In [18]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv('Air_Traffic_Passenger_Statistics (1).csv')
df
```

Out[2]:

| | Activity Period | Operating Airline | Operating Airline IATA Code | Published Airline | Published Airline IATA Code | GEO Summary | GEO Region | Activity Type Code |
|---|---|---|---|---|---|---|---|---|
| **0** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Deplaned |
| **1** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Enplaned |
| **2** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Thru / Transit |
| **3** | 200507 | Air Canada | AC | Air Canada | AC | International | Canada | Deplaned |
| **4** | 200507 | Air Canada | AC | Air Canada | AC | International | Canada | Enplaned |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **15002** | 201603 | Virgin America | VX | Virgin America | VX | Domestic | US | Enplaned |
| **15003** | 201603 | Virgin America | VX | Virgin America | VX | International | Mexico | Deplaned |
| **15004** | 201603 | Virgin America | VX | Virgin America | VX | International | Mexico | Enplaned |
| **15005** | 201603 | Virgin Atlantic | VS | Virgin Atlantic | VS | International | Europe | Deplaned |
| **15006** | 201603 | Virgin Atlantic | VS | Virgin Atlantic | VS | International | Europe | Enplaned |

15007 rows × 16 columns

In [3]:
```
df.shape
```

Out[3]:
```
(15007, 16)
```

In [58]:
```
df.head()
```

Out[58]:

| | Activity Period | Operating Airline | Operating Airline IATA Code | Published Airline | Published Airline IATA Code | GEO Summary | GEO Region | Activity Type Code | P Categ C |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Deplaned | Low I |
| **1** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Enplaned | Low I |
| **2** | 200507 | ATA Airlines | TZ | ATA Airlines | TZ | Domestic | US | Thru / Transit | Low I |
| **3** | 200507 | Air Canada | AC | Air Canada | AC | International | Canada | Deplaned | C |
| **4** | 200507 | Air Canada | AC | Air Canada | AC | International | Canada | Enplaned | C |

In [59]:
```
df.tail()
```

Out[59]:

| | Activity Period | Operating Airline | Operating Airline IATA Code | Published Airline | Published Airline IATA Code | GEO Summary | GEO Region | Activity Type Code | ( |
|---|---|---|---|---|---|---|---|---|---|
| **15002** | 201603 | Virgin America | VX | Virgin America | VX | Domestic | US | Enplaned | I |
| **15003** | 201603 | Virgin America | VX | Virgin America | VX | International | Mexico | Deplaned | I |
| **15004** | 201603 | Virgin America | VX | Virgin America | VX | International | Mexico | Enplaned | I |
| **15005** | 201603 | Virgin Atlantic | VS | Virgin Atlantic | VS | International | Europe | Deplaned | |
| **15006** | 201603 | Virgin Atlantic | VS | Virgin Atlantic | VS | International | Europe | Enplaned | |

In [4]:
```
df.describe()
```

Out[4]:

| | Activity Period | Passenger Count | Adjusted Passenger Count | Year |
|---|---|---|---|---|
| **count** | 15007.000000 | 15007.000000 | 15007.000000 | 15007.000000 |
| **mean** | 201045.073366 | 29240.521090 | 29331.917105 | 2010.385220 |
| **std** | 313.336196 | 58319.509284 | 58284.182219 | 3.137589 |
| **min** | 200507.000000 | 1.000000 | 1.000000 | 2005.000000 |
| **25%** | 200803.000000 | 5373.500000 | 5495.500000 | 2008.000000 |
| **50%** | 201011.000000 | 9210.000000 | 9354.000000 | 2010.000000 |
| **75%** | 201308.000000 | 21158.500000 | 21182.000000 | 2013.000000 |
| **max** | 201603.000000 | 659837.000000 | 659837.000000 | 2016.000000 |

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15007 entries, 0 to 15006
Data columns (total 16 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   Activity Period               15007 non-null  int64 
 1   Operating Airline             15007 non-null  object
 2   Operating Airline IATA Code   14953 non-null  object
 3   Published Airline             15007 non-null  object
 4   Published Airline IATA Code   14953 non-null  object
 5   GEO Summary                   15007 non-null  object
 6   GEO Region                    15007 non-null  object
 7   Activity Type Code            15007 non-null  object
 8   Price Category Code           15007 non-null  object
 9   Terminal                      15007 non-null  object
 10  Boarding Area                 15007 non-null  object
 11  Passenger Count               15007 non-null  int64 
 12  Adjusted Activity Type Code   15007 non-null  object
 13  Adjusted Passenger Count      15007 non-null  int64 
 14  Year                          15007 non-null  int64 
 15  Month                         15007 non-null  object
dtypes: int64(4), object(12)
memory usage: 1.8+ MB
```

In [62]:
```
df.sample()
```

Out[62]:

|  | Activity Period | Operating Airline | Operating Airline IATA Code | Published Airline | Published Airline IATA Code | GEO Summary | GEO Region | Activity Type Code |
|---|---|---|---|---|---|---|---|---|
| **12348** | 201405 | Swiss International | LX | Swiss International | LX | International | Europe | Deplanec |

In [63]:
```
c=df.columns
c
```

Out[63]:
```
Index(['Activity Period', 'Operating Airline', 'Operating Airline IATA Cod
e',
       'Published Airline', 'Published Airline IATA Code', 'GEO Summary',
       'GEO Region', 'Activity Type Code', 'Price Category Code', 'Termina
l',
       'Boarding Area', 'Passenger Count', 'Adjusted Activity Type Code',
       'Adjusted Passenger Count', 'Year', 'Month'],
      dtype='object')
```

In [10]:
```
df.isnull()
```

Out[10]:

| | Activity Period | Operating Airline | Operating Airline IATA Code | Published Airline | Published Airline IATA Code | GEO Summary | GEO Region | Activity Type Code | Cate |
|---|---|---|---|---|---|---|---|---|---|
| **0** | False | False | False | False | False | False | False | False | |
| **1** | False | False | False | False | False | False | False | False | |
| **2** | False | False | False | False | False | False | False | False | |
| **3** | False | False | False | False | False | False | False | False | |
| **4** | False | False | False | False | False | False | False | False | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **15002** | False | False | False | False | False | False | False | False | |
| **15003** | False | False | False | False | False | False | False | False | |
| **15004** | False | False | False | False | False | False | False | False | |
| **15005** | False | False | False | False | False | False | False | False | |
| **15006** | False | False | False | False | False | False | False | False | |

15007 rows × 16 columns

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Activity Period                 0
Operating Airline               0
Operating Airline IATA Code    54
Published Airline               0
Published Airline IATA Code    54
GEO Summary                     0
GEO Region                      0
Activity Type Code              0
Price Category Code             0
Terminal                        0
Boarding Area                   0
Passenger Count                 0
Adjusted Activity Type Code     0
Adjusted Passenger Count        0
Year                            0
Month                           0
dtype: int64
```

In [24]:
```
df.fillna(0,inplace=True)
```

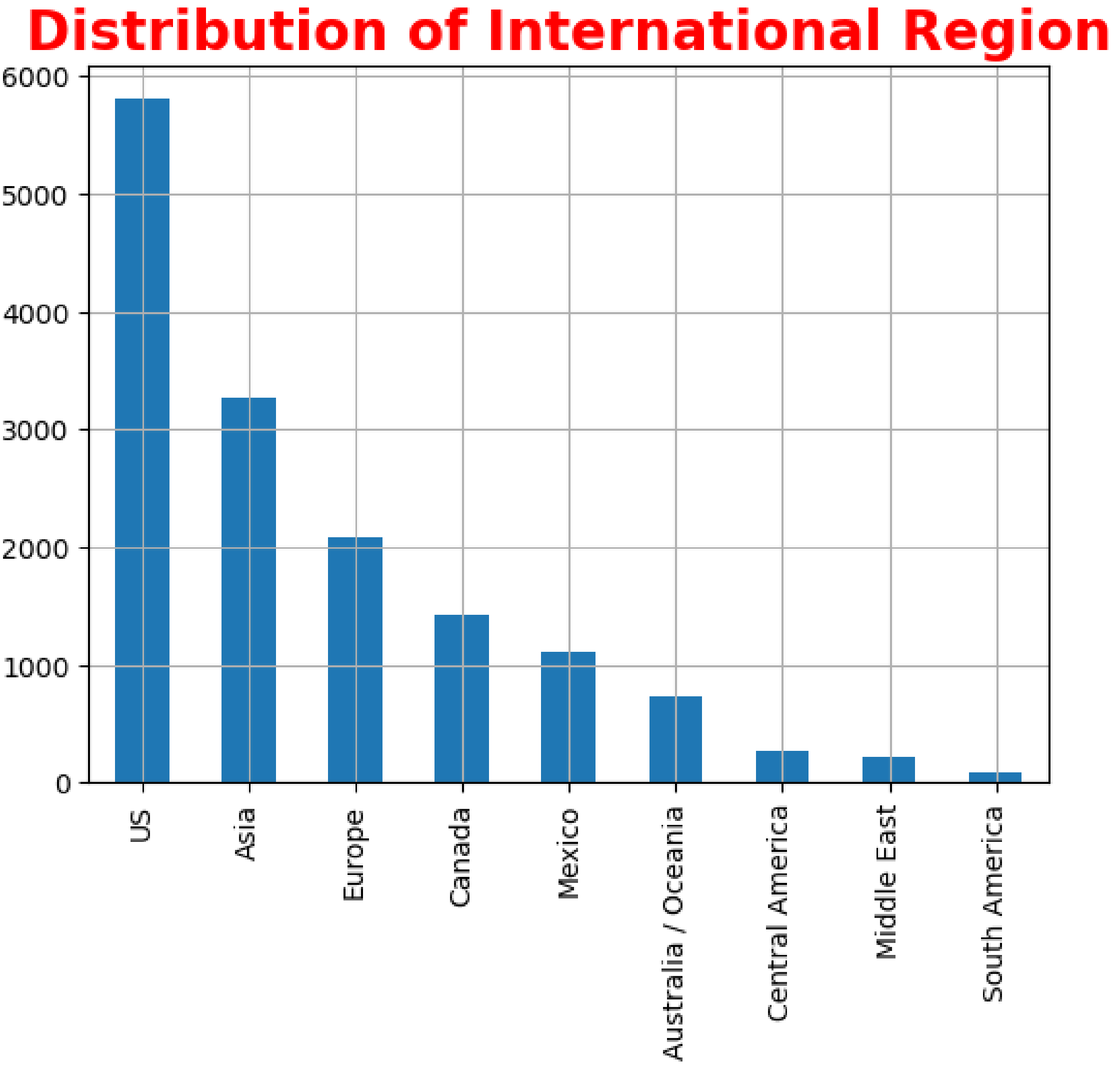
In [25]:
```
df.isnull().sum()
```

Out[25]:
```
Activity Period                0
Operating Airline              0
Operating Airline IATA Code    0
Published Airline              0
Published Airline IATA Code    0
GEO Summary                    0
GEO Region                     0
Activity Type Code             0
Price Category Code            0
Terminal                       0
Boarding Area                  0
Passenger Count                0
Adjusted Activity Type Code    0
Adjusted Passenger Count       0
Year                           0
Month                          0
dtype: int64
```

In [45]:
```
df['Region_Prossed']=df['GEO Region']
df['Region_Prossed']=df['GEO Region'].replace(['Mexico",Canada'],'North Amer
```

In [10]:

