In [14]:
```
#Importing the Libraries
import numpy as np
import pandas as pd
import os
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.express as px
from sklearn import preprocessing
import seaborn as sns
import matplotlib.pyplot as plt
import math
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn import preprocessing
from sklearn import svm
from sklearn.svm import SVC
```

In [6]:
```
#Read the Dataset
df=pd.read_csv("E:\\NMDS\\flightdata1.csv")
df.head()
```

Out[6]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL_NUM
0	2016	1	1	1	5	DL	N836DN	139
1	2016	1	1	1	5	DL	N964DN	147
2	2016	1	1	1	5	DL	N813DN	159
3	2016	1	1	1	5	DL	N587NW	176
4	2016	1	1	1	5	DL	N836DN	182

5 rows × 25 columns

In [36]:
```
#Tofind the shape of the dataset
df.shape
```

Out[36]:
```
(11231, 25)
```

In [35]:
```
#Tofind the datatype of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11231 entries, 0 to 11230
Data columns (total 25 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   YEAR                 11231 non-null  int64
 1   QUARTER              11231 non-null  int64
 2   MONTH                11231 non-null  int64
 3   DAY_OF_MONTH         11231 non-null  int64
 4   DAY_OF_WEEK          11231 non-null  int64
 5   UNIQUE_CARRIER       11231 non-null  object
 6   TAIL_NUM             11231 non-null  object
 7   FL_NUM               11231 non-null  int64
 8   ORIGIN_AIRPORT_ID    11231 non-null  int64
 9   ORIGIN               11231 non-null  object
 10  DEST_AIRPORT_ID      11231 non-null  int64
 11  DEST                 11231 non-null  object
 12  CRS_DEP_TIME         11231 non-null  int64
 13  DEP_TIME             11124 non-null  float64
 14  DEP_DELAY            11124 non-null  float64
 15  DEP_DEL15            11124 non-null  float64
 16  CRS_ARR_TIME         11231 non-null  int64
 17  ARR_TIME             11116 non-null  float64
 18  ARR_DELAY            11043 non-null  float64
 19  ARR_DEL15            11043 non-null  float64
 20  CANCELLED            11231 non-null  int64
 21  DIVERTED             11231 non-null  int64
 22  CRS_ELAPSED_TIME     11231 non-null  int64
 23  ACTUAL_ELAPSED_TIME  11043 non-null  float64
 24  DISTANCE             11231 non-null  int64
dtypes: float64(7), int64(14), object(4)
memory usage: 2.1+ MB
```

In [38]:
```
#to find the null values in the dataset
df.isnull()
```

Out[38]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	TAIL_NUM	FL
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
11226	False	False	False	False	False	False	False	
11227	False	False	False	False	False	False	False	
11228	False	False	False	False	False	False	False	
11229	False	False	False	False	False	False	False	
11230	False	False	False	False	False	False	False	

