

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import plotly.express as px

import folium
from folium.plugins import HeatMap

import warnings
warnings.filterwarnings('ignore')

import os

# Specify the path to your downloaded dataset folder
dataset_path = r'C:\Users\Praka\Documents\python'

# List all files in the dataset folder
for root, dirs, files in os.walk(dataset_path):
    for filename in files:
        print(os.path.join(root, filename))
```

```
C:\Users\Praka\Documents\python\Airline Dataset Updated - v2.csv
C:\Users\Praka\Documents\python\Airline Dataset Updated.csv
C:\Users\Praka\Documents\python\Airline Dataset.csv
C:\Users\Praka\Documents\python\airline-eda.ipynb
C:\Users\Praka\Documents\python\airline.ipynb
C:\Users\Praka\Documents\python\Untitled.ipynb
C:\Users\Praka\Documents\python\.ipynb_checkpoints\airline-checkpoint.ipynb
C:\Users\Praka\Documents\python\.ipynb_checkpoints\airline-eda-checkpoint.ipynb
C:\Users\Praka\Documents\python\.ipynb_checkpoints\Untitled-checkpoint.ipynb
```

```
In [3]: # Loading Dataset
df = pd.read_csv(r'C:\Users\Praka\Documents\python\Airline Dataset Updated - v2.csv')
```

```
In [4]: df.head(10)
```

Out[4]:

	Passenger ID	First Name	Last Name	Gender	Age	Nationality	Airport Name	Airport Country Code	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Pilot Name	Flight Status
0	ABVWlg	Edithe	Leggis	Female	62	Japan	Coldfoot Airport	US	United States	NAM	North America	6/28/2022	CXF	Fransisco Hazeldine	On Time
1	jkXXAX	Elwood	Catt	Male	62	Nicaragua	Kugluktuk Airport	CA	Canada	NAM	North America	12/26/2022	YCO	Marla Parsonage	On Time
2	CdUz2g	Darby	Felgate	Male	67	Russia	Grenoble-Isère Airport	FR	France	EU	Europe	1/18/2022	GNB	Rhonda Amber	On Time
3	BRS38V	Dominica	Pyle	Female	71	China	Ottawa / Gatineau Airport	CA	Canada	NAM	North America	9/16/2022	YND	Kacie Commucci	Delayed
4	9kvTLo	Bay	Pencost	Male	21	China	Gillespie Field	US	United States	NAM	North America	2/25/2022	SEE	Ebonee Tree	On Time
5	nMJKVh	Lora	Durbann	Female	55	Brazil	Coronel Horácio de Mattos Airport	BR	Brazil	SAM	South America	06-10-2022	LEC	Inglis Dolley	On Time
6	8IPFPE	Rand	Bram	Male	73	Ivory Coast	Duxford Aerodrome	GB	United Kingdom	EU	Europe	10/30/2022	QFO	Stanislas Tiffin	Cancelled
7	pqixbY	Perceval	Dallosso	Male	36	Vietnam	Maestro Wilson Fonseca Airport	BR	Brazil	SAM	South America	04-07-2022	STM	Sharyl Eastmead	Cancelled
8	QNAs2R	Aleda	Pigram	Female	35	Palestinian Territory	Venice Marco Polo Airport	IT	Italy	EU	Europe	8/20/2022	VCE	Daryn Bardsley	On Time
9	3jmudz	Burlie	Schustl	Male	13	Thailand	Vermilion Airport	CA	Canada	NAM	North America	04-06-2022	YVG	Alameda Carlyle	On Time

In [5]:

```
# Intial Data Exploration

df.shape
```

Out[5]: (98619, 15)

In [6]:

```
# Checking the datatypes

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 98619 entries, 0 to 98618
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Passenger ID          98619 non-null  object
1   First Name            98619 non-null  object
2   Last Name             98619 non-null  object
3   Gender                98619 non-null  object
4   Age                   98619 non-null  int64
5   Nationality           98619 non-null  object
6   Airport Name          98619 non-null  object
7   Airport Country Code  98619 non-null  object
8   Country Name          98619 non-null  object
9   Airport Continent     98619 non-null  object
10  Continents             98619 non-null  object
11  Departure Date         98619 non-null  object
12  Arrival Airport        98619 non-null  object
13  Pilot Name            98619 non-null  object
14  Flight Status         98619 non-null  object
dtypes: int64(1), object(14)
memory usage: 11.3+ MB
```

```
In [9]: # Checking the null values for confirmation
df.isnull().sum()
```

```
Out[9]: Passenger ID      0
First Name      0
Last Name      0
Gender      0
Age      0
Nationality      0
Airport Name      0
Airport Country Code      0
Country Name      0
Airport Continent      0
Continents      0
Departure Date      0
Arrival Airport      0
Pilot Name      0
Flight Status      0
dtype: int64
```

```
In [14]: # Data cleaning
# Let's check the dataset again
df.head(5)

# Here firstName, LastName and Passenger ID columns doesn't hold much significance so we can drop them. =====> this step followed by next section of code
```

Out[14]:

	Passenger ID	First Name	Last Name	Gender	Age	Nationality	Airport Name	Airport Country Code	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Pilot Name	Flight Status
0	ABVWlg	Edithe	Leggis	Female	62	Japan	Coldfoot Airport	US	United States	NAM	North America	6/28/2022	CXF	Fransisco Hazeldine	On Time
1	jkXXAX	Elwood	Catt	Male	62	Nicaragua	Kugluktuk Airport	CA	Canada	NAM	North America	12/26/2022	YCO	Marla Parsonage	On Time
2	CdUz2g	Darby	Felgate	Male	67	Russia	Grenoble-Isère Airport	FR	France	EU	Europe	1/18/2022	GNB	Rhonda Amber	On Time
3	BRS38V	Dominica	Pyle	Female	71	China	Ottawa / Gatineau Airport	CA	Canada	NAM	North America	9/16/2022	YND	Kacie Commucci	Delayed
4	9kvTL0	Bay	Pencost	Male	21	China	Gillespie Field	US	United States	NAM	North America	2/25/2022	SEE	Ebonee Tree	On Time

In [16]:

```
df = df.drop(['First Name', 'Last Name', 'Passenger ID'], axis = 1)
```

In [18]:

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

Out[18]:

	Gender	Age	Nationality	Airport Name	Airport Country Code	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Pilot Name	Flight Status
0	Female	62	Japan	Coldfoot Airport	US	United States	NAM	North America	6/28/2022	CXF	Fransisco Hazeldine	On Time
1	Male	62	Nicaragua	Kugluktuk Airport	CA	Canada	NAM	North America	12/26/2022	YCO	Marla Parsonage	On Time
2	Male	67	Russia	Grenoble-Isère Airport	FR	France	EU	Europe	1/18/2022	GNB	Rhonda Amber	On Time
3	Female	71	China	Ottawa / Gatineau Airport	CA	Canada	NAM	North America	9/16/2022	YND	Kacie Commucci	Delayed
4	Male	21	China	Gillespie Field	US	United States	NAM	North America	2/25/2022	SEE	Ebonee Tree	On Time

In [20]:

```
# 5. Exploratory Data Analysis
```

In [24]:

```
df['Gender'].unique()
```

Out[24]: array(['Female', 'Male'], dtype=object)

In [26]:

```
data = df['Gender'].value_counts().reset_index()
```

In [28]:

