    elif (x>40 and x<=50 ):
        return '41 a 50'
    else :
        return '50+'

# Criar coluna
jfk_event['age_cat']=jfk_event['age'].apply(age_class)

jfk_event.head()
```

Out[39]:

	year	event_date	event_name	distance	finishers	performance	gender	age	avg_speed	id	age_cat
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:18:42	M	23	15.149	52105	20 a 30
2713362	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:27:07	M	23	14.759	168122	20 a 30
2713363	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:37:16	M	23	14.315	848700	20 a 30
2713364	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:45:33	M	23	13.972	37196	20 a 30
2713365	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:52:54	M	23	13.681	153351	20 a 30

In [40]:

```
# Exemplo de dados de contagem por faixa etária e gênero
contagem_idade_genero = jfk_event.groupby(['age_cat', 'gender']).size().unstack(fill_value=0)

fig = go.Figure()

# Plotar gráfico para o gênero masculino
fig.add_trace(go.Bar(
    x=contagem_idade_genero.index,
    y=contagem_idade_genero['M'],
    text=contagem_idade_genero['M'],
    textposition='outside',
    name='M',
    marker_color='#1D2E42'
))

# Plotar gráfico para o gênero feminino
fig.add_trace(go.Bar(
    x=contagem_idade_genero.index,
    y=contagem_idade_genero['F'],
    name='F',
    text=contagem_idade_genero['F'],
    textposition='outside',
    marker_color='#A4B9D2'
))

# Formatar gráfico
fig.update_layout(
    title='Distribuição de atletas por idade e gênero',
    title_font= dict(size= 20,color='#414950', family='Arial Black'),
    yaxis=dict(
        title='',
        showticklabels=False
    ),
    width=800,
    height=800,
    legend= dict(font=dict(size=20)),
    plot_bgcolor='white',
    paper_bgcolor='white'
)

fig.show()
```



A maior quantidade de atletas do público masculino se encontra entre 20 e 30 anos, enquanto o público feminino está alocado no grupo de participantes de 31 a 40 anos.

Não houve nenhum competidor com menos de 20 anos.

4º Passo

Velocidade dos atletas divididas por gênero e faixa etária

Comparação entre o primeiro e último colocado

Velocidade dos atletas vs tempo de prova

In [43]:
```
# Plotar gráfico para o gênero masculino
fig = go.Figure()
fig.add_trace(go.Violin(
    x=jfk_event['age_cat'][jfk_event['gender'] == 'M'],
    y=jfk_event['avg_speed'][jfk_event['gender'] == 'M'],
    legendgroup='M', scalegroup='M', name='Masculino',
    line_color='#1D2E42',
    fillcolor='#1D2E42',
    box_visible=True,
    meanline_visible=True
))

# Plotar gráfico para o gênero feminino
fig.add_trace(go.Violin(
    x=jfk_event['age_cat'][jfk_event['gender'] == 'F'],
    y=jfk_event['avg_speed'][jfk_event['gender'] == 'F'],
    legendgroup='F', scalegroup='F', name='Feminino',
    line_color='#A4B9D2',
    fillcolor='#A4B9D2',
```

```
    box_visible=True,
    meanline_visible=True
))

# Configuração do layout
fig.update_layout(
    title='Distribuição de Velocidade Média por Idade e Gênero',
    title_font=dict(size=20, color='#414950', family='Arial Black'),
    xaxis=dict(title=''),
    yaxis=dict(title=''),
    violinmode='group',
    width=800,
    height=600,
    plot_bgcolor='white',
    paper_bgcolor='white',
    legend=dict(title='Gênero', font=dict(size=14))
)