```
df['GEO Region'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Distribution of International Region',fontsize=20,fontweight='bo
plt.show()
```



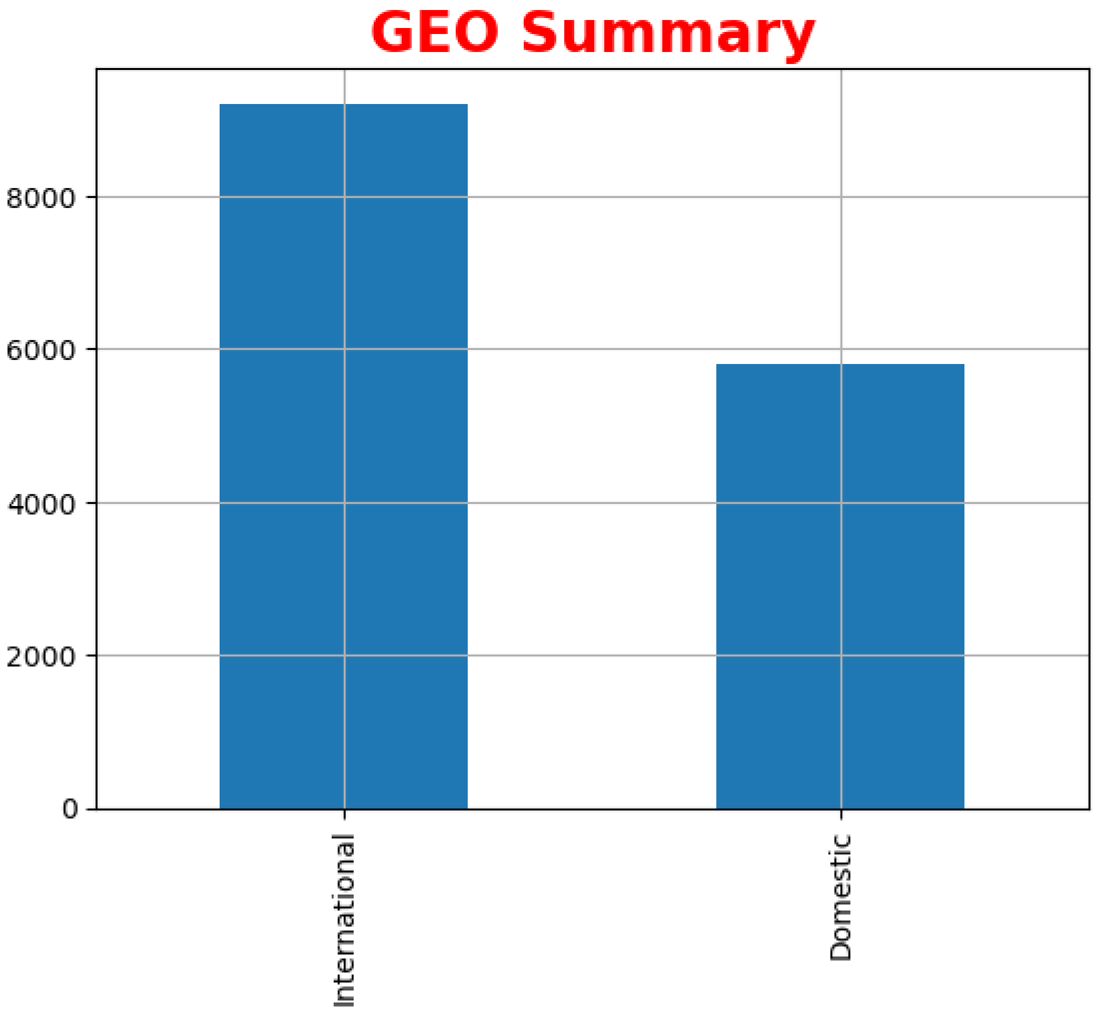
In [14]:

```
df['GEO Summary'].value_counts()
```

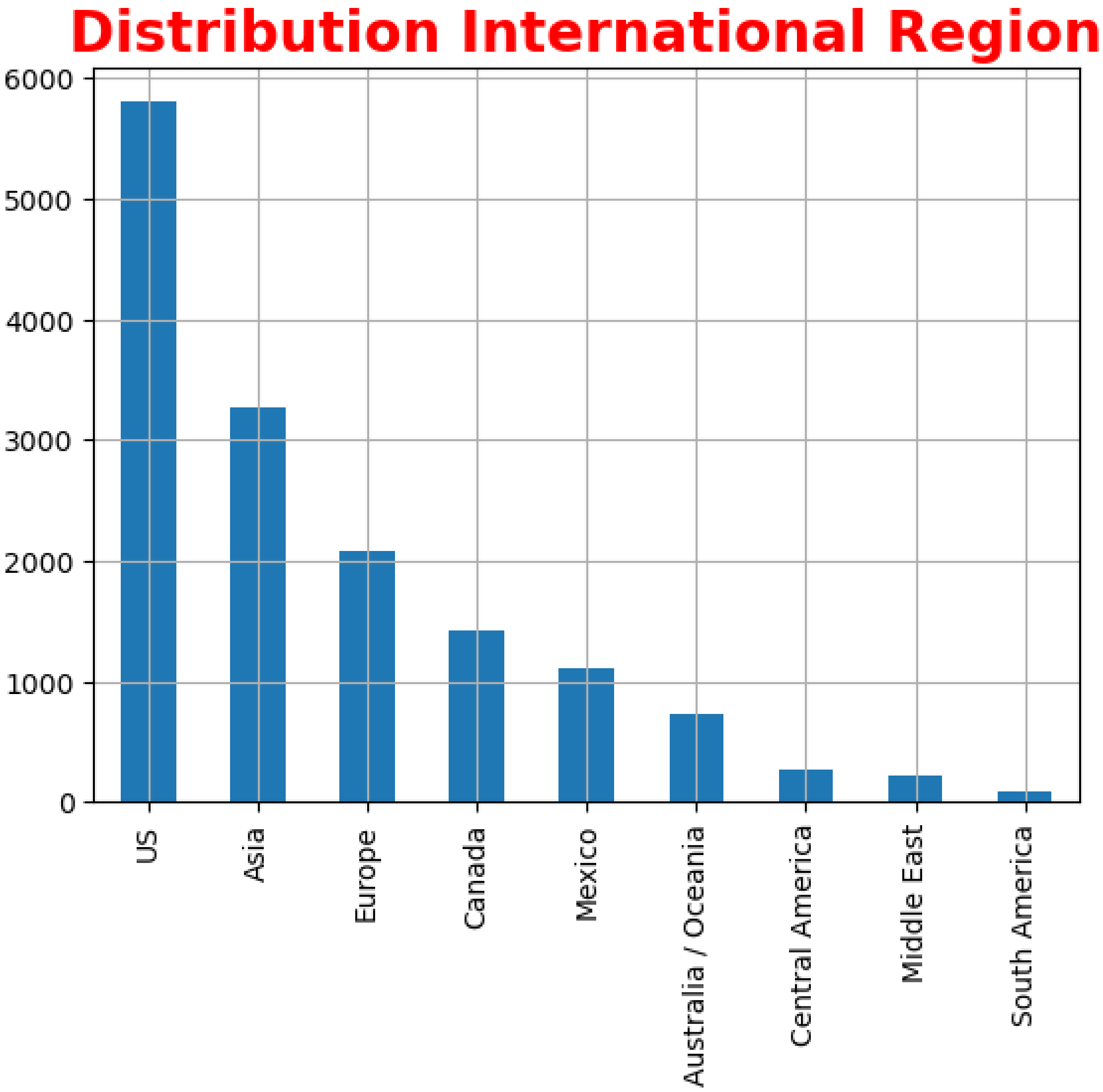
Out[14]:

```
International    9210
Domestic         5797
Name: GEO Summary, dtype: int64
```

```
In [25]: df['GEO Summary'].value_counts().plot(kind='bar')
         plt.grid()
         plt.title('GEO Summary',fontsize=20,fontweight='bold',c='red')
         plt.show()
```

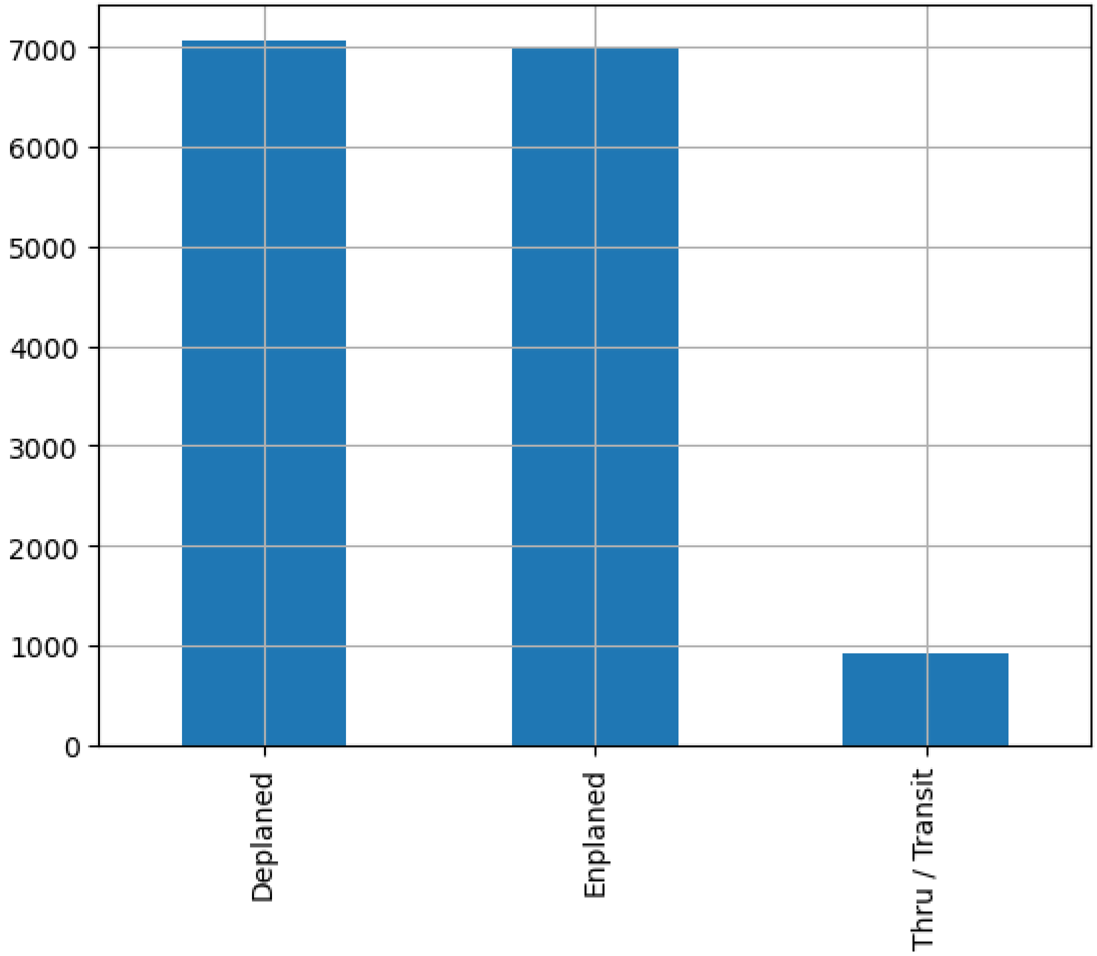


In [12]:
```
df['GEO Region'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Distribution International Region',fontsize=20,fontweight='bold',
plt.show()
```

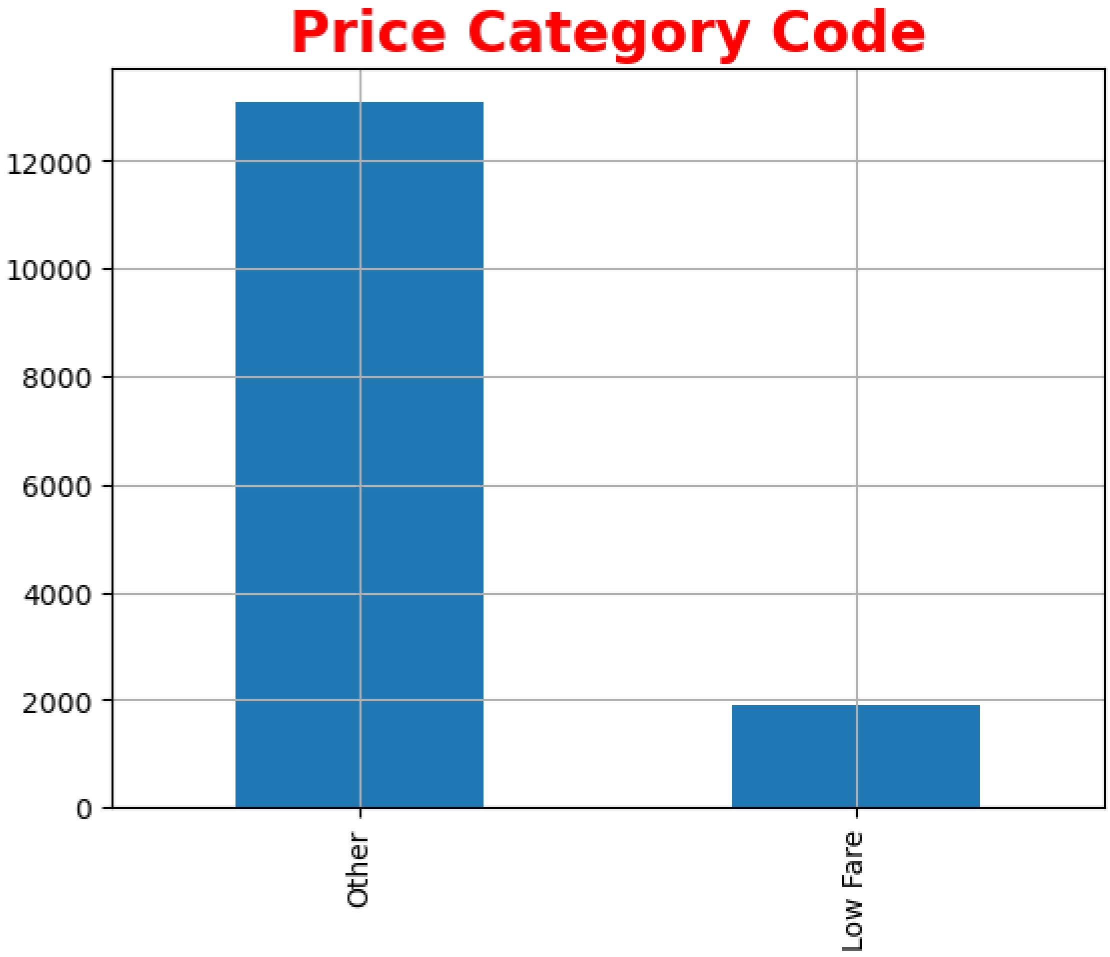


In [27]:

```
df['Activity Type Code'].value_counts().plot(kind='bar')
plt.grid()

plt.show()
```

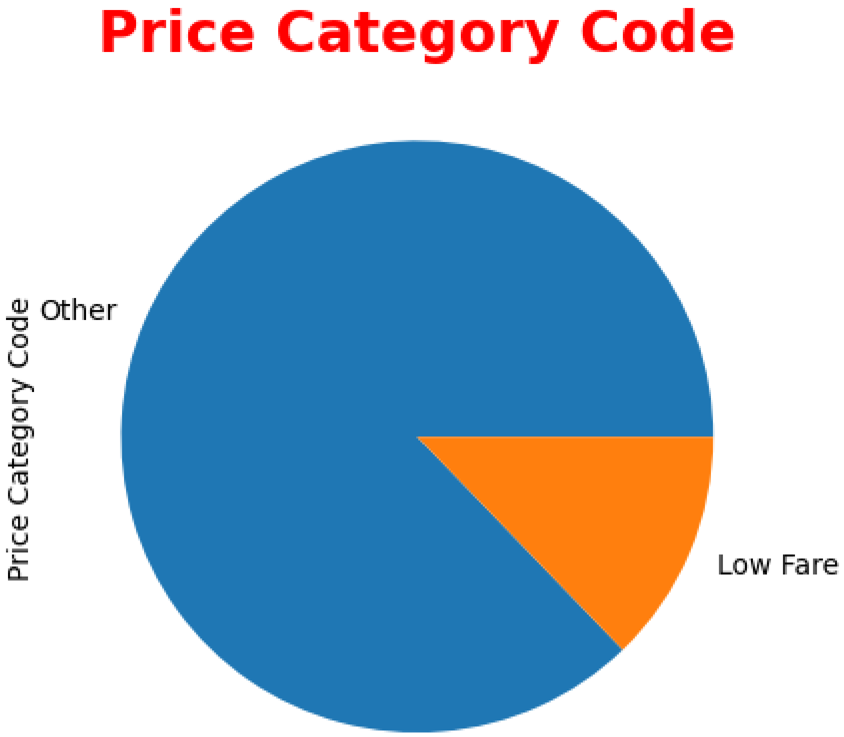


In [28]:
```
df['Price Category Code'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Price Category Code',fontsize=20,fontweight='bold',c='red')
plt.show()
```

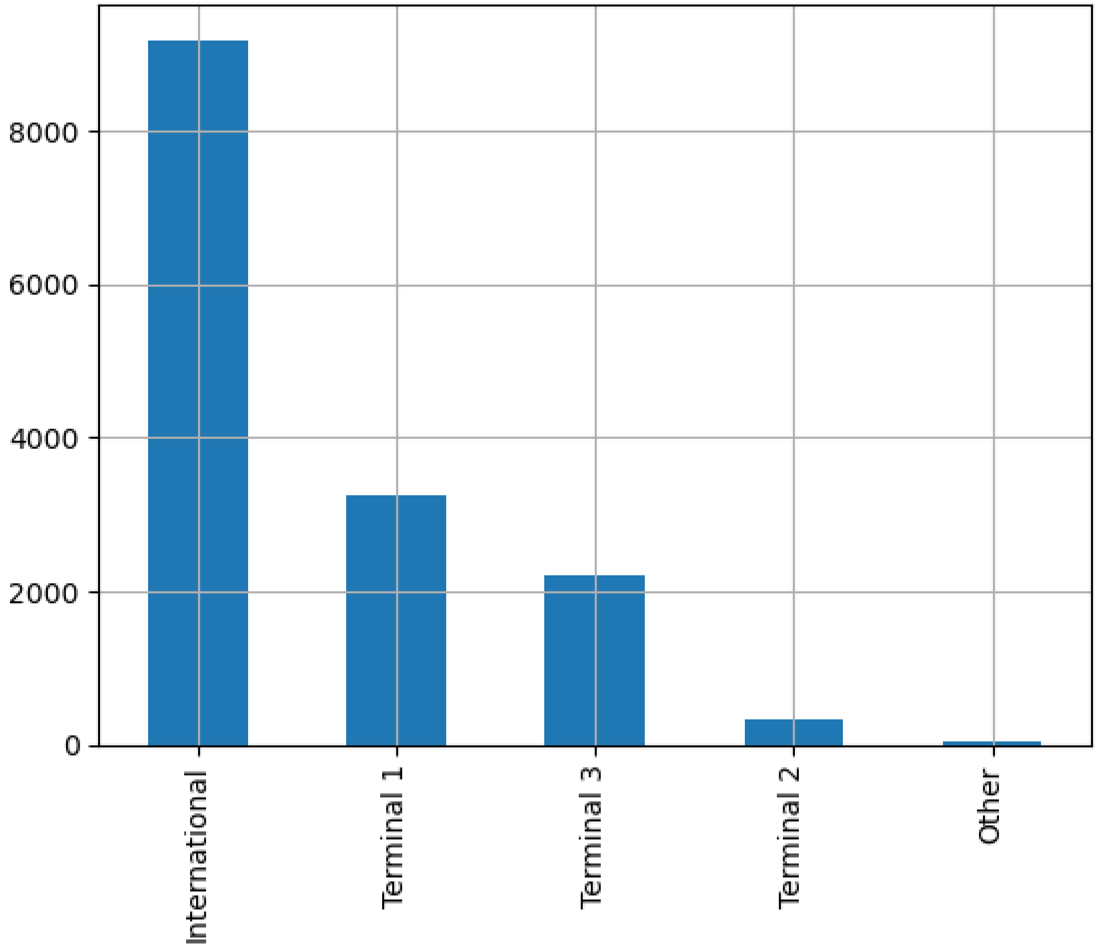


In [21]:

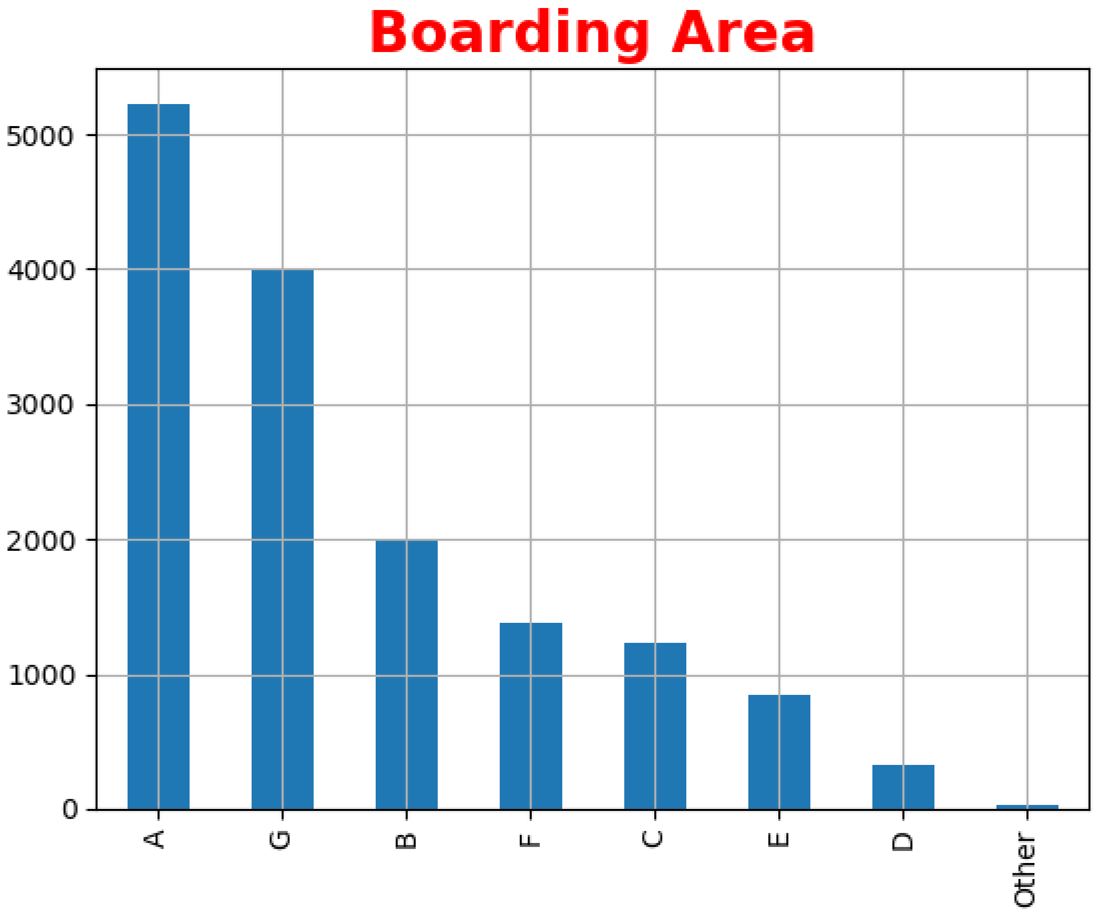
```
df['Price Category Code'].value_counts().plot(kind='pie')
plt.grid()
plt.title('Price Category Code',fontsize=20,fontweight='bold',c='red')
plt.show()
```



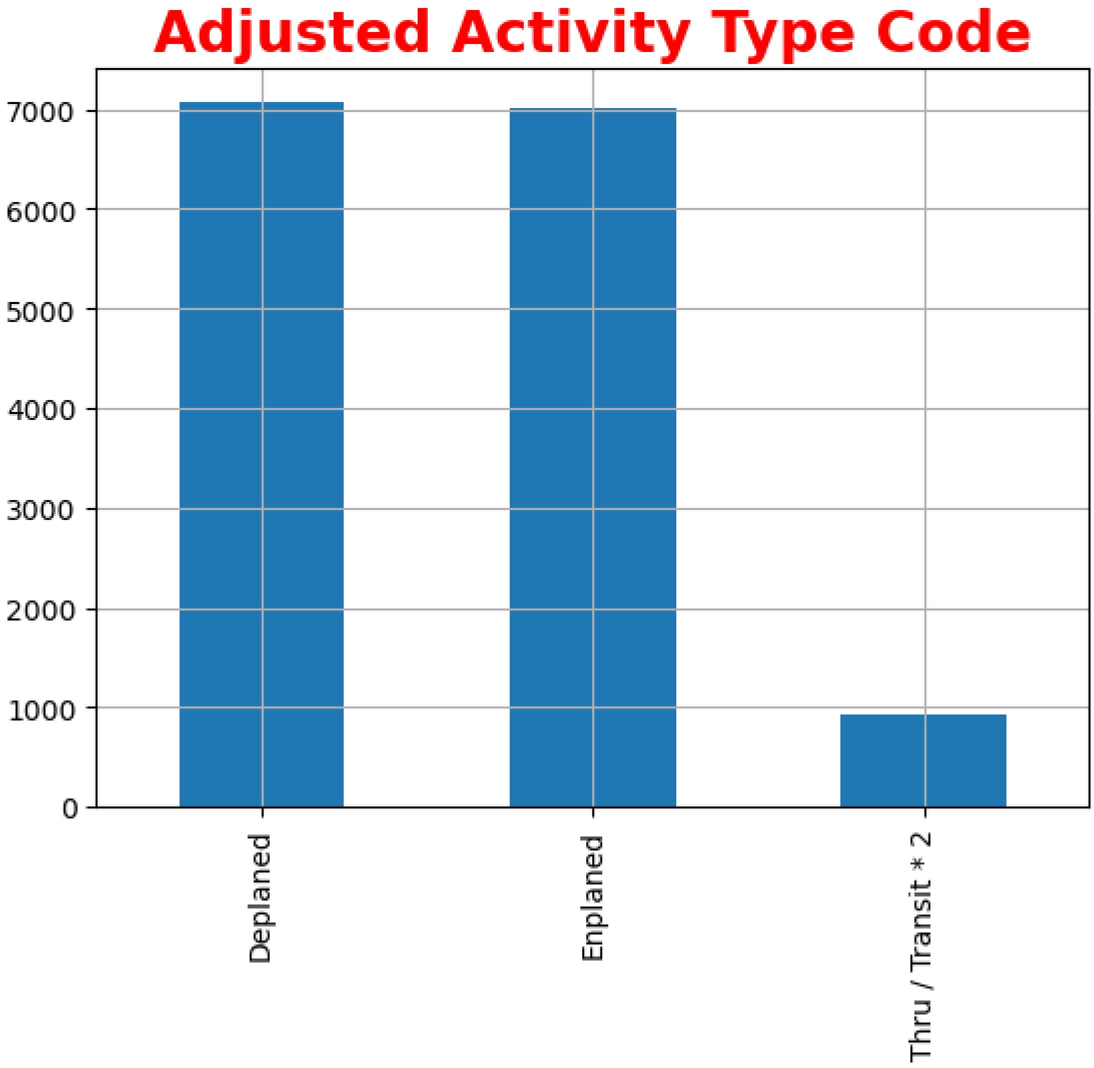
```
In [16]: df['Terminal'].value_counts().plot(kind='bar')
         plt.grid()
         plt.show()
```



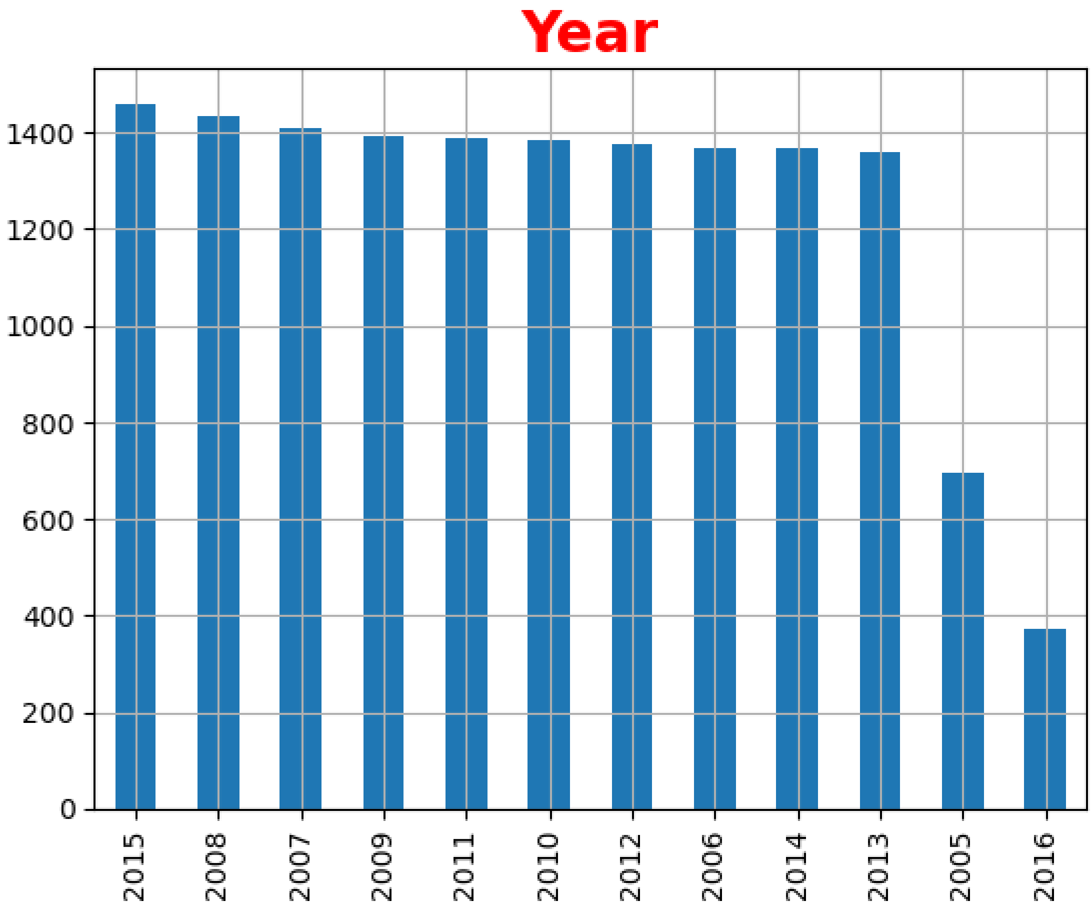
In [29]:
```
df['Boarding Area'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Boarding Area',fontsize=20,fontweight='bold',c='red')
plt.show()
```



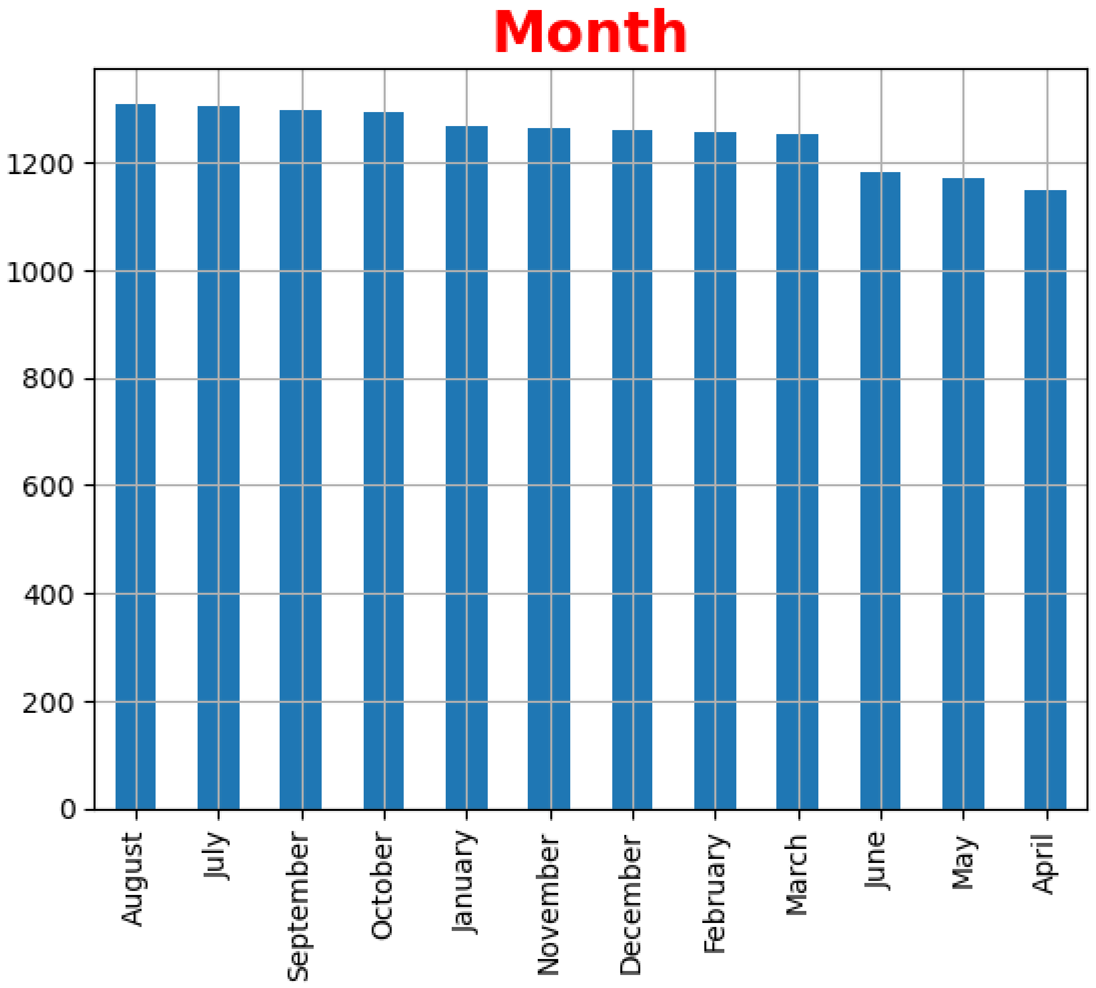
```
In [32]: df['Adjusted Activity Type Code'].value_counts().plot(kind='bar')
         plt.grid()
         plt.title('Adjusted Activity Type Code',fontsize=20,fontweight='bold',c='re