```
data
```

Out[28]:

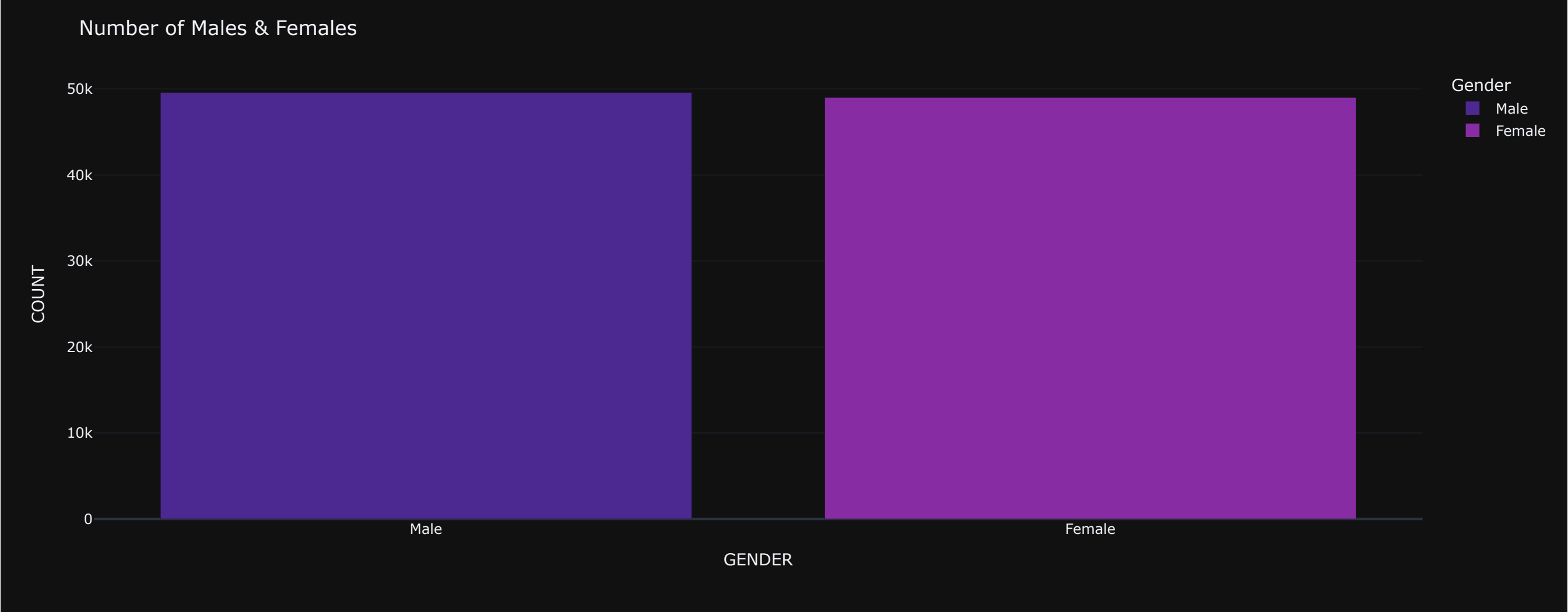
	Gender	count
0	Male	49598
1	Female	49021

In [30]:

```
# We have total no. of Males = 49598 & total no. of Females = 49021
```

In [32]:

```
# Visualization
fig=px.bar(data, x='Gender',y='count',color='Gender',color_discrete_sequence=px.colors.sequential.Agsunset,template = 'plotly_dark')
fig.update_layout(title_text='Number of Males & Females', xaxis_title='GENDER', yaxis_title='COUNT')
fig.show()
```

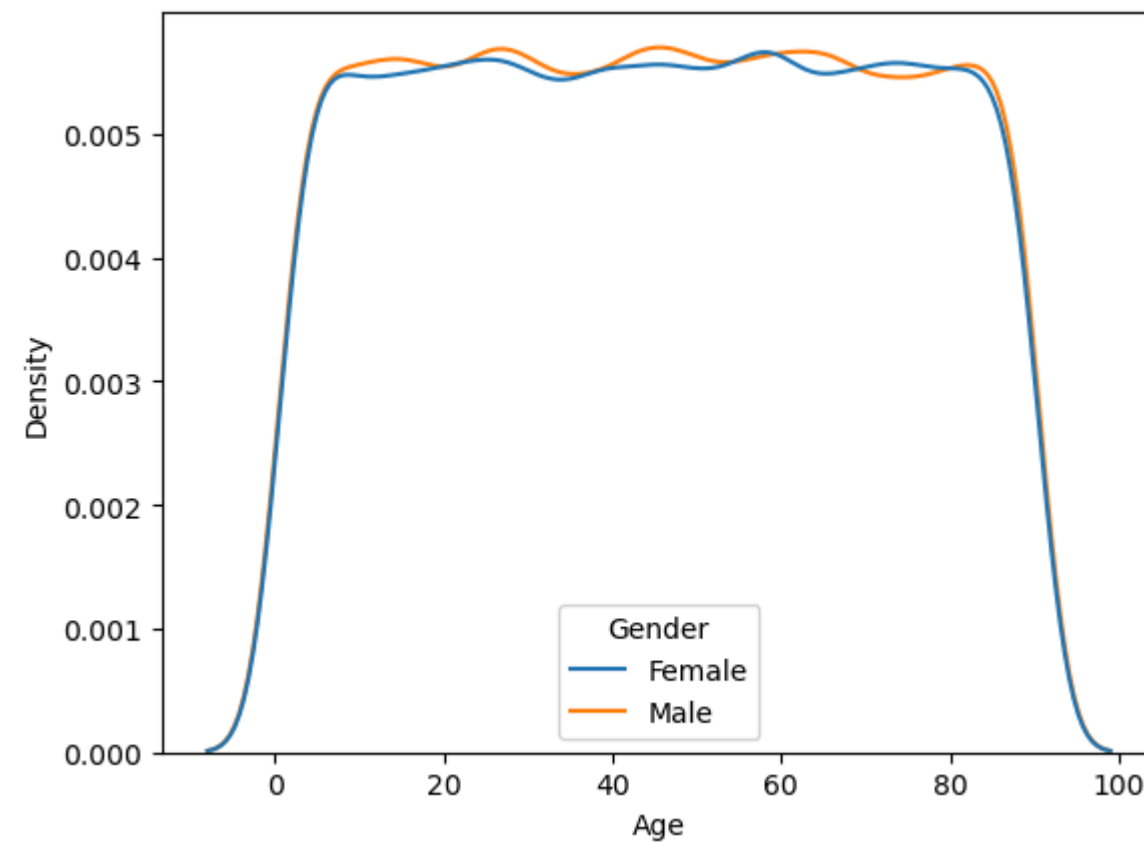


```
In [34]: #Number of males and females in the dataset are comparable
```

```
In [36]: # Age
```

```
In [38]: from seaborn import kdeplot
kdeplot(data=df, x='Age', hue='Gender')
```

Out[38]: <Axes: xlabel='Age', ylabel='Density'>



```
In [40]: # A Kernel Density Estimate (KDE) plot is a statistical tool used to visualize the probability density of a continuous variable.
```