11231 rows × 25 columns

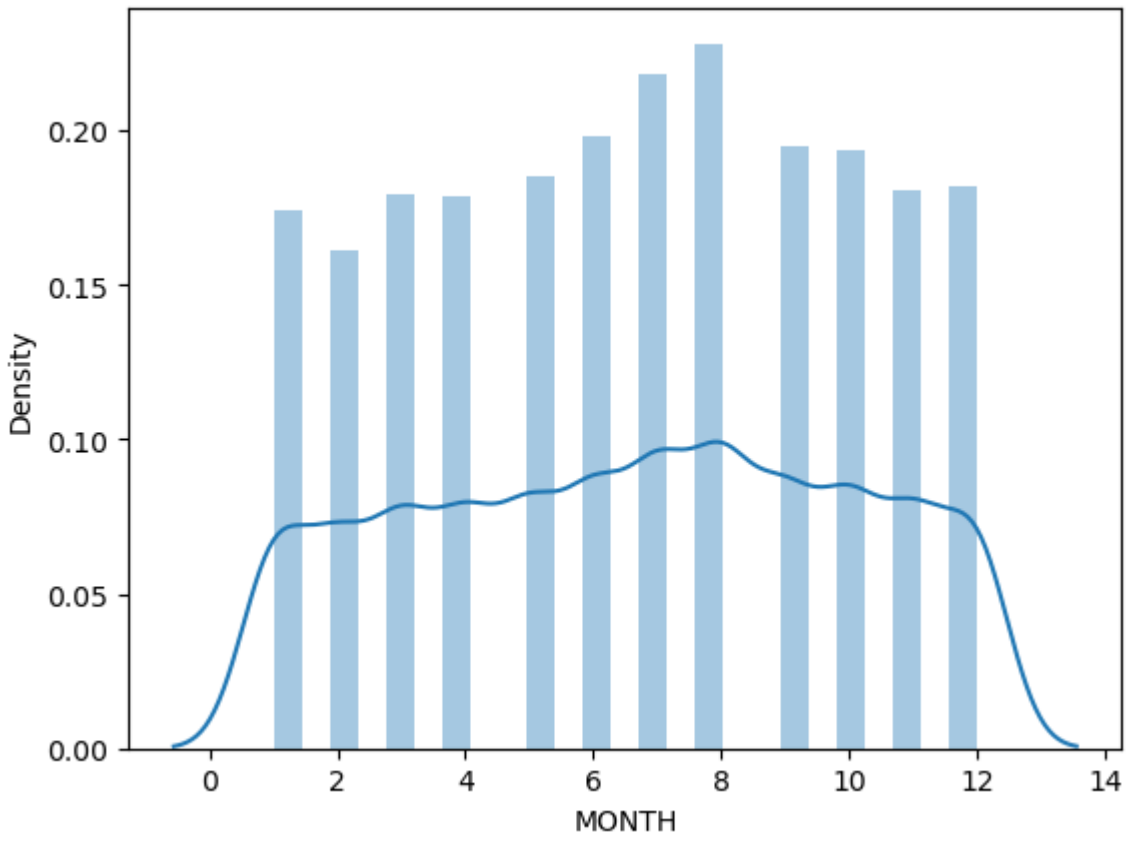
In [39]:
```
#to find the total numbers of null values in the dataset
df.isnull().sum()
```

Out[39]:
```
YEAR                  0
QUARTER               0
MONTH                 0
DAY_OF_MONTH          0
DAY_OF_WEEK           0
UNIQUE_CARRIER        0
TAIL_NUM              0
FL_NUM                0
ORIGIN_AIRPORT_ID     0
ORIGIN                0
DEST_AIRPORT_ID       0
DEST                  0
CRS_DEP_TIME          0
DEP_TIME            107
DEP_DELAY           107
DEP_DEL15           107
CRS_ARR_TIME          0
ARR_TIME            115
ARR_DELAY           188
ARR_DEL15           188
CANCELLED             0
DIVERTED              0
CRS_ELAPSED_TIME      0
ACTUAL_ELAPSED_TIME 188
DISTANCE              0
dtype: int64
```