         plt.show()
```



In [31]:
```
df['Year'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Year',fontsize=20,fontweight='bold',c='red')
plt.show()
```



In [30]:
```
df['Month'].value_counts().plot(kind='bar')
plt.grid()
plt.title('Month',fontsize=20,fontweight='bold',c='red')
plt.show()
```

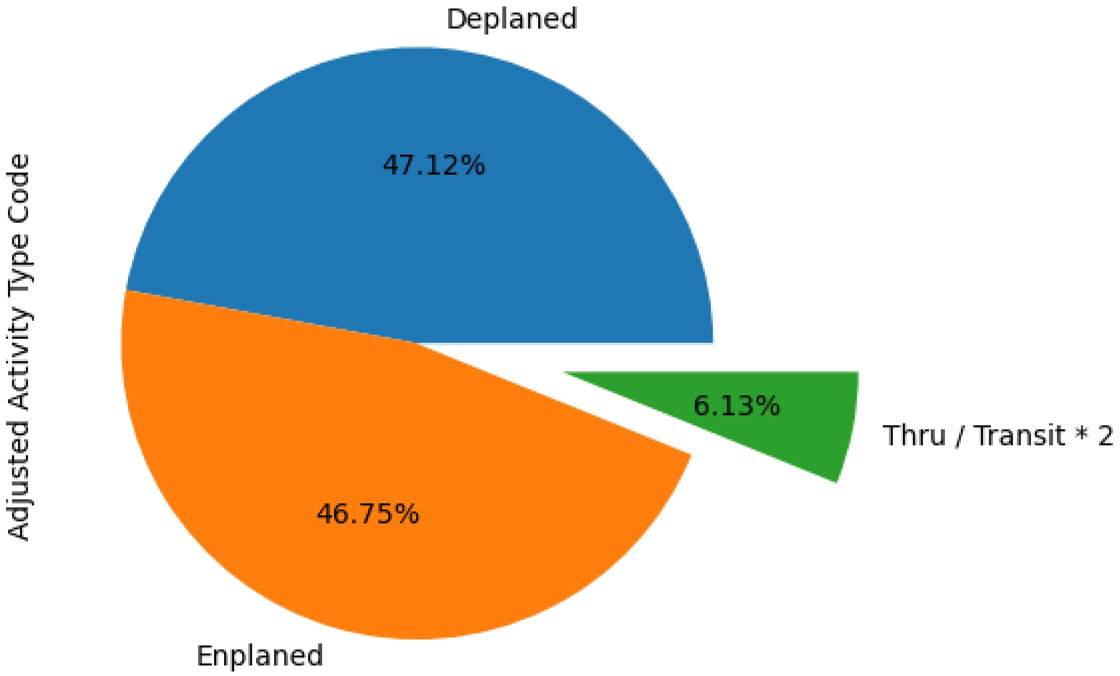


In [11]:
```
df.columns
```

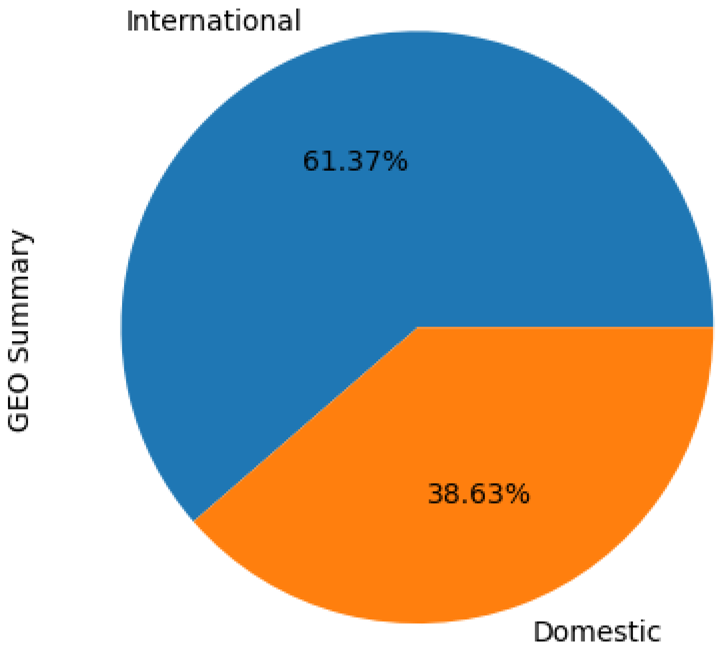
Out[11]:
```
Index(['Activity Period', 'Operating Airline', 'Operating Airline IATA Cod
e',
       'Published Airline', 'Published Airline IATA Code', 'GEO Summary',
       'GEO Region', 'Activity Type Code', 'Price Category Code', 'Termina
l',
       'Boarding Area', 'Passenger Count', 'Adjusted Activity Type Code',
       'Adjusted Passenger Count', 'Year', 'Month'],
      dtype='object')
```

# Pie Chart

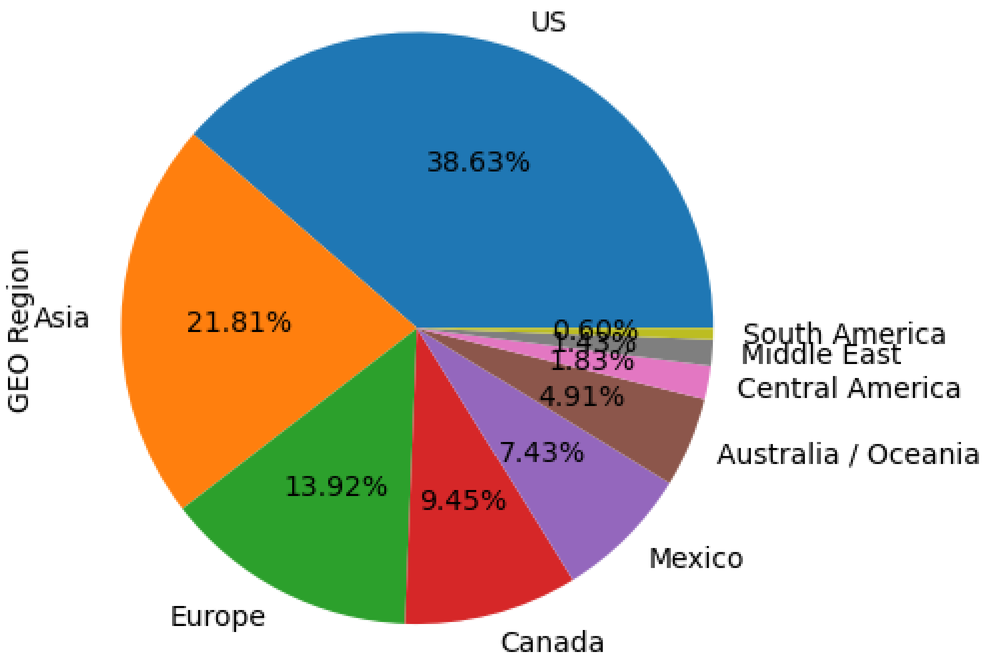
```
In [35]: df['Adjusted Activity Type Code'].value_counts().plot(kind='pie', autopct='%
         plt.show()
```



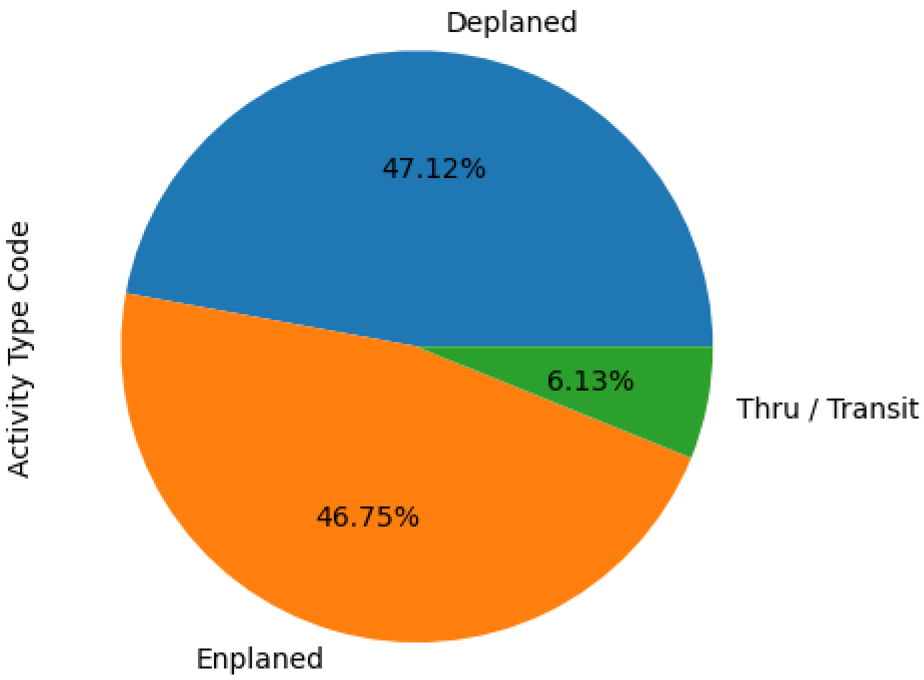
```
In [47]: df['GEO Summary'].value_counts().plot(kind='pie', autopct='%0.2f%%')
         plt.show()