```
In [42]: # Nationality
```

```
In [44]: df['Nationality'].unique()
```

```
Out[44]: array(['Japan', 'Nicaragua', 'Russia', 'China', 'Brazil', 'Ivory Coast',
               'Vietnam', 'Palestinian Territory', 'Thailand', 'Tunisia',
               'Sweden', 'Colombia', 'Greece', 'Philippines', 'Ukraine',
               'Indonesia', 'Croatia', 'Democratic Republic of the Congo',
               'France', 'Peru', 'Latvia', 'Germany', 'Czech Republic', 'Ireland',
               'Iran', 'Madagascar', 'United States', 'Ghana', 'Mali',
               'Guatemala', 'Kyrgyzstan', 'Haiti', 'Afghanistan', 'Ethiopia',
               'Tanzania', 'Nigeria', 'Cameroon', 'Morocco', 'Armenia', 'Mexico',
               'Chile', 'Albania', 'Luxembourg', 'Honduras', 'Portugal',
               'Kazakhstan', 'Pakistan', 'Poland', 'South Africa', 'Kenya',
               'Serbia', 'Maldives', 'Argentina', 'Uruguay', 'Netherlands',
               'Botswana', 'Egypt', 'Syria', 'Myanmar', 'Slovenia', 'Spain',
               'Macedonia', 'Mongolia', 'Yemen', 'Benin', 'Laos', 'Norway',
               'Sao Tome and Principe', 'Venezuela', 'United Kingdom', 'Denmark',
               'Dominican Republic', 'North Korea', 'Bangladesh', 'Mozambique',
               'Bosnia and Herzegovina', 'Canada', 'Lithuania', 'Sri Lanka',
               'Belarus', 'Niger', 'Uzbekistan', 'Malaysia', 'Sierra Leone',
               'Moldova', 'Switzerland', 'Nepal', 'Belgium', 'Iraq', 'Finland',
               'Uganda', 'Azerbaijan', 'East Timor', 'Marshall Islands',
               'Georgia', 'Gabon', 'New Caledonia', 'South Korea',
               'Central African Republic', 'Guinea-Bissau', 'Paraguay',
               'Suriname', 'Jamaica', 'Saint Lucia', 'Israel', 'Zambia',
               'Turkmenistan', 'Burkina Faso', 'Jordan', 'New Zealand', 'Libya',
               'Mauritius', 'Somalia', 'Kiribati', 'Ecuador', 'Namibia',
               'Estonia', 'Guam', 'Costa Rica', 'Palau', 'Angola', 'Austria',
               'Tajikistan', 'Oman', 'Cyprus', 'French Guiana', 'Sudan',
               'Montenegro', 'Lebanon', 'Australia', 'Republic of the Congo',
               'Cuba', 'Malta', 'Saudi Arabia', 'Italy', 'Romania', 'Dominica',
               'Bhutan', 'Western Sahara', 'Guinea', 'Panama', 'Bolivia', 'Samoa',
               'Comoros', 'Kosovo', 'Bulgaria', 'Cambodia', 'Liechtenstein',
               'Bahamas', 'Kuwait', 'Zimbabwe', 'Hungary', 'Taiwan', 'Iceland',
               'French Polynesia', 'Cape Verde', 'Lesotho', 'Cocos Islands',
               'Saint Kitts and Nevis', 'Chad', 'El Salvador', 'Bahrain',
               'Reunion', 'Christmas Island', 'Mauritania', 'Trinidad and Tobago',
               'Guadeloupe', 'Bermuda', 'Equatorial Guinea', 'Solomon Islands',
               'Senegal', 'Turkey', 'Papua New Guinea', 'Antigua and Barbuda',
               'Nauru', 'Togo', 'Slovakia', 'Guyana', 'Liberia',
               'United Arab Emirates', 'Micronesia', 'Norfolk Island', 'Qatar',
               'Gambia', 'Malawi', 'Djibouti', 'Singapore', 'Cayman Islands',
               'Anguilla', 'Aruba', 'South Sudan', 'Curacao', 'Fiji',
               'Puerto Rico', 'Hong Kong', 'Faroe Islands', 'Aland Islands',
               'Burundi', 'Eritrea', 'San Marino', 'Vanuatu', 'Mayotte', 'India',
               'Algeria', 'Andorra', 'U.S. Virgin Islands', 'Belize', 'Grenada',
               'Monaco', 'Seychelles', 'Jersey', 'Barbados',
               'Saint Vincent and the Grenadines', 'Martinique', 'American Samoa',
               'South Georgia and the South Sandwich Islands', 'Tuvalu',
               'Isle of Man', 'Niue', 'Saint Helena', 'Montserrat',
               'Cook Islands', 'Swaziland', 'Svalbard and Jan Mayen',
               'Falkland Islands', 'Saint Pierre and Miquelon',
               'Wallis and Futuna', 'French Southern Territories',
               'Bonaire, Saint Eustatius and Saba ', 'Rwanda', 'Tokelau', 'Tonga',
               'Macao', 'Saint Martin', 'Pitcairn', 'Northern Mariana Islands',
               'Greenland', 'Brunei', 'British Virgin Islands', 'Sint Maarten'],
            dtype=object)
```

```
In [48]: df['Nationality'].nunique()
# Here are 240 different Nationalities
```

```
Out[48]: 240
```

```
In [50]: nation_count=df['Nationality'].value_counts().reset_index()
```

```
In [52]: nation_count
```

Out[52]:

	Nationality	count
0	China	18317
1	Indonesia	10559
2	Russia	5693
3	Philippines	5239
4	Brazil	3791
...
235	Romania	2
236	British Virgin Islands	2
237	Jersey	1
238	Norfolk Island	1
239	Sint Maarten	1

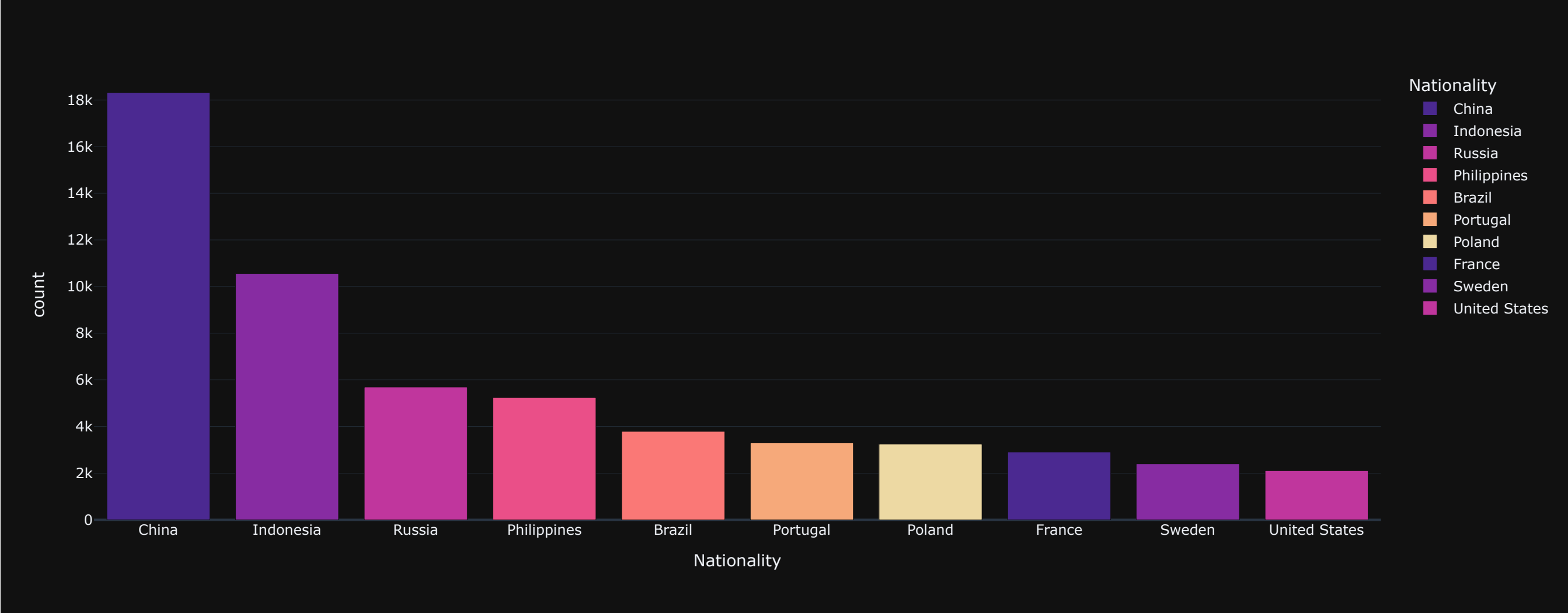
240 rows × 2 columns

```
In [54]: # Getting the top 10 Nationalities
top_10_countries=nation_count.nlargest(10, 'count')
top_10_countries.reset_index()
top_10_countries
```

Out[54]:

	Nationality	count
0	China	18317
1	Indonesia	10559
2	Russia	5693
3	Philippines	5239
4	Brazil	3791
5	Portugal	3299
6	Poland	3245
7	France	2907
8	Sweden	2397
9	United States	2105

```
In [58]: # Let's visualise top 10 Nationalities
px.bar(top_10_countries, x='Nationality', y = 'count',color='Nationality',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```