In [44]:
```
sns.distplot(df.MONTH)
```

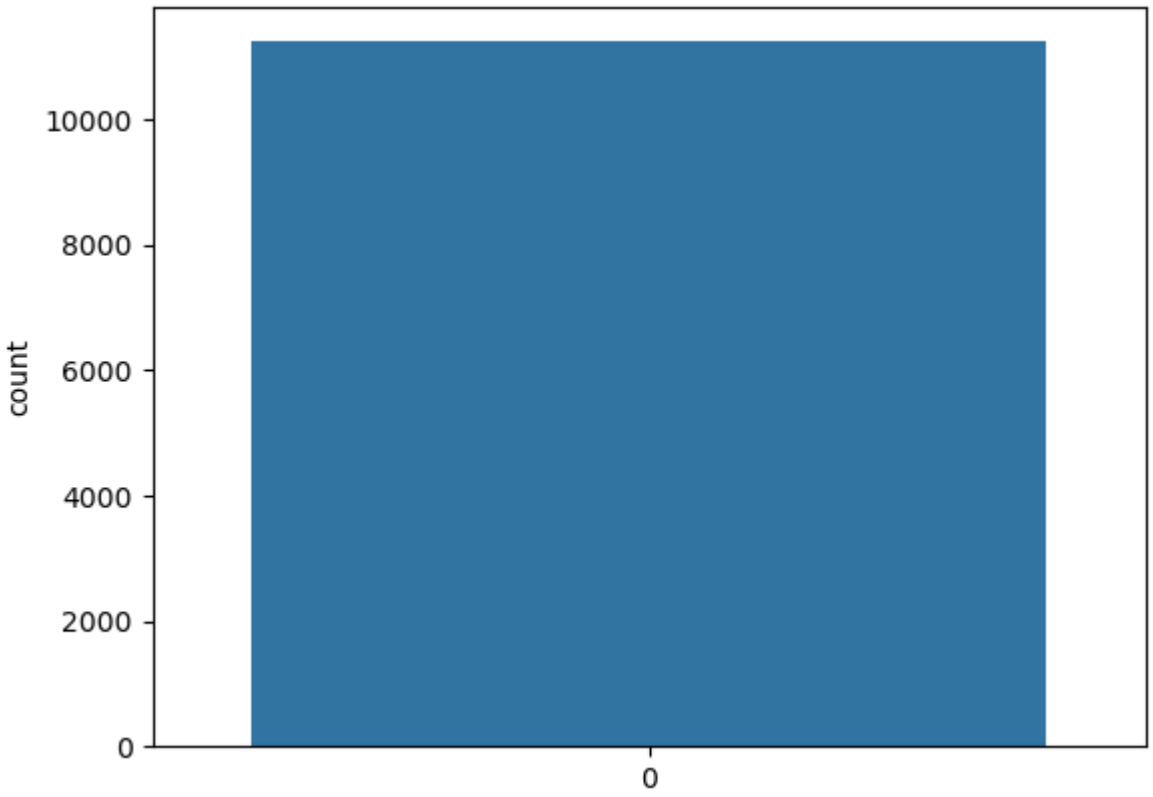
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_456\1297616383.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df.MONTH)
```

Out[44]: <Axes: xlabel='MONTH', ylabel='Density'>



In [51]:
```
sns.countplot(df.CANCELLED)
```

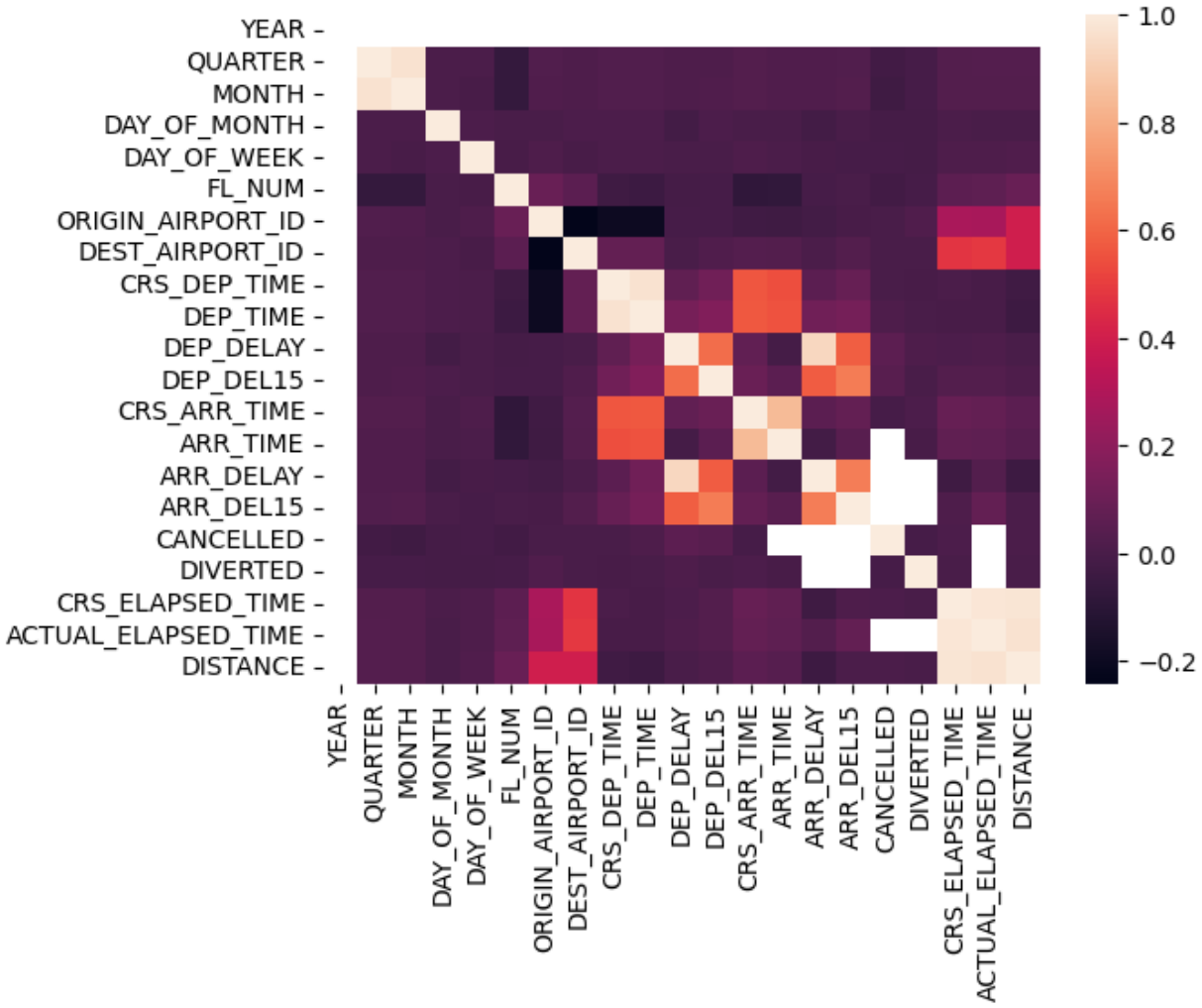
Out[51]: <Axes: ylabel='count'>