```



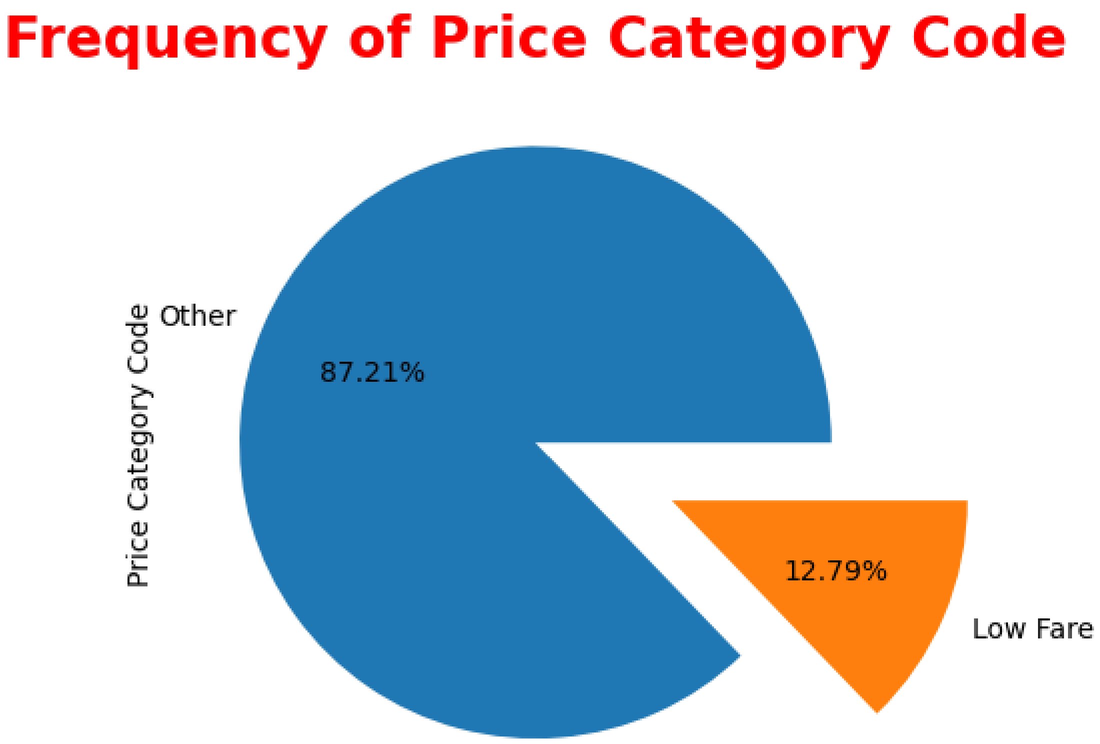
```
In [48]: df['GEO Region'].value_counts().plot(kind='pie', autopct='%0.2f%%')
         plt.show()
```



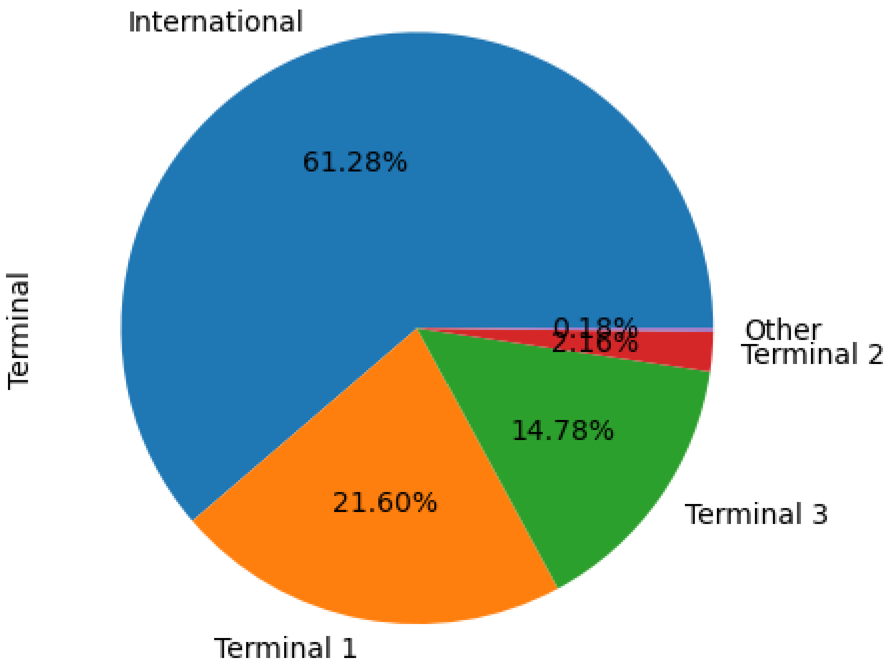
```
In [49]: df['Activity Type Code'].value_counts().plot(kind='pie', autopct='%0.2f%%')
         plt.show()
```



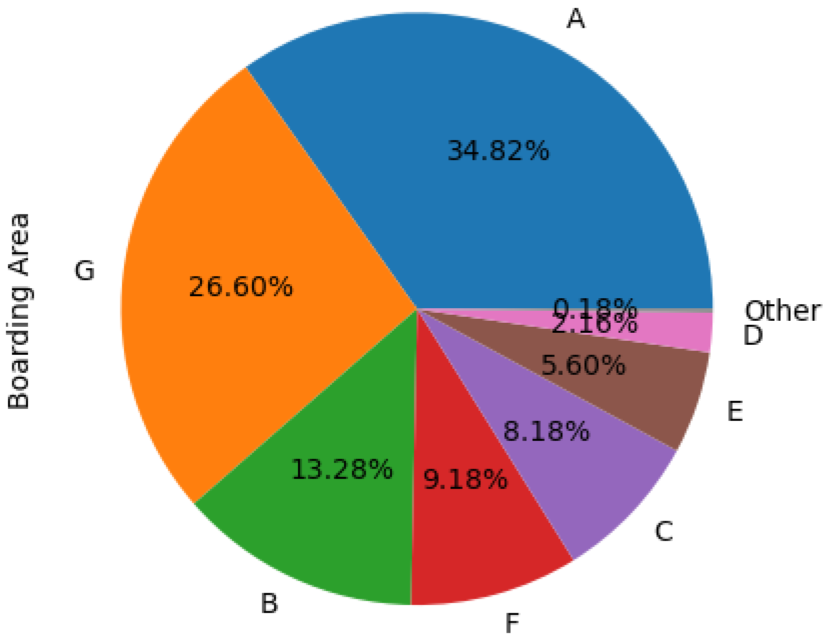
In [26]:
```
df['Price Category Code'].value_counts().plot(kind='pie', autopct='%0.2f%%'
plt.title("Frequency of Price Category Code",fontsize=20,fontweight='bold',
plt.show()
```



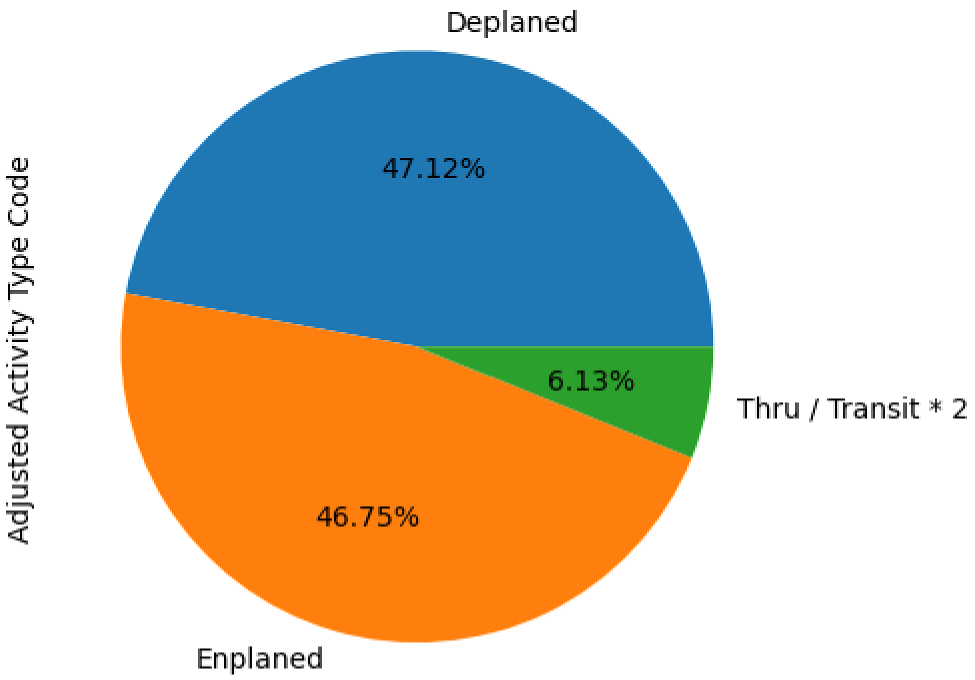
In [51]:
```
df['Terminal'].value_counts().plot(kind='pie', autopct='%0.2f%%')
plt.show()
```



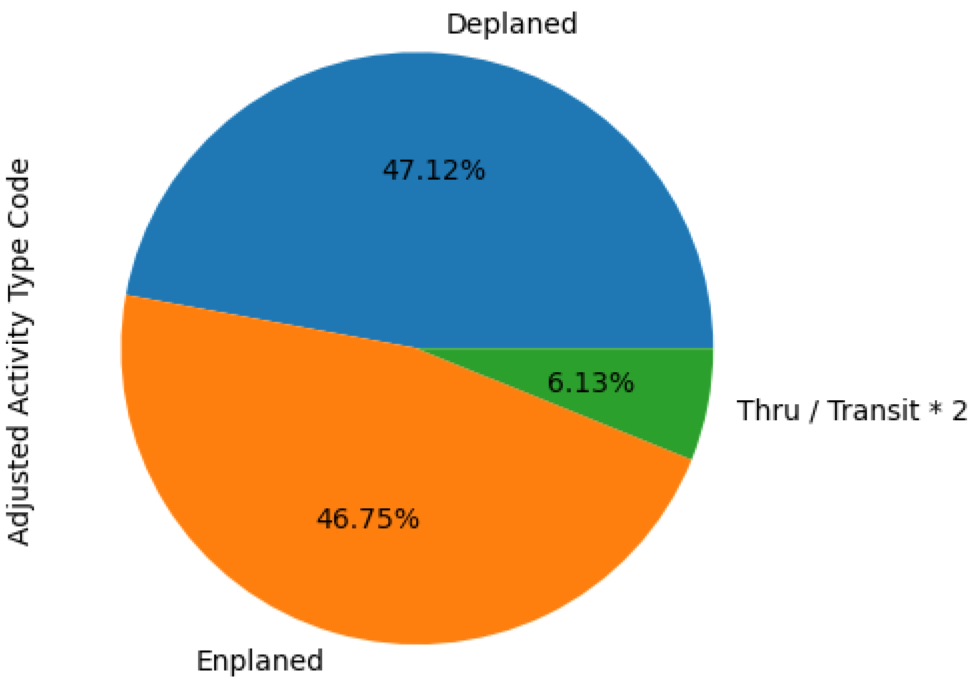
In [52]:
```
df['Boarding Area'].value_counts().plot(kind='pie', autopct='%0.2f%%')
plt.show()
```



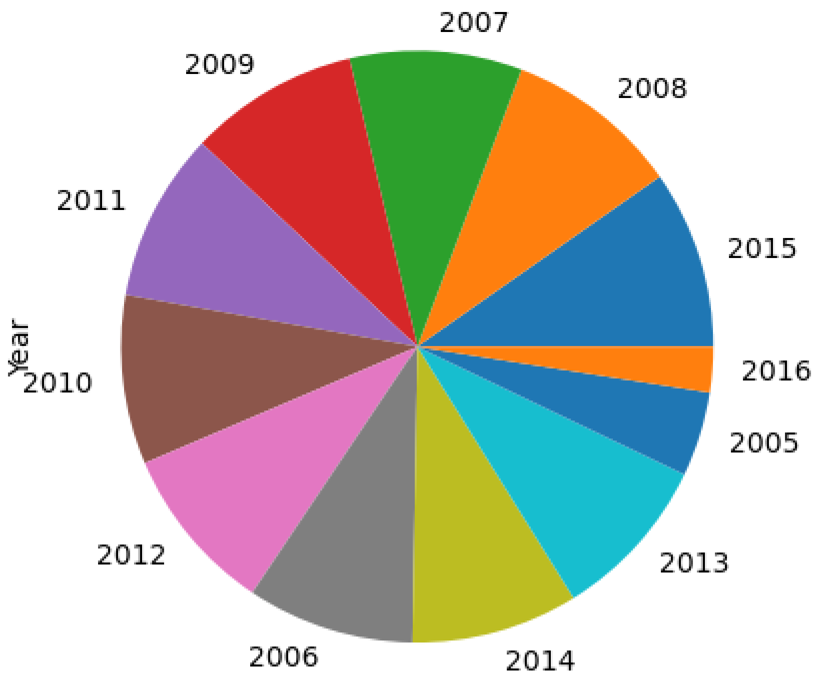
In [64]:
```
df['Adjusted Activity Type Code'].value_counts().plot(kind='pie', autopct='%
plt.show()
```



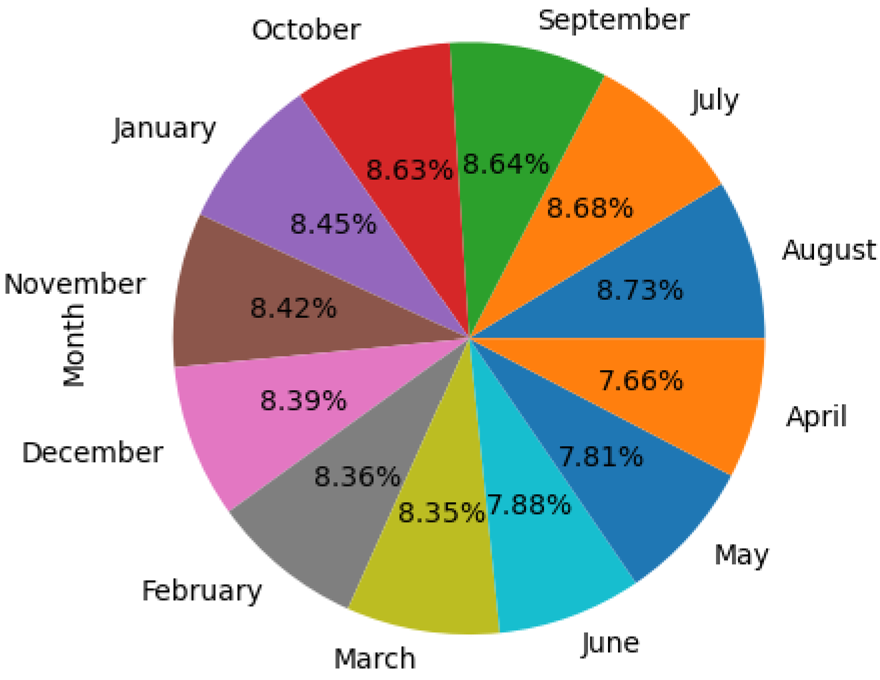
```
In [69]: df['Adjusted Activity Type Code'].value_counts().plot(kind='pie', autopct='%
         plt.show()
```