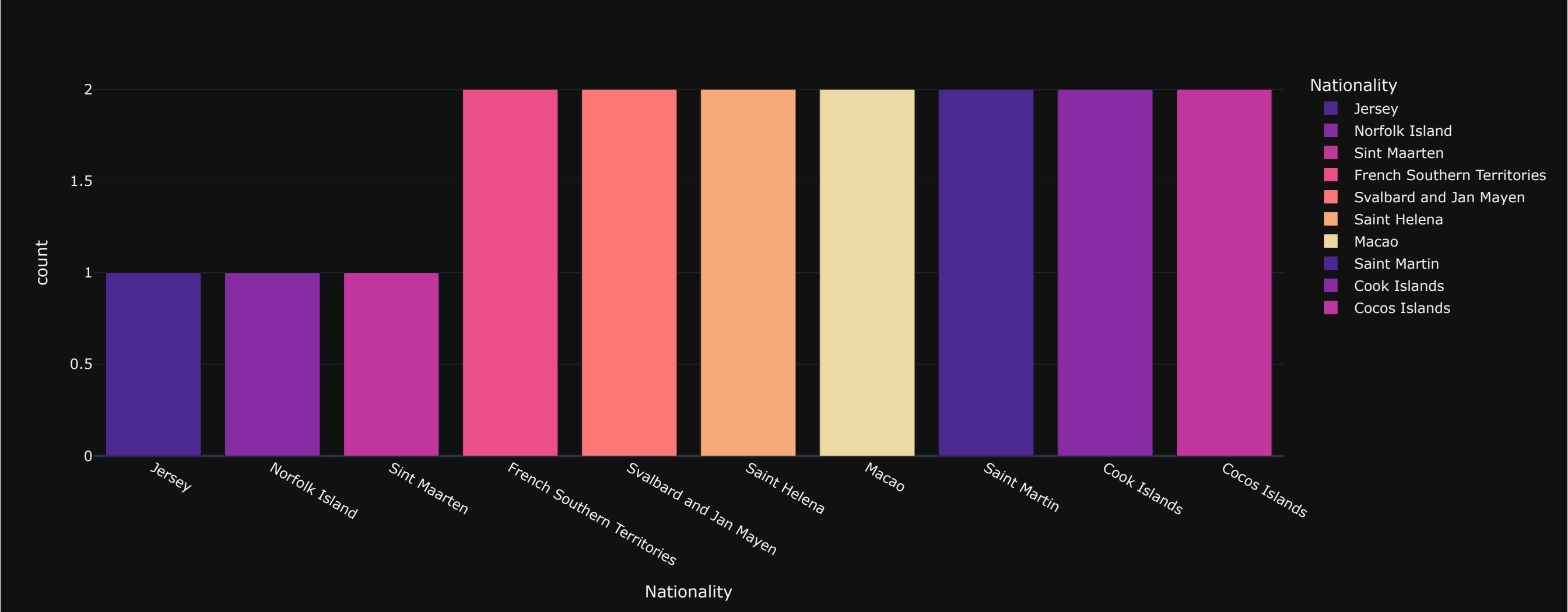



```
In [60]: # Lowest top 10 Nationalities
lowest_10_countries=nation_count.nsmallest(10,'count')
lowest_10_countries.reset_index()
lowest_10_countries
```

Out[60]:

	Nationality	count
237	Jersey	1
238	Norfolk Island	1
239	Sint Maarten	1
228	French Southern Territories	2
229	Svalbard and Jan Mayen	2
230	Saint Helena	2
231	Macao	2
232	Saint Martin	2
233	Cook Islands	2
234	Cocos Islands	2

```
In [63]: # Let;s visualise Lowest number of Passenger Nationalities
px.bar(lowest_10_countries,x='Nationality',y='count',color='Nationality',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```



```
In [66]: # Airport Name (Boarding airport)

In [68]: df['Airport Name'].unique()

Out[68]: array(['Coldfoot Airport', 'Kugluktuk Airport', 'Grenoble-Isère Airport',
..., 'Camaxilo Airport', 'Babelthuap Airport',
'Westchester County Airport'], dtype=object)

In [73]: df['Airport Name'].nunique()

# There are 9062 unique entries from where the passengers took the flight.

Out[73]: 9062

In [75]: airport_name=df['Airport Name'].value_counts().reset_index()
airport_name
```

Out[75]:

	Airport Name	count
0	San Pedro Airport	43
1	Santa Maria Airport	38
2	Böblingen Flugfeld	36
3	Santa Ana Airport	35
4	San Fernando Airport	31
...
9057	Balıkesir Merkez Airport	2
9058	Arkalyk North Airport	2
9059	Juba International Airport	2
9060	Falcon State Airport	1
9061	Hiroshima Airport	1

9062 rows × 2 columns

In [77]:

```
# Getting the top 10 airports with the highest number of passengers

top10_airport = airport_name.nlargest(10,'count')
top10_airport
```

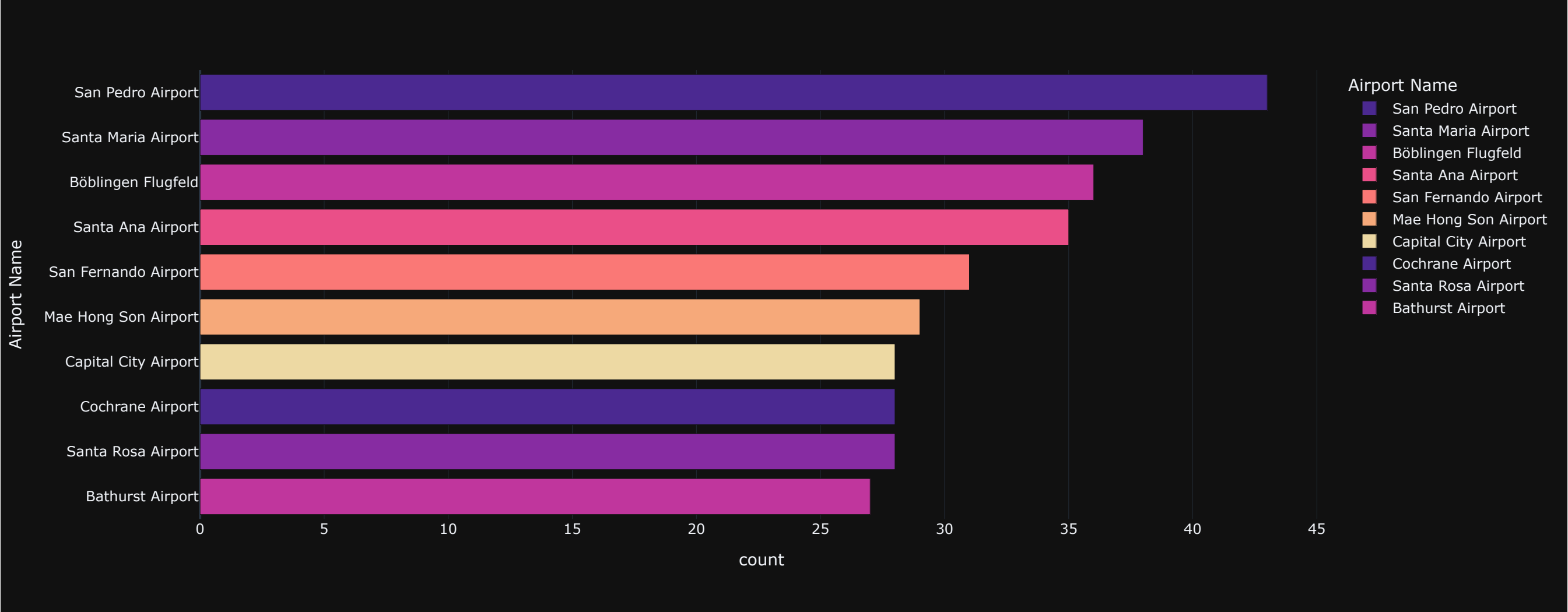
Out[77]:

	Airport Name	count
0	San Pedro Airport	43
1	Santa Maria Airport	38
2	Böblingen Flugfeld	36
3	Santa Ana Airport	35
4	San Fernando Airport	31
5	Mae Hong Son Airport	29
6	Capital City Airport	28
7	Cochrane Airport	28
8	Santa Rosa Airport	28
9	Bathurst Airport	27

In [79]:

```
# Let's plot top 10 airport

px.bar(top10_airport,x='count',y='Airport Name',color='Airport Name',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```



