importing Libraries

```
import pandas as pd
import numpy as np
```

loading dataset

```
fd = pd.read_csv('full_data_flightdelay.csv', na_values='?')
```

Data set overview

```
fd.head()
```

```
   MONTH  DAY_OF_WEEK  DEP_DEL15 DEP_TIME_BLK  DISTANCE_GROUP
SEGMENT_NUMBER  \
0      1            7          0    0800-0859               2
1
1      1            7          0    0700-0759               7
1
2      1            7          0    0600-0659               7
1
3      1            7          0    0600-0659               9
1
4      1            7          0    0001-0559               7
1

   CONCURRENT_FLIGHTS  NUMBER_OF_SEATS            CARRIER_NAME  \
0                  25              143  Southwest Airlines Co.
1                  29              191    Delta Air Lines Inc.
2                  27              199    Delta Air Lines Inc.
3                  27              180    Delta Air Lines Inc.
4                  10              182        Spirit Air Lines

   AIRPORT_FLIGHTS_MONTH  ...  PLANE_AGE       DEPARTING_AIRPORT
LATITUDE  \
0                  13056  ...          8  McCarran International
36.08
1                  13056  ...          3  McCarran International
36.08
2                  13056  ...         18  McCarran International
36.08
3                  13056  ...          2  McCarran International
36.08
4                  13056  ...          1  McCarran International
36.08

   LONGITUDE  PREVIOUS_AIRPORT  PRCP  SNOW SNWD  TMAX  AWND
0   -115.152              NONE   0.0   0.0  0.0  65.0  2.91
1   -115.152              NONE   0.0   0.0  0.0  65.0  2.91
2   -115.152              NONE   0.0   0.0  0.0  65.0  2.91
```

```
3   -115.152              NONE   0.0   0.0  0.0  65.0  2.91
4   -115.152              NONE   0.0   0.0  0.0  65.0  2.91

[5 rows x 26 columns]
```

```
fd.tail()
```

```
         MONTH  DAY_OF_WEEK  DEP_DEL15 DEP_TIME_BLK  DISTANCE_GROUP  \
6489057     12            7          0    2300-2359               1
6489058     12            7          0    1800-1859               1
6489059     12            7          0    2000-2059               1
6489060     12            7          0    2100-2159               1
6489061     12            7          1    2100-2159               1

         SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS  \
6489057              11                   3              123
6489058              11                   2              123
6489059              11                   2              123
6489060              12                   3              123
6489061              12                   3              123

                   CARRIER_NAME  AIRPORT_FLIGHTS_MONTH  ...  PLANE_AGE
\
6489057  Hawaiian Airlines Inc.                   1318  ...         18

6489058  Hawaiian Airlines Inc.                   1318  ...         16

6489059  Hawaiian Airlines Inc.                   1318  ...         18

6489060  Hawaiian Airlines Inc.                   1318  ...         18

6489061  Hawaiian Airlines Inc.                   1318  ...         15


         DEPARTING_AIRPORT  LATITUDE  LONGITUDE
PREVIOUS_AIRPORT  PRCP  \
6489057      Lihue Airport    21.979   -159.346  Honolulu
International  0.06
6489058      Lihue Airport    21.979   -159.346  Honolulu
International  0.06
6489059      Lihue Airport    21.979   -159.346  Honolulu
International  0.06
6489060      Lihue Airport    21.979   -159.346  Honolulu
International  0.06
6489061      Lihue Airport    21.979   -159.346  Honolulu
International  0.06

         SNOW SNWD  TMAX   AWND
6489057   0.0  0.0  84.0  15.21
6489058   0.0  0.0  84.0  15.21
6489059   0.0  0.0  84.0  15.21
```

```
6489060   0.0  0.0  84.0  15.21
6489061   0.0  0.0  84.0  15.21

[5 rows x 26 columns]
```

number of observation and features

```
fd.shape
```

```
(6489062, 26)
```

Data types of features

```
fd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6489062 entries, 0 to 6489061
Data columns (total 26 columns):
 #   Column                         Dtype  
---  ------                         -----  
 0   MONTH                          int64  
 1   DAY_OF_WEEK                    int64  
 2   DEP_DEL15                      int64  
 3   DEP_TIME_BLK                   object 
 4   DISTANCE_GROUP                 int64  
 5   SEGMENT_NUMBER                 int64  
 6   CONCURRENT_FLIGHTS             int64  
 7   NUMBER_OF_SEATS                int64  
 8   CARRIER_NAME                   object 
 9   AIRPORT_FLIGHTS_MONTH          int64  
 10  AIRLINE_FLIGHTS_MONTH          int64  
 11  AIRLINE_AIRPORT_FLIGHTS_MONTH  int64  
 12  AVG_MONTHLY_PASS_AIRPORT       int64  
 13  AVG_MONTHLY_PASS_AIRLINE       int64  
 14  FLT_ATTENDANTS_PER_PASS        float64
 15  GROUND_SERV_PER_PASS           float64
 16  PLANE_AGE                      int64  
 17  DEPARTING_AIRPORT              object 
 18  LATITUDE                       float64
 19  LONGITUDE                      float64
 20  PREVIOUS_AIRPORT               object 
 21  PRCP                           float64
 22  SNOW                           float64
 23  SNWD                           float64
 24  TMAX                           float64
 25  AWND                           float64
dtypes: float64(9), int64(13), object(4)
memory usage: 1.3+ GB
```