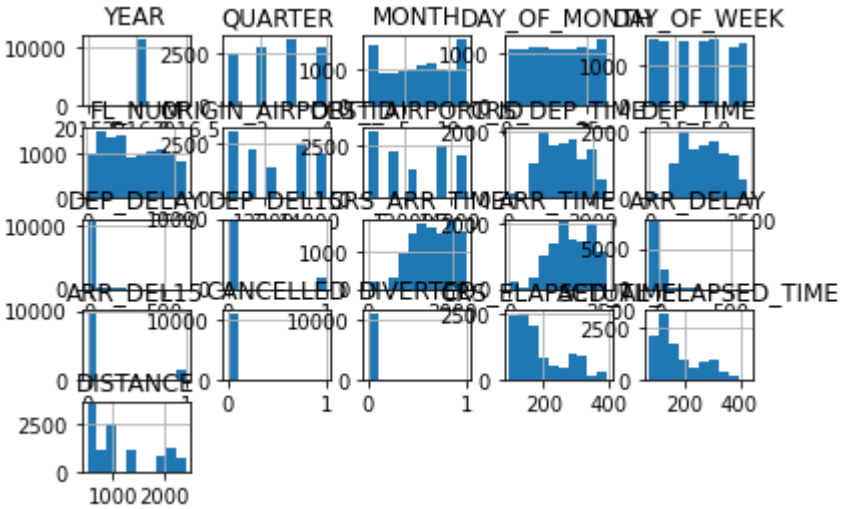
In [52]:
```
sns.heatmap(df.corr())
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_456\58359773.py:1: FutureWarning: The defa
ult value of numeric_only in DataFrame.corr is deprecated. In a future version, it wi
ll default to False. Select only valid columns or specify the value of numeric_only t
o silence this warning.