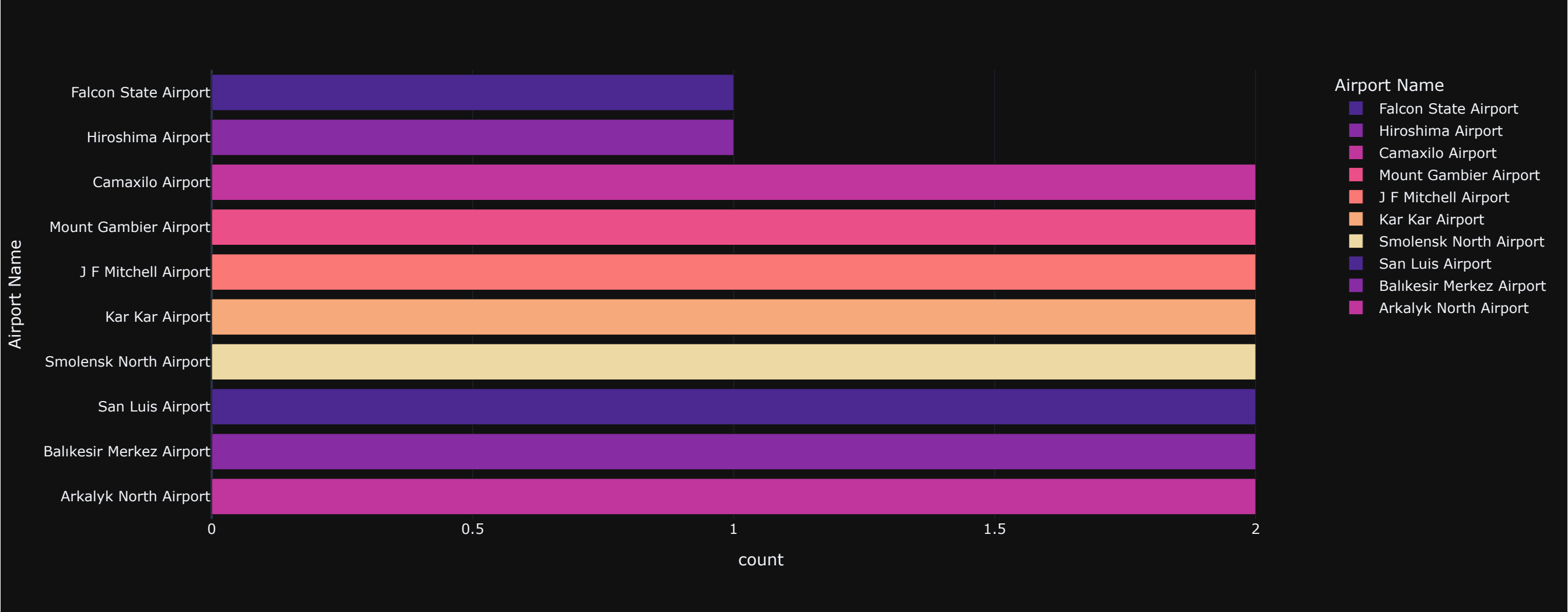
```
In [81]: # Getting the top 10 airports with the lowest number of passengers

bottom10_airport = airport_name.nsmallest(10, 'count')
bottom10_airport
```

Out[81]:

	Airport Name	count
9060	Falcon State Airport	1
9061	Hiroshima Airport	1
9051	Camaxilo Airport	2
9052	Mount Gambier Airport	2
9053	J F Mitchell Airport	2
9054	Kar Kar Airport	2
9055	Smolensk North Airport	2
9056	San Luis Airport	2
9057	Balikesir Merkez Airport	2
9058	Arkalyk North Airport	2

```
In [83]: # Let's plot lowest number of passengers
px.bar(bottom10_airport,x='count',y='Airport Name',color='Airport Name',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```



```
In [85]: # Countries
```

```
In [91]: df['Country Name'].unique()
```

```
Out[91]: array(['United States', 'Canada', 'France', 'Brazil', 'United Kingdom',
               'Italy', 'Mexico', 'Panama', 'Pakistan', 'Australia',
               'South Africa', 'Venezuela, Bolivarian Republic of', 'China',
               'Madagascar', 'Colombia', 'Mauritania', 'Myanmar', 'Congo',
               'Zambia', 'Samoa', 'New Caledonia', 'Equatorial Guinea',
               'Papua New Guinea', 'Bahamas', 'Indonesia', 'Russian Federation',
               'Bolivia, Plurinational State of', 'Kenya', 'Burkina Faso',
               'Guatemala', 'Taiwan, Province of China', 'India', 'Norway',
               'Costa Rica', 'Portugal', 'Sri Lanka', 'French Polynesia',
               'Mozambique', 'Algeria', 'Nepal', 'Bulgaria',
               'Iran, Islamic Republic of', 'Honduras', 'Sudan', 'Japan',
               'Dominican Republic', 'Iceland', 'Viet Nam',
               'Syrian Arab Republic', 'Romania', 'Cayman Islands', 'Germany',
               'Slovakia', 'Botswana', 'Saudi Arabia', 'Ecuador', 'Cameroon',
               'Mongolia', 'Dominica', 'Paraguay', 'Israel', 'Croatia',
               'Argentina', 'Vanuatu', 'New Zealand',
               'Congo, The Democratic Republic of the', 'Ukraine', 'Sweden',
               'Nicaragua', 'Malaysia', 'Ethiopia', 'Poland', 'Cuba', 'Spain',
               'Cambodia', 'Oman', 'Peru', 'Guyana', 'Uruguay', 'Morocco',
               'Estonia', 'Kuwait', 'Greece', 'Senegal', 'Gabon', 'Somalia',
               "Côte d'Ivoire", 'Falkland Islands (Malvinas)', 'Solomon Islands',
               'Zimbabwe', 'Grenada', 'Chile', 'Virgin Islands, U.S.',
               'Marshall Islands', 'Jamaica', 'Philippines', 'Suriname',
               'Puerto Rico', 'Wallis and Futuna', 'Tunisia', 'Turkey', 'Hungary',
               'Tanzania, United Republic of', 'Slovenia', 'Tonga', 'Angola',
               'Afghanistan', 'Lesotho', 'Egypt', 'Libya', 'Malawi', 'Cabo Verde',
               'Netherlands', 'Macao', 'Liberia', 'Kiribati', 'Thailand',
               "Lao People's Democratic Republic", 'Northern Mariana Islands',
               'Switzerland', 'Austria', 'Togo', 'Azerbaijan', 'Norfolk Island',
               'Yemen', 'Jordan', 'Nigeria', 'Belize', 'Uganda', 'Rwanda',
               'Maldives', 'Korea, Republic of', 'Faroe Islands', 'Lithuania',
               'Micronesia, Federated States of', 'Finland',
               'United States Minor Outlying Islands', 'Eritrea', 'Timor-Leste',
               'Fiji', 'Western Sahara', 'Denmark', 'Bangladesh', 'Belarus',
               'Turkmenistan', 'Saint Martin (French part)',
               'Palestine, State of', 'Cyprus', 'Tuvalu', 'Czechia',
               'Central African Republic', 'Nauru', 'Antigua and Barbuda',
               'South Sudan', 'Greenland', 'Djibouti', 'Belgium', 'Iraq',
               'Sierra Leone', 'Saint Lucia', 'Malta', 'Chad', 'North Macedonia',
               'Moldova, Republic of', 'Bosnia and Herzegovina', 'Uzbekistan',
               'Burundi', 'Qatar', 'United Arab Emirates', 'Réunion',
               'Kazakhstan', 'Saint Helena, Ascension and Tristan da Cunha',
               'Ireland', 'Turks and Caicos Islands', 'Saint Kitts and Nevis',
               'Sao Tome and Principe', 'Cook Islands', 'Guinea', 'Eswatini',
               'Guadeloupe', 'Kyrgyzstan', 'Bermuda', 'French Guiana',
               'American Samoa', 'Hong Kong', 'Virgin Islands, British', 'Benin',
               'Haiti', 'Guinea-Bissau', 'Armenia', 'El Salvador', 'Guam', 'Mali',
               'Singapore', 'Cocos (Keeling) Islands', 'Gambia', 'Montenegro',
               'Niger', 'Bonaire, Sint Eustatius and Saba', 'Luxembourg',
               'Saint Vincent and the Grenadines', 'Comoros', 'Saint Barthélemy',
               'Tajikistan', 'Seychelles', 'Ghana', 'Mauritius', 'Niue', 'Bhutan',
               'Latvia', 'Trinidad and Tobago', 'Anguilla', 'Isle of Man',
               'Lebanon', 'Serbia', 'Guernsey', 'Georgia', 'Christmas Island',
               'Saint Pierre and Miquelon', 'Mayotte', 'Brunei Darussalam',
               'Jersey', "Korea, Democratic People's Republic of", 'Gibraltar',
               'Andorra', 'Bahrain', 'Barbados', 'Sint Maarten (Dutch part)',
               'British Indian Ocean Territory', 'Martinique', 'Albania',
               'Montserrat', 'Monaco', 'Aruba', 'Palau'], dtype=object)
```