# Exibe o gráfico
fig.show()
```

A faixa de densidade mostra que a maioria dos corredores mantiveram a velocidade entre 6 e 8 milhas por hora (mph), indicando uma convergência nas performances em torno dessa faixa de tempo.

A maior velocidade em mph foi do gênero masculino no grupo de 20 a 30 anos, com 15,149 mph. Logo, o vencedor da competição está nessa categoria, percorrendo aproximadamente 80,47 km numa velocidade média de 24,38 km/h.

A maior velocidade em mph do grupo feminino está nos participantes de 31 a 40 anos.

In [45]:
```
#Comparação entre vencedor da prova e último colocado
(jfk_event.loc[(jfk_event['performance'] == jfk_event['performance'].min()) | (jfk_event['performance'] == jfk_
```

Out[45]:

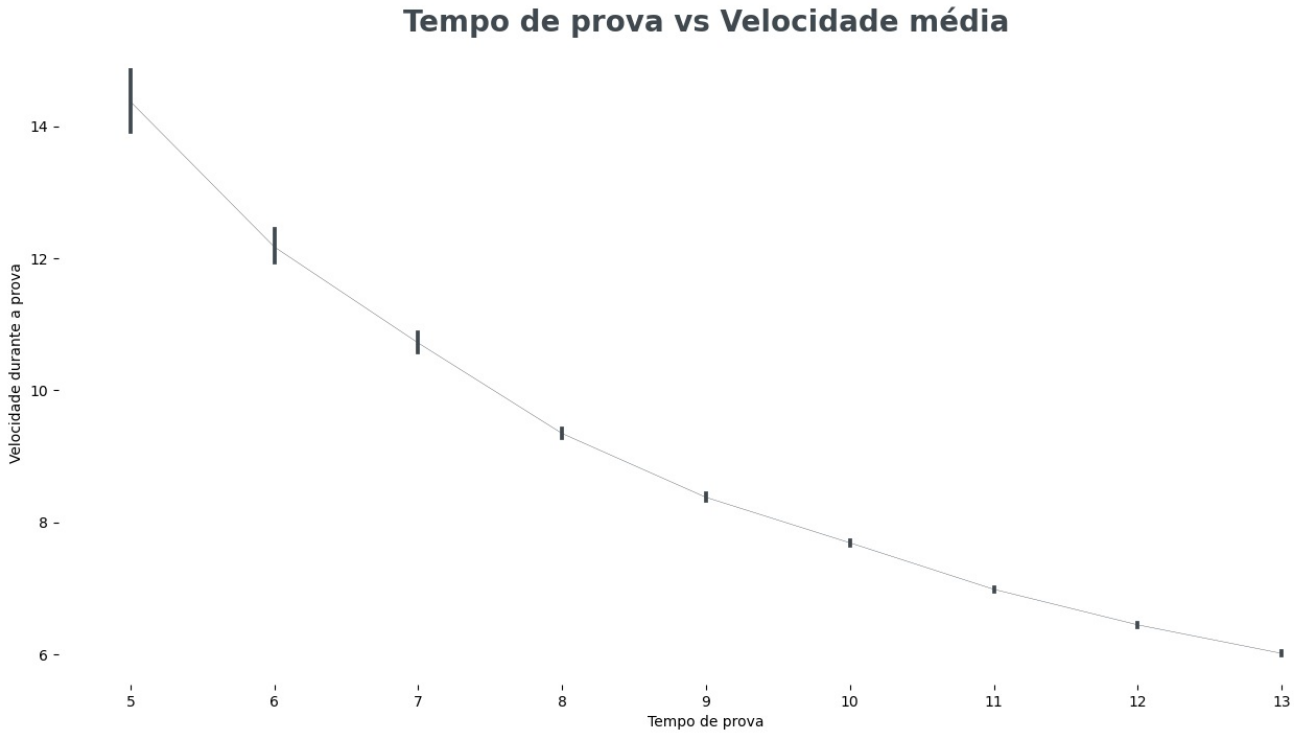
	year	event_date	event_name	distance	finishers	performance	gender	age	avg_speed	id	age_cat
2713361	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 05:18:42	M	23	15.149	52105	20 a 30
2713996	2020	21.11.2020	JFK 50 Mile (USA)	50mi	636	2020-11-21 13:47:24	M	50	5.835	366056	41 a 50

In [46]:
```
print("O primeiro colocado teve uma velocidade", round(jfk_event['avg_speed'].max() / jfk_event['avg_speed'].mi
      "vezes maior que o último colocado.")
```

```
O primeiro colocado teve uma velocidade 2.6 vezes maior que o último colocado.
```

In [47]:
```
#Plotar gráfico
```

```
plt.figure(figsize=(16,8))
sns.pointplot(
              x=jfk_event['performance'].dt.hour,
              y=jfk_event['avg_speed'],
              markers = 'o',
              estimator='mean',
              scale=0.1,
              color = '#414950'
                )

# Formatar gráfico
plt.xlabel('Tempo de prova')
plt.ylabel('Velocidade durante a prova')
plt.title('Tempo de prova vs Velocidade média', fontsize = 20, color = '#414950', fontweight = 'bold')
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['left'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
```



O gráfico mostra de forma esquematizada a relação do tempo de prova dos atletas de acordo com sua velocidade média mantida durante a prova.