  sns.heatmap(df.corr())
```

Out[52]:
```
<Axes: >
```

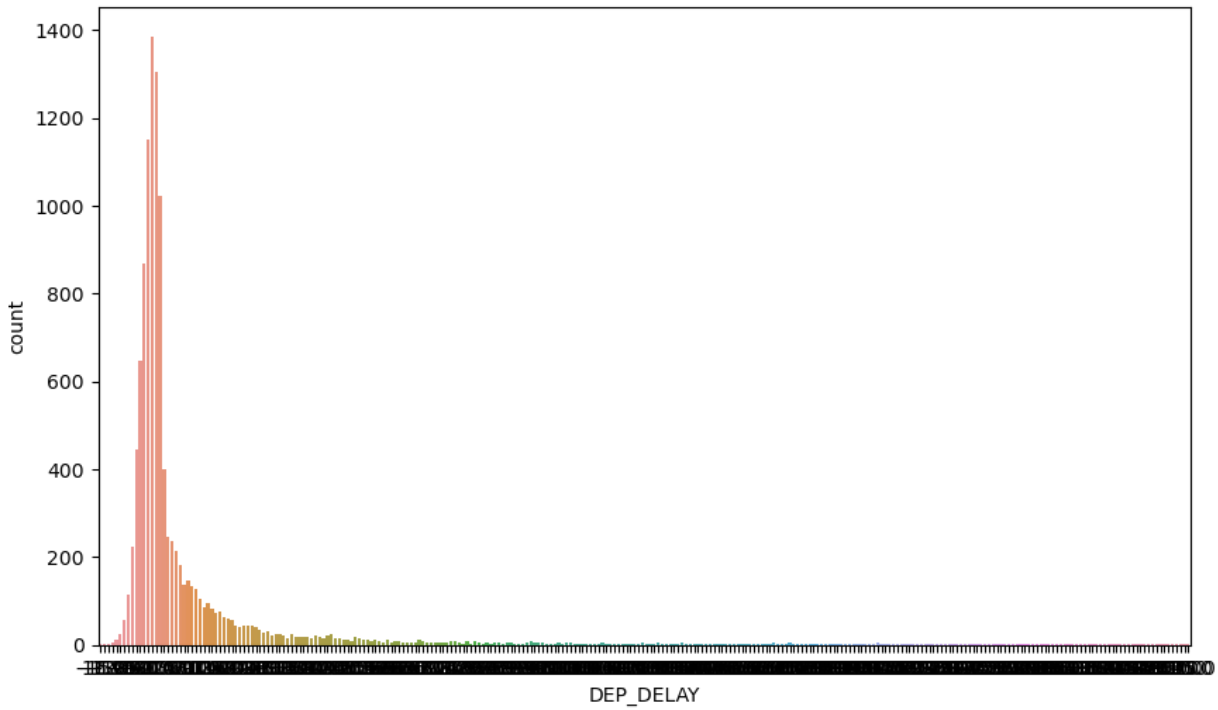


In [8]:
```
#Univariate Analysis
df.hist()
plt.show()
```

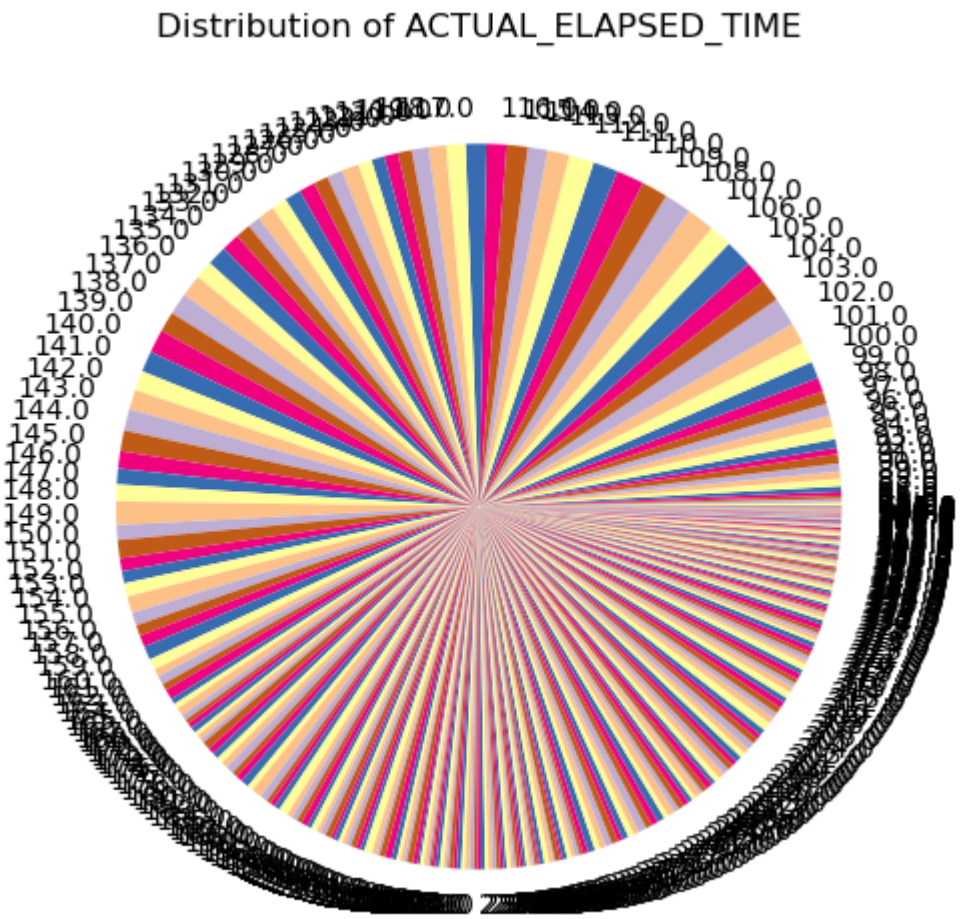


In [9]:
```
plt.figure(figsize = (10, 6), dpi = 100)
# setting the different color palette
color_palette = sns.color_palette("Accent_r")
sns.set_palette(color_palette)
```