```
In [93]: df['Country Name'].nunique()
```

Out[93]: 235

```
In [95]: country=df['Country Name'].value_counts().reset_index()  
country
```

Out[95]:

	Country Name	count
0	United States	22104
1	Australia	6370
2	Canada	5424
3	Brazil	4504
4	Papua New Guinea	4081
...
230	Andorra	7
231	Monaco	7
232	Nauru	7
233	Palau	5
234	Jersey	4

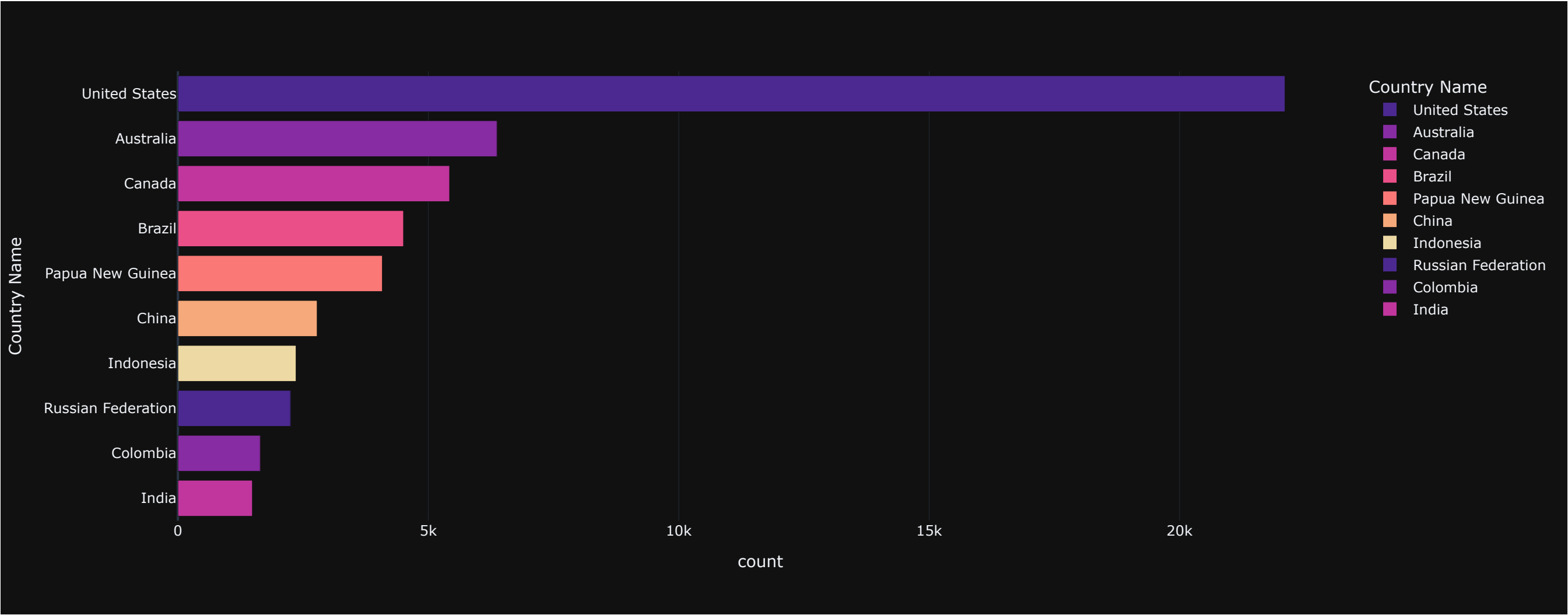
235 rows × 2 columns

```
In [99]: # Top 10 countries  
  
top10_country=country.nlargest(10, 'count')  
top10_country
```

Out[99]:

	Country Name	count
0	United States	22104
1	Australia	6370
2	Canada	5424
3	Brazil	4504
4	Papua New Guinea	4081
5	China	2779
6	Indonesia	2358
7	Russian Federation	2247
8	Colombia	1643
9	India	1486

```
In [101... # Let's plot this
px.bar(top10_country,x='count',y='Country Name',color='Country Name',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```



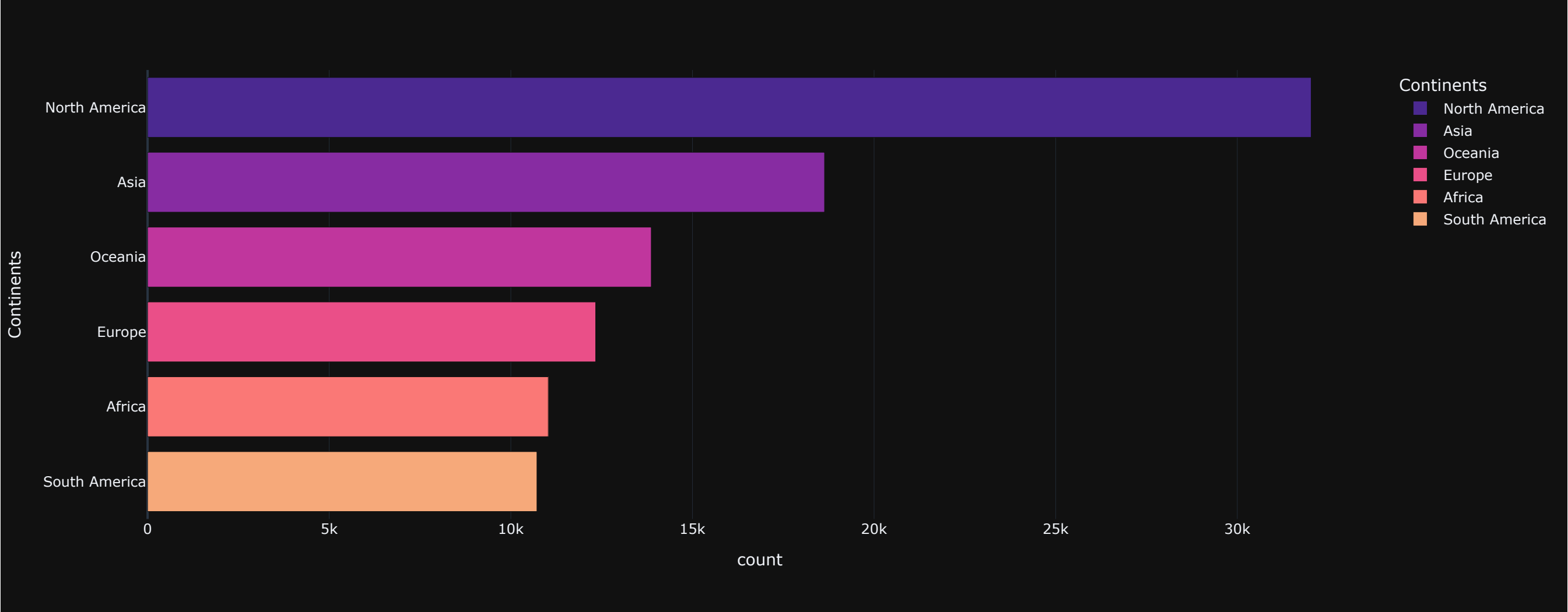
```
In [103... # Continents
```

```
In [105... continent=df['Continents'].value_counts().reset_index()
continent
```

Out[105...