find duplicate rows in Dataset

```
fd[fd.duplicated()]
```

```
         MONTH  DAY_OF_WEEK  DEP_DEL15 DEP_TIME_BLK  DISTANCE_GROUP  \
44           1            7          0    0700-0759               5   
46           1            7          0    0700-0759               5   
51           1            7          0    0800-0859               1   
73           1            7          0    0700-0759               6   
85           1            7          0    0700-0759               6   
...        ...          ...        ...          ...             ...   
6488861     12            3          0    0700-0759               1   
6488862     12            3          0    0700-0759               1   
6488979     12            7          0    0700-0759               1   
6488980     12            7          0    0700-0759               1   
6489045     12            7          0    1000-1059               1   

         SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS  \
44                    1                  29              129   
46                    1                  29              129   
51                    1                  25              143   
73                    1                  29              129   
85                    1                  29              129   
...                 ...                 ...              ...   
6488861               2                   4              123   
6488862               2                   4              123   
6488979               2                   4              123   
6488980               2                   4              123   
6489045               4                   2              123   

                   CARRIER_NAME  AIRPORT_FLIGHTS_MONTH  ...  PLANE_AGE  
\
44               Allegiant Air                  13056  ...         11   

46               Allegiant Air                  13056  ...         11   

51      Southwest Airlines Co.                  13056  ...         16   

73               Allegiant Air                  13056  ...         11   

85               Allegiant Air                  13056  ...         11   

...                        ...                    ...  ...        ...   

6488861  Hawaiian Airlines Inc.                  2484  ...         18   

6488862  Hawaiian Airlines Inc.                  2484  ...         18   

6488979  Hawaiian Airlines Inc.                  2484  ...         18   

6488980  Hawaiian Airlines Inc.                  2484  ...         18   
```

```
6489045  Hawaiian Airlines Inc.                        1318  ...         15


              DEPARTING_AIRPORT  LATITUDE  LONGITUDE
PREVIOUS_AIRPORT  \
44       McCarran International    36.080   -115.152
NONE
46       McCarran International    36.080   -115.152
NONE
51       McCarran International    36.080   -115.152
NONE
73       McCarran International    36.080   -115.152
NONE
85       McCarran International    36.080   -115.152
NONE
...                         ...       ...        ...
...
6488861         Kahului Airport    20.901   -156.434  Honolulu
International
6488862         Kahului Airport    20.901   -156.434  Honolulu
International
6488979         Kahului Airport    20.901   -156.434  Honolulu
International
6488980         Kahului Airport    20.901   -156.434  Honolulu
International
6489045           Lihue Airport    21.979   -159.346  Honolulu
International

         PRCP  SNOW SNWD  TMAX   AWND
44       0.00   0.0  0.0  65.0   2.91
46       0.00   0.0  0.0  65.0   2.91
51       0.00   0.0  0.0  65.0   2.91
73       0.00   0.0  0.0  65.0   2.91
85       0.00   0.0  0.0  65.0   2.91
...       ...   ...  ...   ...    ...
6488861  0.00   0.0  0.0  85.0   4.92
6488862  0.00   0.0  0.0  85.0   4.92
6488979  0.00   0.0  0.0  84.0  10.29
6488980  0.00   0.0  0.0  84.0  10.29
6489045  0.06   0.0  0.0  84.0  15.21

[28473 rows x 26 columns]
```

encode the categorical data

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

def clean_labels_encoder(list_of_labels, fd):
```

```
    for label in list_of_labels:
        fd[label] = le.fit_transform(fd[label])
    return fd

# clean the labels
list_of_labels = ['CARRIER_NAME', 'DEPARTING_AIRPORT',
'PREVIOUS_AIRPORT', 'DEP_TIME_BLK']
fd = clean_labels_encoder(list_of_labels, fd)

# show head of the dataset
fd.head()
```

```
   MONTH  DAY_OF_WEEK  DEP_DEL15  DEP_TIME_BLK  DISTANCE_GROUP  \
0      1            7          0             3               2
1      1            7          0             2               7
2      1            7          0             1               7
3      1            7          0             1               9
4      1            7          0             0               7

   SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS
CARRIER_NAME  \
0               1                  25              143              14

1               1                  29              191               6

2               1                  27              199               6

3               1                  27              180               6

4               1                  10              182              15


   AIRPORT_FLIGHTS_MONTH  ...  PLANE_AGE  DEPARTING_AIRPORT  LATITUDE
\
0                  13056  ...          8                 44     36.08