```
sns.countplot(x = "DEP_DELAY", data = df)

plt.show()
```

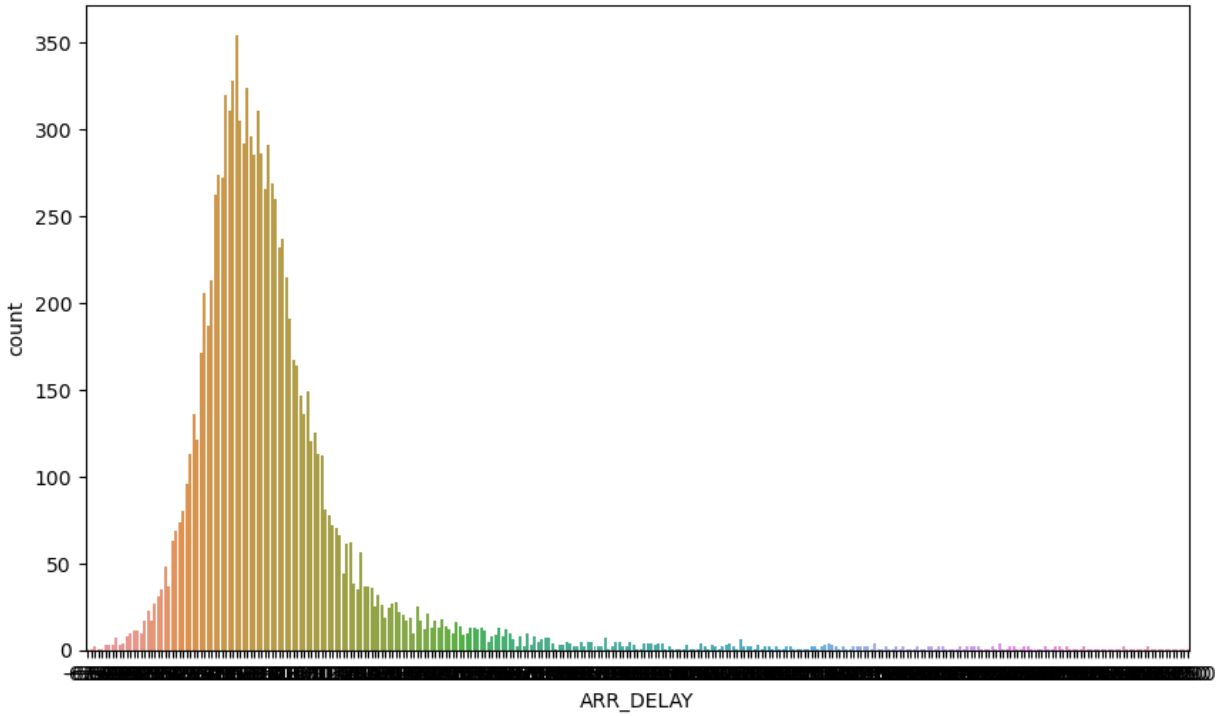


In [11]:
```
#Data Visualization  Distribution of CGPA
plt.figure(figsize = (10, 6), dpi = 100)
grp = dict(df.groupby('ACTUAL_ELAPSED_TIME').groups)