	Continents	count
0	North America	32033
1	Asia	18637
2	Oceania	13866
3	Europe	12335
4	Africa	11030
5	South America	10718

```
In [108... # Plotting the continents passenger count
px.bar(continent,x='count',y='Continents',color='Continents',color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```

```
In [110... # Let's Look at the dataset again
```

```
In [112... df.head(10)
```

Out[112...

	Gender	Age	Nationality	Airport Name	Airport Country Code	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Pilot Name	Flight Status
0	Female	62	Japan	Coldfoot Airport	US	United States	NAM	North America	6/28/2022	CXF	Fransisco Hazeldine	On Time
1	Male	62	Nicaragua	Kugluktuk Airport	CA	Canada	NAM	North America	12/26/2022	YCO	Marla Parsonage	On Time
2	Male	67	Russia	Grenoble-Isère Airport	FR	France	EU	Europe	1/18/2022	GNB	Rhonda Amber	On Time
3	Female	71	China	Ottawa / Gatineau Airport	CA	Canada	NAM	North America	9/16/2022	YND	Kacie Commucci	Delayed
4	Male	21	China	Gillespie Field	US	United States	NAM	North America	2/25/2022	SEE	Ebonee Tree	On Time
5	Female	55	Brazil	Coronel Horácio de Mattos Airport	BR	Brazil	SAM	South America	06-10-2022	LEC	Inglis Dolley	On Time
6	Male	73	Ivory Coast	Duxford Aerodrome	GB	United Kingdom	EU	Europe	10/30/2022	QFO	Stanislas Tiffin	Cancelled
7	Male	36	Vietnam	Maestro Wilson Fonseca Airport	BR	Brazil	SAM	South America	04-07-2022	STM	Sharyl Eastmead	Cancelled
8	Female	35	Palestinian Territory	Venice Marco Polo Airport	IT	Italy	EU	Europe	8/20/2022	VCE	Daryn Bardsley	On Time
9	Male	13	Thailand	Vermilion Airport	CA	Canada	NAM	North America	04-06-2022	YVG	Alameda Carlyle	On Time

```
In [114... # We can drop the country code and pilot name too:
```

```
In [116... df=df.drop(['Pilot Name', 'Airport Country Code'], axis = 1)
```

```
In [118... df.head(5)
```

Out[118...

	Gender	Age	Nationality	Airport Name	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Flight Status
0	Female	62	Japan	Coldfoot Airport	United States	NAM	North America	6/28/2022	CXF	On Time
1	Male	62	Nicaragua	Kugluktuk Airport	Canada	NAM	North America	12/26/2022	YCO	On Time
2	Male	67	Russia	Grenoble-Isère Airport	France	EU	Europe	1/18/2022	GNB	On Time
3	Female	71	China	Ottawa / Gatineau Airport	Canada	NAM	North America	9/16/2022	YND	Delayed
4	Male	21	China	Gillespie Field	United States	NAM	North America	2/25/2022	SEE	On Time

```
In [120... # Airport Name (Arrival Airports)
```

```
In [122... airport=df['Arrival Airport'].value_counts().reset_index()  
airport
```

Out[122...

	Arrival Airport	count
0		873
1	JNB	37
2	PHM	36
3	MPT	32
4	PCO	27
...
9019	CXM	2
9020	IPI	2
9021	BQU	2
9022	FAL	1
9023	HIJ	1

9024 rows × 2 columns

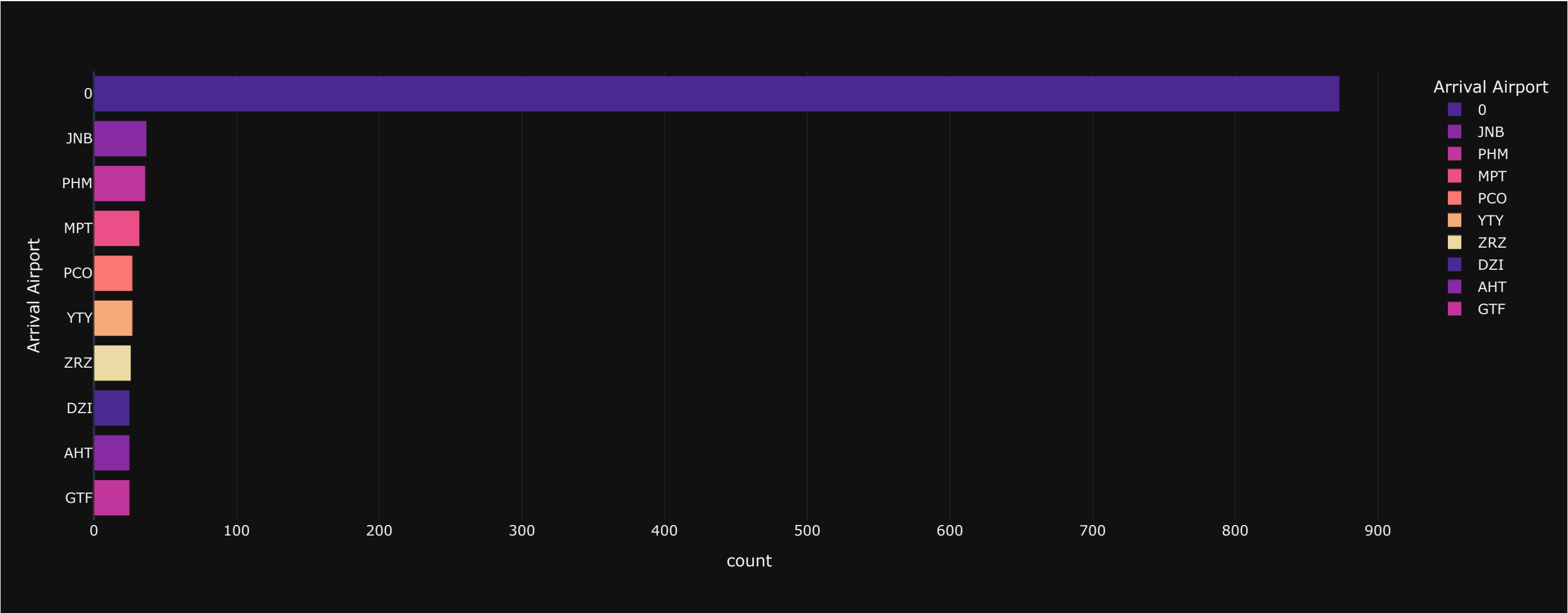
```
In [124... # Top 10 arrival airports  
top=airport.nlargest(10,'count')  
top
```

Out[124...

	Arrival Airport	count
0	0	873
1	JNB	37
2	PHM	36
3	MPT	32
4	PCO	27
5	YTY	27
6	ZRZ	26
7	DZI	25
8	AHT	25
9	GTF	25

In [127...

```
px.bar(top,x='count',y='Arrival Airport',color='Arrival Airport', color_discrete_sequence=px.colors.sequential.Agsunset ,template = 'plotly_dark')
```



In [129...