```
In [7]: df['Year'].value_counts().plot(kind='pie')
        plt.show()
```



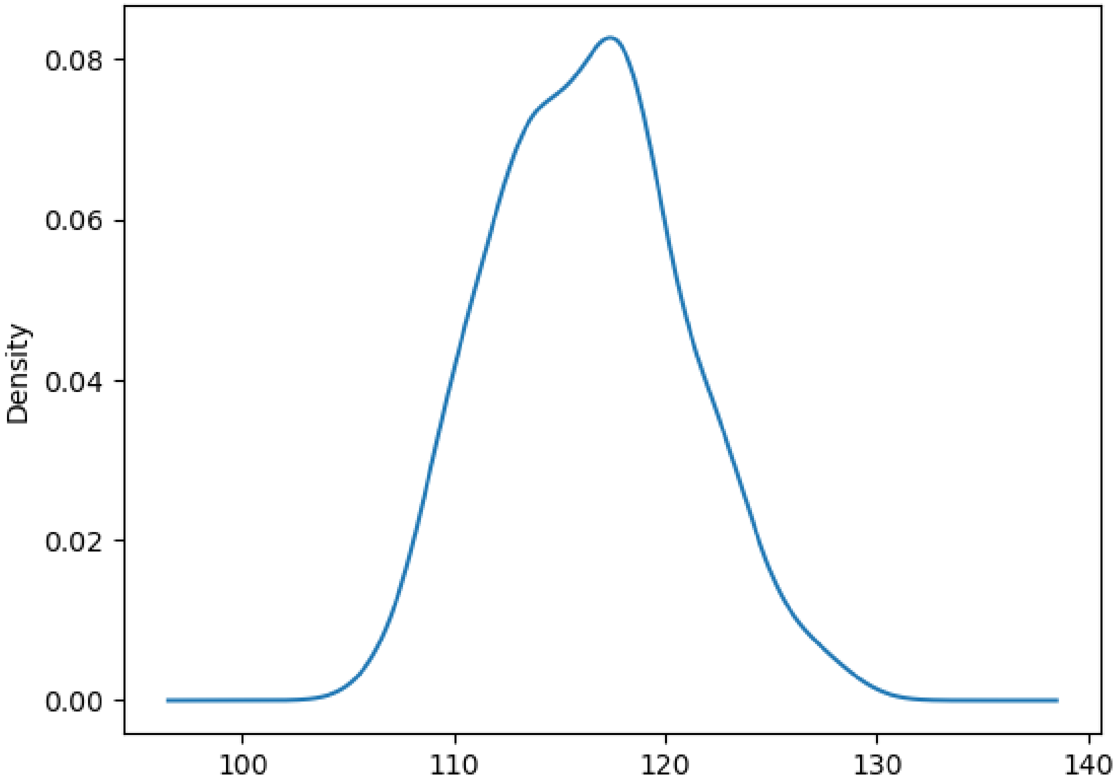
```
In [38]: df['Month'].value_counts().plot(kind='pie', autopct='%0.2f%%')
         plt.show()
```