m = {}

for key, val in grp.items():

    if key in m:
        m[key] += len(val)

    else:
        m[key] = len(val)


plt.title("Distribution of ACTUAL_ELAPSED_TIME")
plt.pie(m.values(), labels = m.keys())
plt.show()
```

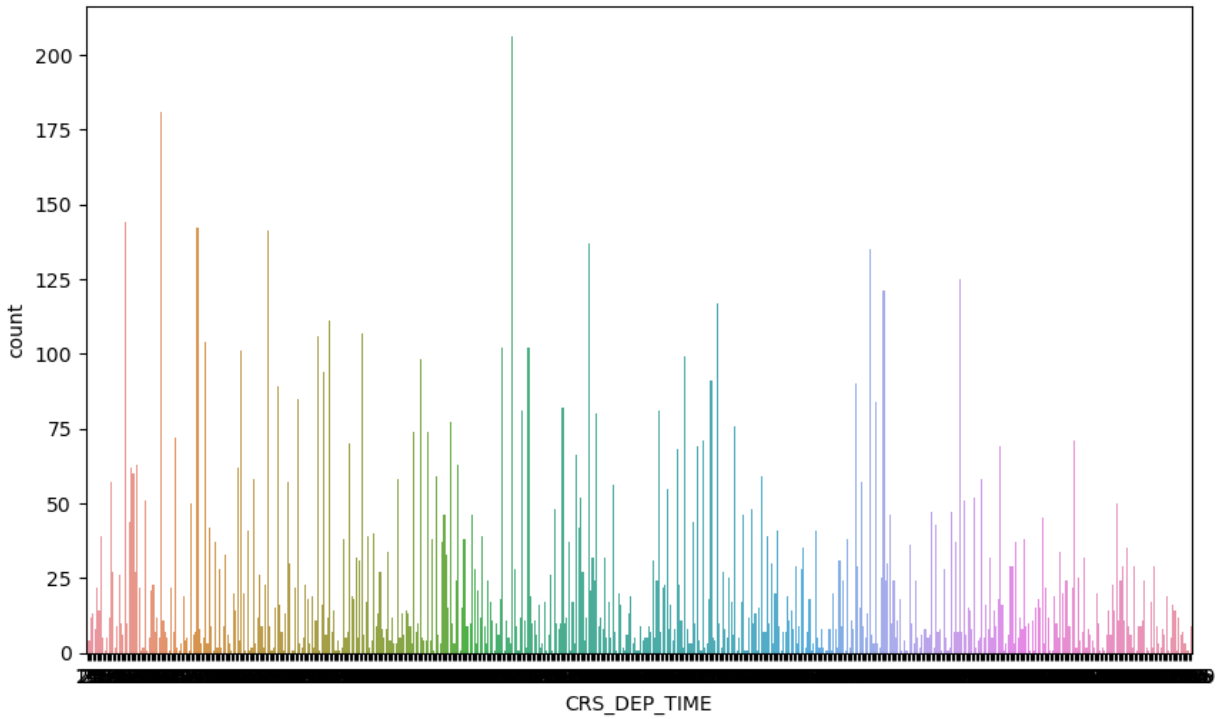


In [12]:
```
#Exploratory Data Abnliysis
#Data Visualization count of ARR_DELAY
plt.figure(figsize = (10, 6), dpi = 100)
color_palette = sns.color_palette("Accent_r")
sns.set_palette(color_palette)
sns.countplot(x = "ARR_DELAY", data = df)
```

Out[12]:
```
<AxesSubplot:xlabel='ARR_DELAY', ylabel='count'>
```



In [13]:
```
#Data Visualization count of CRS_DEP_TIME
plt.figure(figsize = (10, 6), dpi = 100)
color_palette = sns.color_palette("cool")
sns.set_palette(color_palette)
sns.countplot(x = "CRS_DEP_TIME", data = df)
plt.show()
```



In [15]:
```
df.skew()
```