```
# Flight Status
```

In [136...

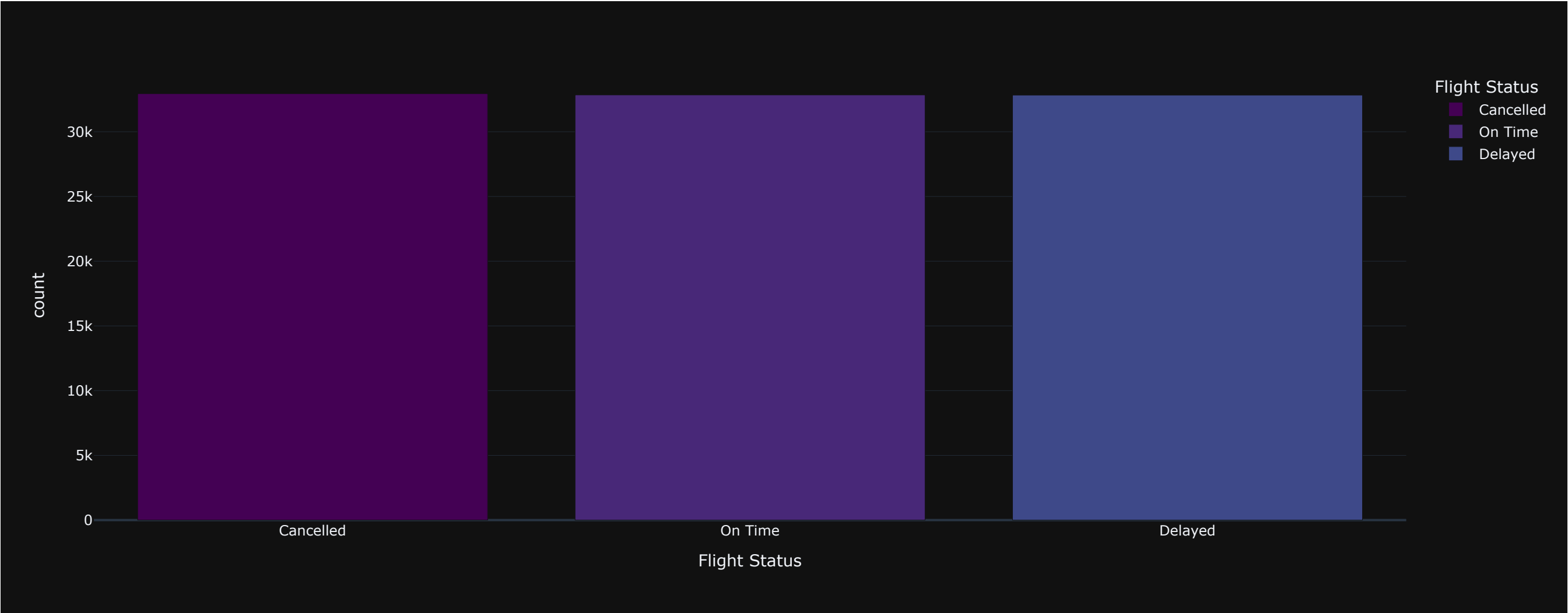
```
status=df['Flight Status'].value_counts().reset_index()  
status
```

Out[136...

	Flight Status	count
0	Cancelled	32942
1	On Time	32846
2	Delayed	32831

In [138...

```
# Visualisation of Flight Status
px.bar(status,x='Flight Status',y='count',color='Flight Status',color_discrete_sequence=px.colors.sequential.Viridis,template = 'plotly_dark')
```



In [140...

```
# The numbers are suprisingly identical
# Continent wise Delay Analysis

# North America
```

In [142...

```
N_America=df[df['Continents']=='North America']
N_America
```

Out[142...

	Gender	Age	Nationality		Airport Name	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Flight Status
0	Female	62	Japan		Coldfoot Airport	United States	NAM	North America	6/28/2022	CXF	On Time
1	Male	62	Nicaragua		Kugluktuk Airport	Canada	NAM	North America	12/26/2022	YCO	On Time
3	Female	71	China		Ottawa / Gatineau Airport	Canada	NAM	North America	9/16/2022	YND	Delayed
4	Male	21	China		Gillespie Field	United States	NAM	North America	2/25/2022	SEE	On Time
9	Male	13	Thailand		Vermilion Airport	Canada	NAM	North America	04-06-2022	YVG	On Time
...
98597	Female	38	Peru		Fort Chipewyan Airport	Canada	NAM	North America	01-10-2022	YPY	On Time
98604	Female	47	Brazil		St Cloud Regional Airport	United States	NAM	North America	02-08-2022	STC	On Time
98605	Female	7	China		Newport Municipal Airport	United States	NAM	North America	09-07-2022	ONP	On Time
98608	Female	24	Chile	Luis Munoz Marin International Airport		Puerto Rico	NAM	North America	3/23/2022	SJU	Cancelled
98612	Male	82	Indonesia		Five Mile Airport	United States	NAM	North America	3/17/2022	FMC	Delayed

32033 rows × 10 columns

In [146...

```
status_na=N_America['Flight Status'].value_counts().reset_index()
status_na
```

Out[146...

	Flight Status	count
0	Delayed	10696
1	Cancelled	10693
2	On Time	10644

In [148...

```
# Asia
```

In [150...

```
asia=df[df['Continents']=='Asia']
asia
```

Out[150...

	Gender	Age	Nationality	Airport Name	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Flight Status
12	Female	47	Sweden	Loralai Airport	Pakistan	AS	Asia	3/19/2022	LRG	Delayed
17	Male	12	Greece	Enshi Airport	China	AS	Asia	3/29/2022	ENH	Delayed
19	Male	62	China	Guilin Liangjiang International Airport	China	AS	Asia	09-05-2022	KWL	Cancelled
26	Female	14	Democratic Republic of the Congo	Bagan Airport	Myanmar	AS	Asia	10-03-2022	NYU	Delayed
42	Female	56	China	Rar Gwamar Airport	Indonesia	AS	Asia	07-08-2022	DOB	Delayed
...
98603	Male	39	Haiti	Senipah Heliport	Indonesia	AS	Asia	10-08-2022	SZH	On Time
98609	Female	22	China	Wuzhou Changzhoudao Airport	China	AS	Asia	4/23/2022	WUZ	On Time
98610	Male	83	France	Warangal Airport	India	AS	Asia	08-05-2022	WGC	Cancelled
98611	Female	41	Canada	Ipil Airport	Philippines	AS	Asia	06-06-2022	IPE	On Time
98613	Female	47	Serbia	Arugam Bay SPB	Sri Lanka	AS	Asia	05-12-2022	AYY	Delayed

18637 rows × 10 columns

In [152...

```
status_as=asia['Flight Status'].value_counts().reset_index()
status_as
```

Out[152...

	Flight Status	count
0	On Time	6242
1	Cancelled	6235
2	Delayed	6160

In [154...

```
# Oceania
```

In [156...

```
oceania=df[df['Continents']=='Oceania']
oceania
```

Out[156...

	Gender	Age	Nationality	Airport Name	Country Name	Airport Continent	Continents	Departure Date	Arrival Airport	Flight Status
13	Female	77	Russia	Cudal Airport	Australia	OC	Oceania	3/24/2022	CUG	Delayed
30	Female	14	China	Faleolo International Airport	Samoa	OC	Oceania	7/22/2022	APW	Delayed
32	Male	19	Germany	Nesson Airport	New Caledonia	OC	Oceania	3/25/2022	HLU	Cancelled
35	Male	9	Sweden	Gora Airstrip	Papua New Guinea	OC	Oceania	6/25/2022	GOC	On Time
37	Male	32	China	Moki Airport	Papua New Guinea	OC	Oceania	12/21/2022	MJJ	Cancelled
...
98538	Male	76	Indonesia	Pimaga Airport	Papua New Guinea	OC	Oceania	12/28/2022	PMP	Delayed
98567	Female	5	Finland	Meekatharra Airport	Australia	OC	Oceania	12-09-2022	MKR	Delayed
98580	Male	82	Norway	Kurwina Airport	Papua New Guinea	OC	Oceania	1/29/2022	KWV	Cancelled
98589	Female	29	Brazil	Austral Downs Airport	Australia	OC	Oceania	5/29/2022	AWP	Cancelled
98600	Male	47	Uganda	Eliptamin Airport	Papua New Guinea	OC	Oceania	6/21/2022	EPT	Delayed

13866 rows × 10 columns

In [160...

```
status_oc=asia['Flight Status'].value_counts().reset_index()
status_oc
```

Out[160...

	Flight Status	count
0	On Time	6242
1	Cancelled	6235
2	Delayed	6160

In [162...

```
# Europe
```

In [164...

```
eu=df[df['Continents']=='Europe']
status_eu=eu['Flight Status'].value_counts().reset_index()
status_eu
```

Out[164...

	Flight Status	count
0	Delayed	4178
1	Cancelled	4095
2	On Time	4062

In [166...

```
# Africa
```

In [168...

```
af=df[df['Continents']=='Africa']
status_af=af['Flight Status'].value_counts().reset_index()
status_af
```

Out[168...

	Flight Status	count
0	On Time	3719
1	Cancelled	3657
2	Delayed	3654

In [170...

```
# North America
```

In [172...

```
sa=df[df['Continents']=='South America']
status_sa=sa['Flight Status'].value_counts().reset_index()
status_sa
```

Out[172...

	Flight Status	count
0	Cancelled	3643
1	On Time	3566
2	Delayed	3509

In []:

```
###...
We observed that:
1) Most cancelled Flights are from North America.
2) Most delayed are from North America.
3) Most On Time are from North America.
4) Number of male and female passengers are identical.
5) North America as a continent and USA as a country has the highest numbers of passengers.
```