```
In [36]: df.columns
```

```
Out[36]: Index(['Activity Period', 'Operating Airline', 'Operating Airline IATA Cod
         e',
                'Published Airline', 'Published Airline IATA Code', 'GEO Summary',
                'GEO Region', 'Activity Type Code', 'Price Category Code', 'Termina
         l',
                'Boarding Area', 'Passenger Count', 'Adjusted Activity Type Code',
                'Adjusted Passenger Count', 'Year', 'Month'],
               dtype='object')
```

```
In [5]: df['Activity Period'].value_counts().plot(kind='kde')
        plt.show()
```



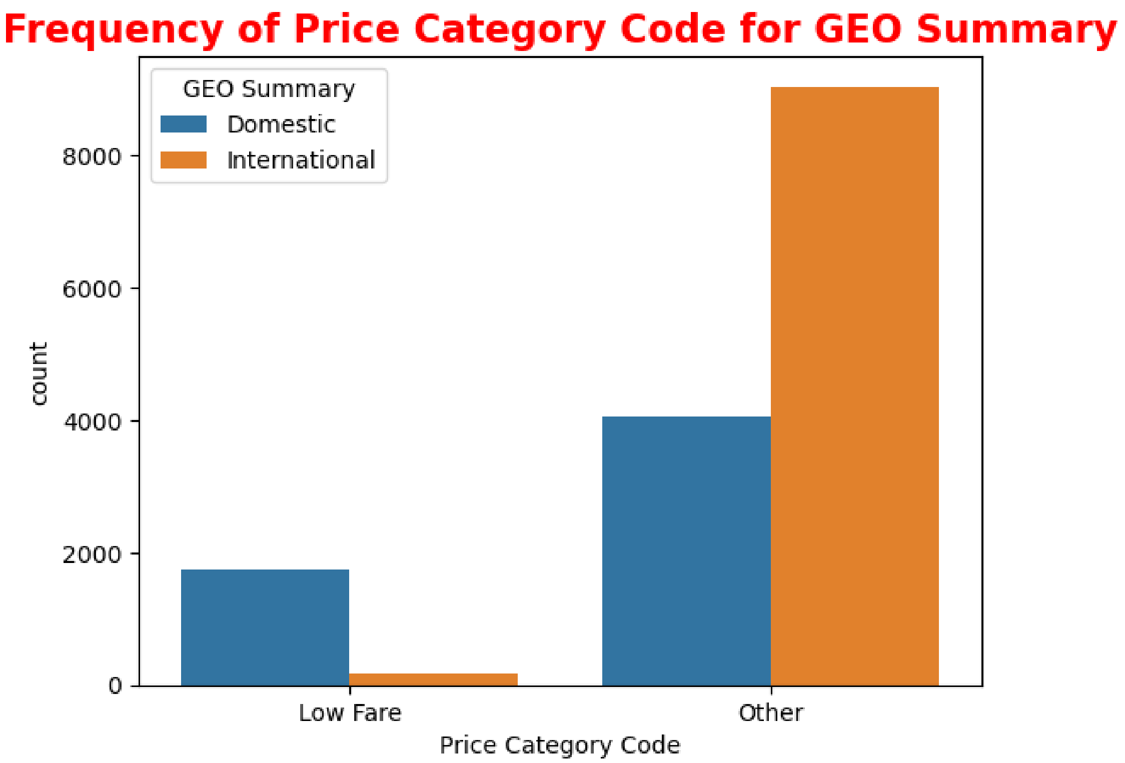
```
In [7]: df['Year'].min()