```
C:\Users\Administrator\AppData\Local\Temp\ipykernel_3012\1665899112.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  df.skew()
```

Out[15]:

```
YEAR                    0.000000
QUARTER                -0.072046
MONTH                  -0.068162
DAY_OF_MONTH           -0.000712
DAY_OF_WEEK             0.028410
FL_NUM                  0.179378
ORIGIN_AIRPORT_ID       0.176974
DEST_AIRPORT_ID         0.207055
CRS_DEP_TIME            0.060403
DEP_TIME                0.029767
DEP_DELAY               7.093088
DEP_DEL15               2.041667
CRS_ARR_TIME           -0.407793
ARR_TIME               -0.421207
ARR_DELAY               5.898520
ARR_DEL15               2.274841
CANCELLED               9.775138
DIVERTED               12.199043
CRS_ELAPSED_TIME        0.904010
ACTUAL_ELAPSED_TIME     0.890397
DISTANCE                0.786107
dtype: float64
```

In [18]:

```
df=pd.read_csv("E:\\NMDS\\flightdata1.csv")
df.describe()
```

Out[18]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_NUM	ORIGIN_A
count	11231.0	11231.000000	11231.000000	11231.000000	11231.000000	11231.000000	11
mean	2016.0	2.544475	6.628973	15.790758	3.960199	1334.325617	12
std	0.0	1.090701	3.354678	8.782056	1.995257	811.875227	1
min	2016.0	1.000000	1.000000	1.000000	1.000000	7.000000	10
25%	2016.0	2.000000	4.000000	8.000000	2.000000	624.000000	10
50%	2016.0	3.000000	7.000000	16.000000	4.000000	1267.000000	12
75%	2016.0	3.000000	9.000000	23.000000	6.000000	2032.000000	13
max	2016.0	4.000000	12.000000	31.000000	7.000000	2853.000000	14

8 rows × 21 columns

In [20]:

```
df.isna().sum()
```

Out[20]:
```
YEAR                   0
QUARTER                0
MONTH                  0
DAY_OF_MONTH           0
DAY_OF_WEEK            0
UNIQUE_CARRIER         0
TAIL_NUM               0
FL_NUM                 0
ORIGIN_AIRPORT_ID      0
ORIGIN                 0
DEST_AIRPORT_ID        0
DEST                   0
CRS_DEP_TIME           0
DEP_TIME             107
DEP_DELAY            107
DEP_DEL15            107
CRS_ARR_TIME           0
ARR_TIME             115
ARR_DELAY            188
ARR_DEL15            188
CANCELLED              0
DIVERTED               0
CRS_ELAPSED_TIME       0
ACTUAL_ELAPSED_TIME  188
DISTANCE               0
dtype: int64
```

In [23]:
```
df.duplicated().sum()
```

Out[23]:
```
0
```

In [ ]:
```

```