Out[7]: 2005
```

```
In [8]: df['Year'].max()

Out[8]: 2016
```

```
In [9]: df['Operating Airline'].value_counts()

Out[9]: United Airlines - Pre 07/01/2013    2154
        SkyWest Airlines                     963
        United Airlines                      892
        Alaska Airlines                      751
        Delta Air Lines                      386
                                            ...
        Evergreen International Airlines       2
        Atlas Air, Inc                         2
        Xtra Airways                           2
        Pacific Aviation                       2
        Boeing Company                         1
        Name: Operating Airline, Length: 77, dtype: int64
```

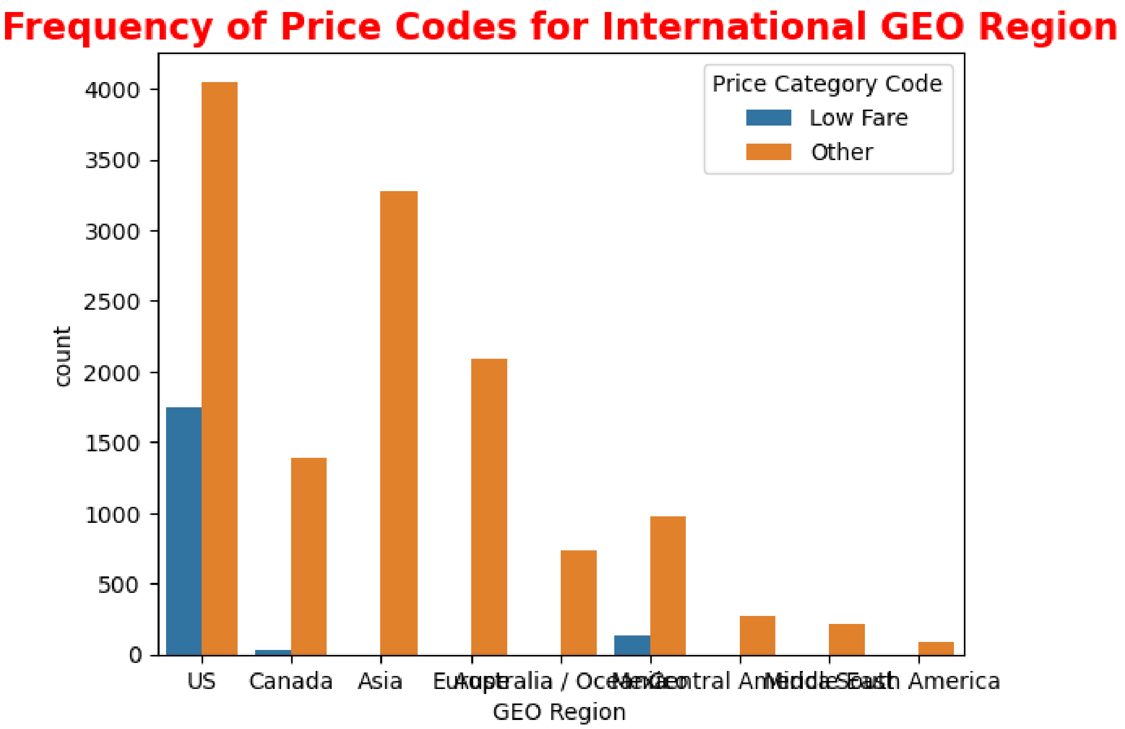
In [39]:
```
sns.countplot(x="Price Category Code",data=df,hue='GEO Summary')
plt.title('Frequency of Price Category Code for GEO Summary',fontsize=16,for
```

Out[39]:
```
Text(0.5, 1.0, 'Frequency of Price Category Code for GEO Summary')
```



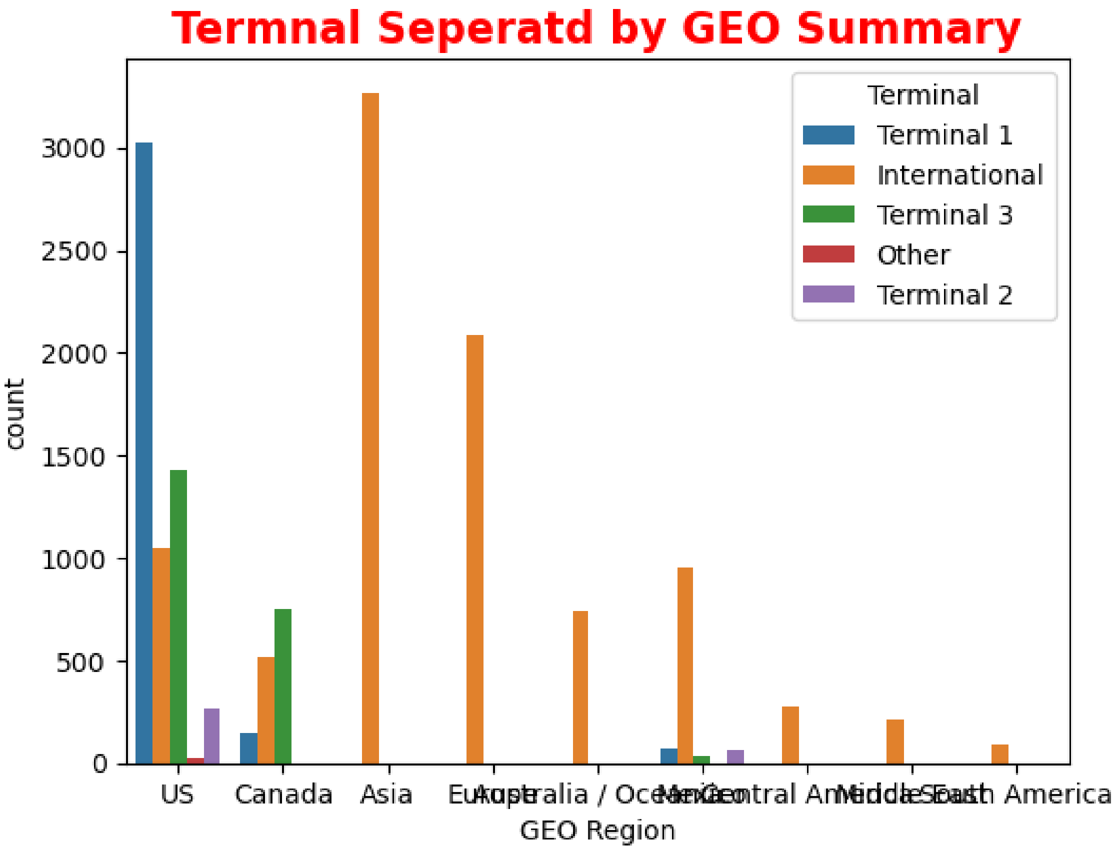
In [40]:
```
sns.countplot(x="GEO Region",data=df,hue='Price Category Code')
plt.title('Frequency of Price Codes for International GEO Region',fontsize=1
```

Out[40]:
```
Text(0.5, 1.0, 'Frequency of Price Codes for International GEO Region')
```

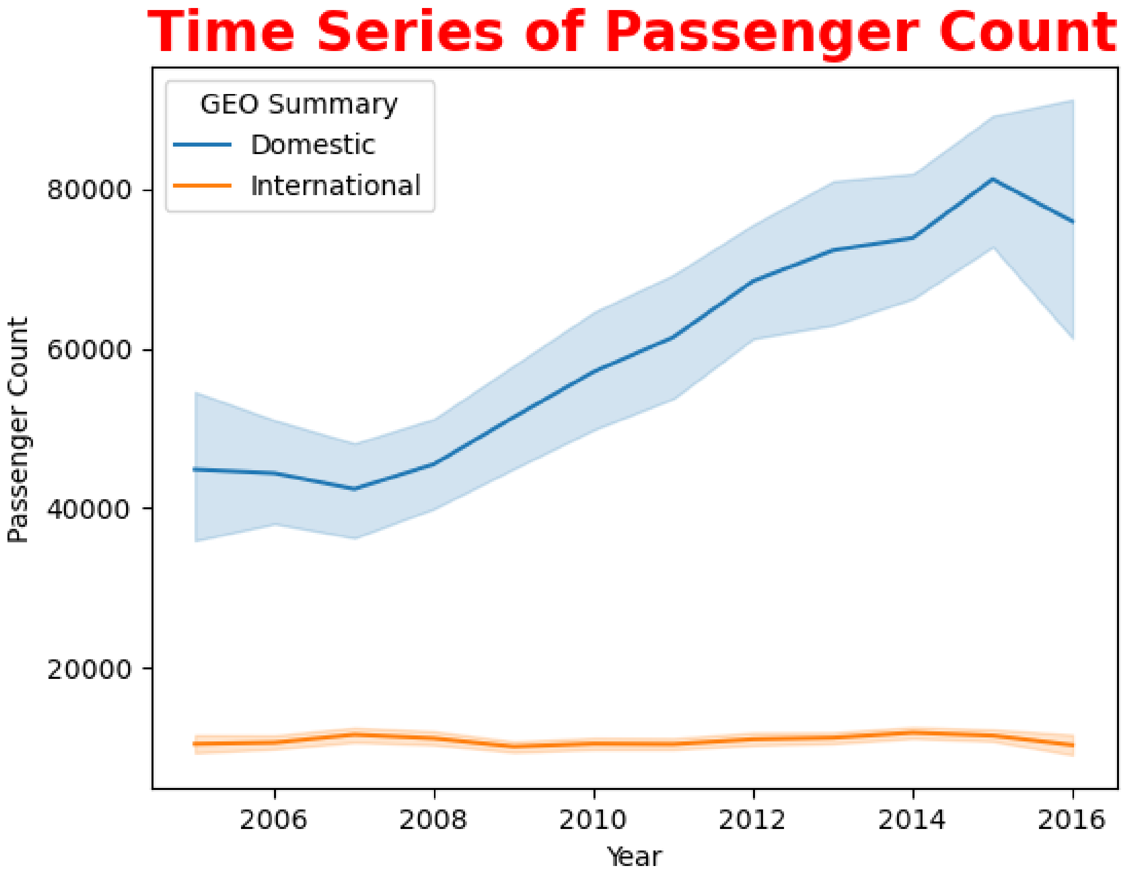


In [42]:
```
sns.countplot(x="GEO Region",data=df,hue='Terminal')
plt.title('Termnal Seperatd by GEO Summary',fontsize=16,fontweight='bold',c=
```

Out[42]:
```
Text(0.5, 1.0, 'Termnal Seperatd by GEO Summary')
```

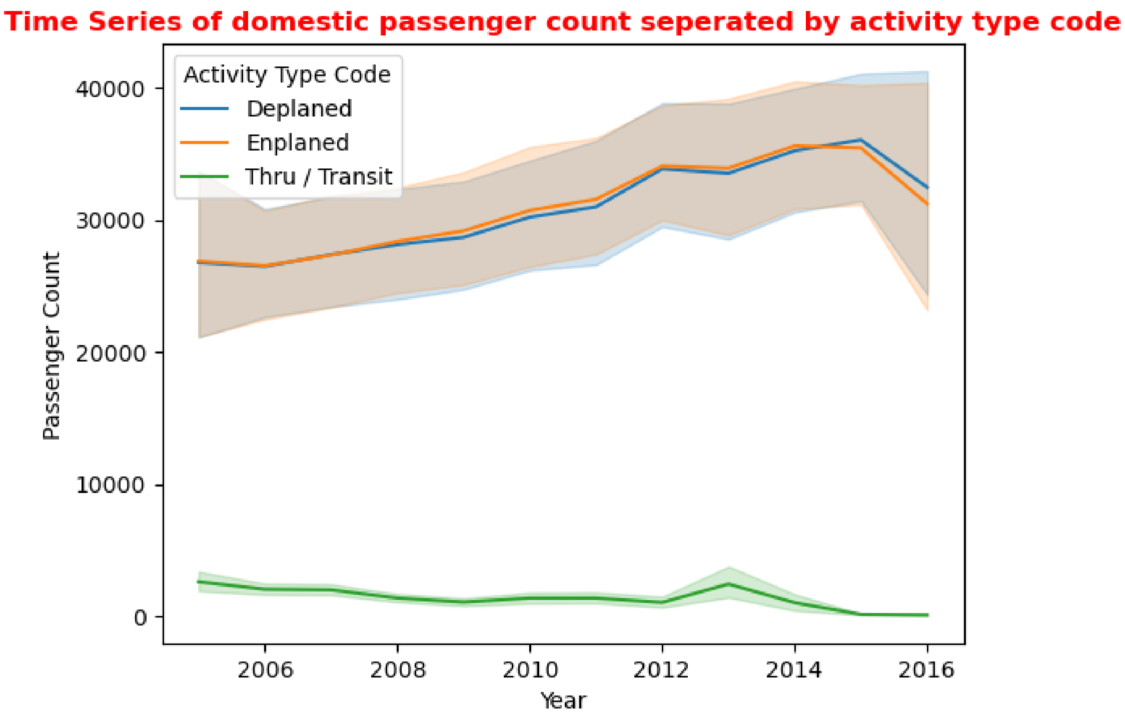


In [51]:
```
sns.lineplot(data=df,x='Year',y='Passenger Count',hue='GEO Summary')
plt.title('Time Series of Passenger Count',fontsize=20,fontweight='bold',c=
plt.show()
```



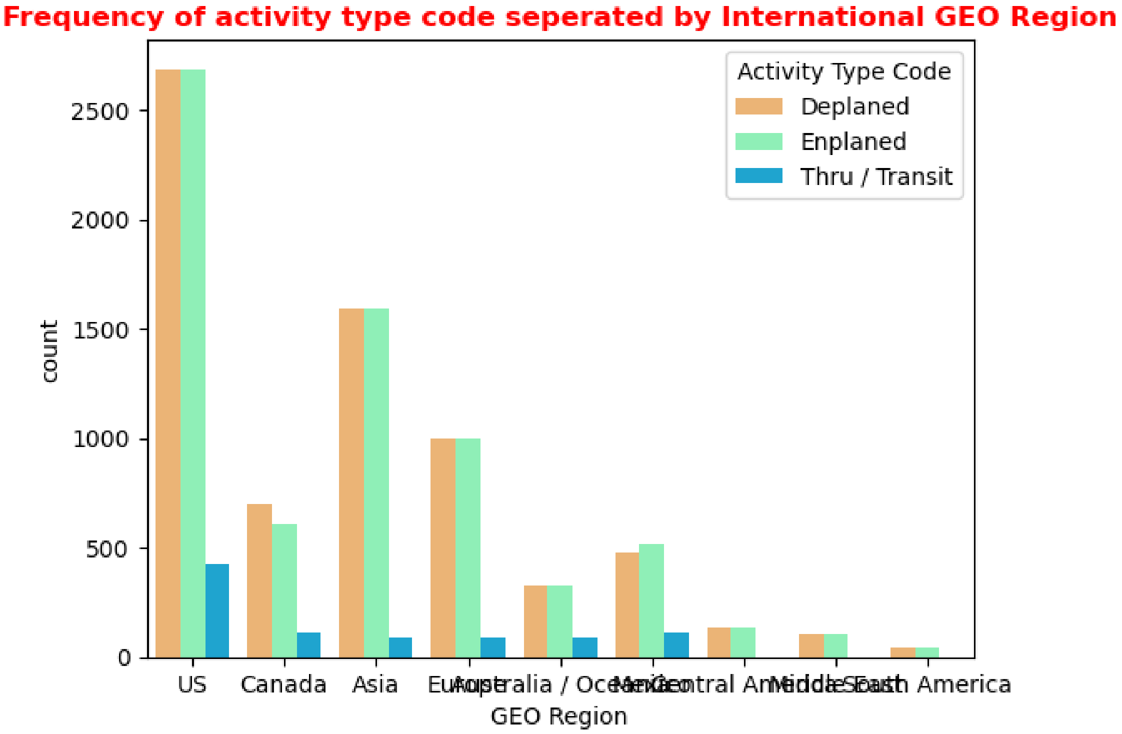
In [54]:
```
sns.lineplot(x='Year',y='Passenger Count',data=df,hue='Activity Type Code')
plt.title('Time Series of domestic passenger count seperated by activity typ
```

Out[54]:
```
Text(0.5, 1.0, 'Time Series of domestic passenger count seperated by activ
ity type code')
```



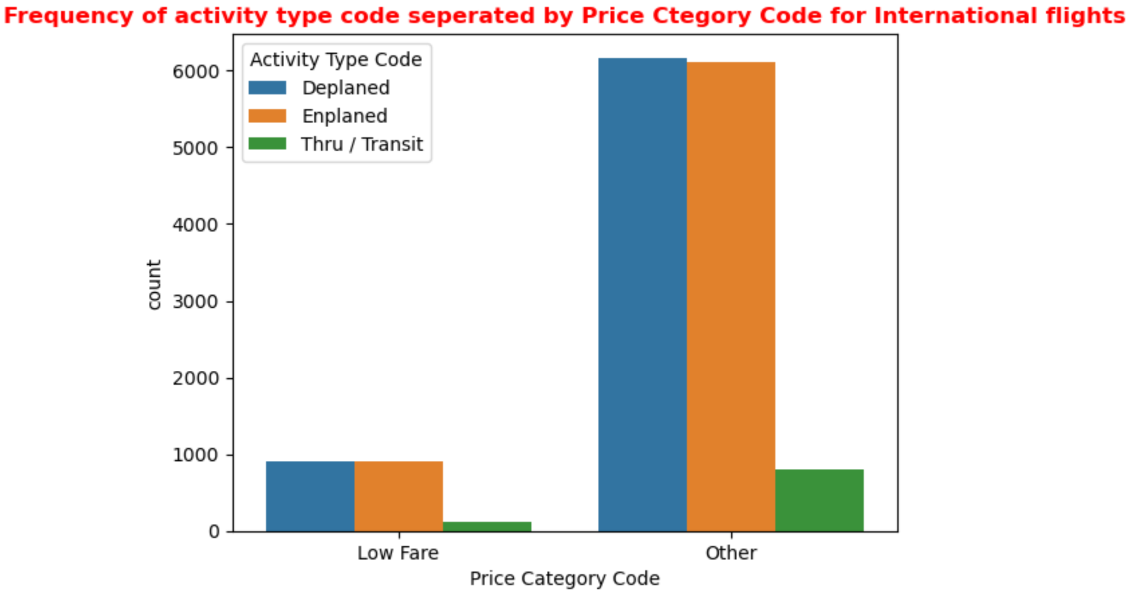
In [56]:
```
sns.countplot(x='GEO Region',data=df,hue='Activity Type Code')
plt.title('Frequency of activity type code seperated by International GEO Re
```

Out[56]:
```
Text(0.5, 1.0, 'Frequency of activity type code seperated by International
GEO Region')
```



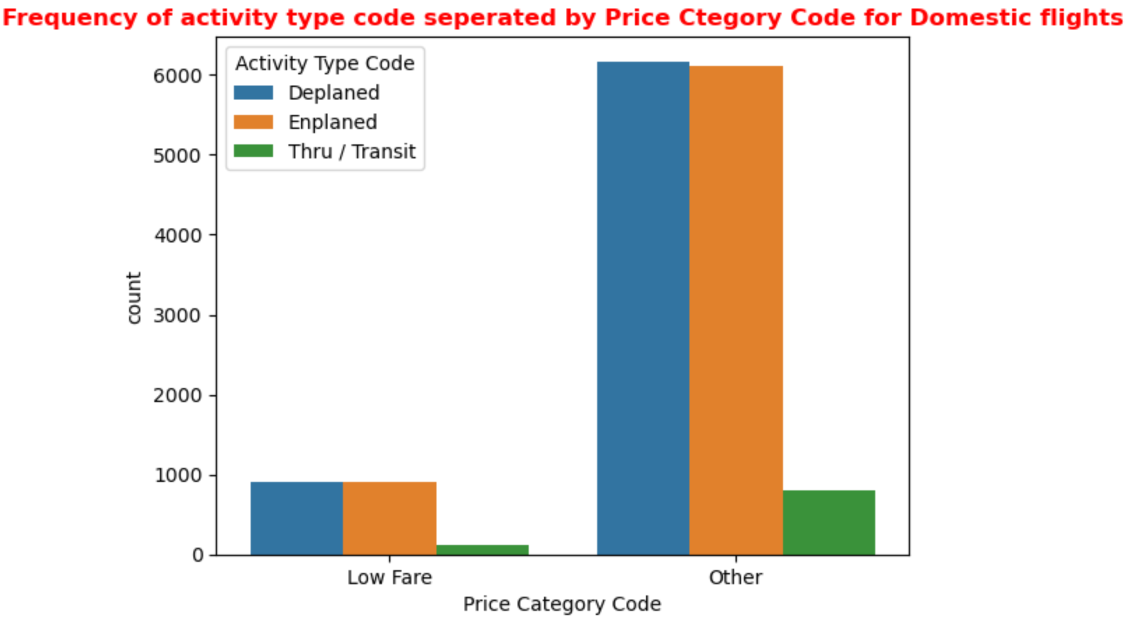
In [57]:
```
sns.countplot(x='Price Category Code',data=df,hue='Activity Type Code')
plt.title('Frequency of activity type code seperated by Price Ctegory Code f
```

Out[57]:
```
Text(0.5, 1.0, 'Frequency of activity type code seperated by Price Ctegory
Code for International flights')
```

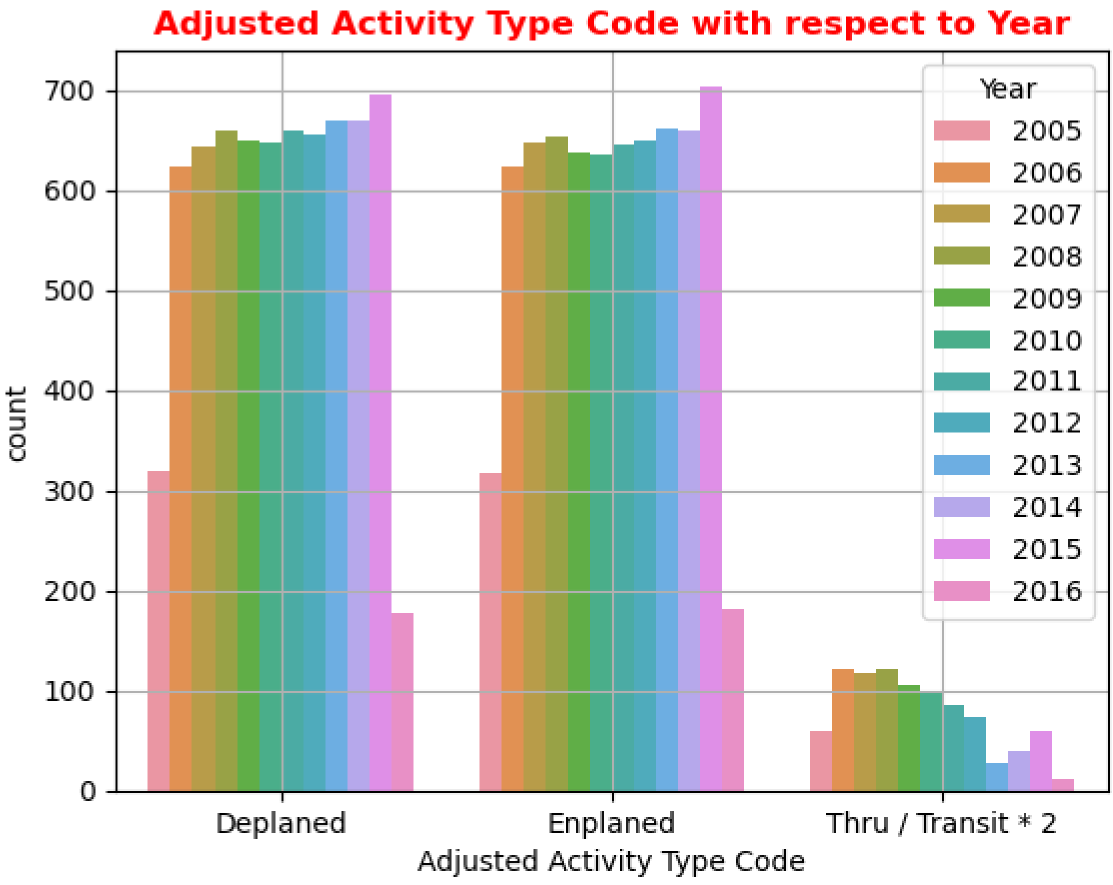


In [61]:
```
sns.countplot(x='Price Category Code',data=df,hue='Activity Type Code')
plt.title('Frequency of activity type code seperated by Price Ctegory Code f
```

Out[61]:
```
Text(0.5, 1.0, 'Frequency of activity type code seperated by Price Ctegory
Code for Domestic flights')
```



In [20]:
```
sns.countplot(x='Adjusted Activity Type Code',data=df,hue='Year')
plt.title('Adjusted Activity Type Code with respect to Year',fontsize=12,for
plt.grid()
plt.show()
```


Adjusted Activity Type Code with respect to Year
count
Year
Deplaned
Enplaned
Thru / Transit * 2
Adjusted Activity Type Code