1                  13056  ...          3                 44     36.08

2                  13056  ...         18                 44     36.08

3                  13056  ...          2                 44     36.08

4                  13056  ...          1                 44     36.08


   LONGITUDE  PREVIOUS_AIRPORT  PRCP  SNOW  SNWD  TMAX  AWND
0   -115.152               216   0.0   0.0   0.0  65.0  2.91
1   -115.152               216   0.0   0.0   0.0  65.0  2.91
2   -115.152               216   0.0   0.0   0.0  65.0  2.91
3   -115.152               216   0.0   0.0   0.0  65.0  2.91
4   -115.152               216   0.0   0.0   0.0  65.0  2.91
```

```
[5 rows x 26 columns]
```

Encoding categorical data is a crucial step in many machine learning tasks. Categorical variables, such as 'CARRIER_NAME', 'DEPARTING_AIRPORT', 'PREVIOUS_AIRPORT', and 'DEP_TIME_BLK' in this case, contain non-numeric values and need to be converted into numeric representations before they can be used as input for machine learning algorithms.

The reason for using sklearn.preprocessing.LabelEncoder from the sklearn library is its simplicity and effectiveness in transforming categorical variables into numerical ones. Here's why LabelEncoder is commonly used:

Simple Interface: LabelEncoder provides a simple interface for encoding categorical variables. You just need to instantiate the encoder and apply the fit_transform() method to encode your data.

Numeric Representation: It converts categorical labels into numeric representations, which is essential for many machine learning algorithms that expect numerical input.

Efficiency: LabelEncoder is optimized for performance, making it efficient for large datasets. Compatible with Scikit-learn: LabelEncoder is part of the sklearn.preprocessing module, making it seamlessly integrated with other tools and utilities provided by scikit-learn.

Handles Unknown Categories: It can handle unseen categories during the transformation, assigning them a unique numerical value.

Other methods of encoding categorical data, such as one-hot encoding or ordinal encoding, may also be appropriate depending on the nature of the data and the specific requirements of the machine learning task. However, LabelEncoder is a popular choice for its simplicity and effectiveness in many scenarios.

```
fd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6489062 entries, 0 to 6489061
Data columns (total 26 columns):
 #   Column                          Dtype
---  ------                          -----
 0   MONTH                           int64
 1   DAY_OF_WEEK                     int64
 2   DEP_DEL15                       int64
 3   DEP_TIME_BLK                    int64
 4   DISTANCE_GROUP                  int64
 5   SEGMENT_NUMBER                  int64
 6   CONCURRENT_FLIGHTS              int64
 7   NUMBER_OF_SEATS                 int64
 8   CARRIER_NAME                    int64
 9   AIRPORT_FLIGHTS_MONTH           int64
 10  AIRLINE_FLIGHTS_MONTH           int64
 11  AIRLINE_AIRPORT_FLIGHTS_MONTH   int64
 12  AVG_MONTHLY_PASS_AIRPORT        int64
```

```
 13  AVG_MONTHLY_PASS_AIRLINE       int64
 14  FLT_ATTENDANTS_PER_PASS        float64
 15  GROUND_SERV_PER_PASS           float64
 16  PLANE_AGE                      int64
 17  DEPARTING_AIRPORT              int64
 18  LATITUDE                       float64
 19  LONGITUDE                      float64
 20  PREVIOUS_AIRPORT               int64
 21  PRCP                           float64
 22  SNOW                           float64
 23  SNWD                           float64
 24  TMAX                           float64
 25  AWND                           float64
dtypes: float64(9), int64(17)
memory usage: 1.3 GB
```

Now we can see that all categorical columns are change to numeric data

describe the dataset

```
fd.describe()
```

```
              MONTH   DAY_OF_WEEK     DEP_DEL15  DEP_TIME_BLK
DISTANCE_GROUP  \
count  6.489062e+06  6.489062e+06  6.489062e+06  6.489062e+06
6.489062e+06
mean   6.607062e+00  3.935598e+00  1.891441e-01  8.197697e+00
3.821102e+00
std    3.396853e+00  1.995200e+00  3.916231e-01  4.886607e+00
2.382233e+00
min    1.000000e+00  1.000000e+00  0.000000e+00  0.000000e+00
1.000000e+00
25%    4.000000e+00  2.000000e+00  0.000000e+00  4.000000e+00
2.000000e+00
50%    7.000000e+00  4.000000e+00  0.000000e+00  8.000000e+00
3.000000e+00
75%    1.000000e+01  6.000000e+00  0.000000e+00  1.200000e+01
5.000000e+00
max    1.200000e+01  7.000000e+00  1.000000e+00  1.800000e+01
1.100000e+01

       SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS
CARRIER_NAME  \
count    6.489062e+06        6.489062e+06     6.489062e+06
6.489062e+06
mean     3.046890e+00        2.783675e+01     1.337397e+02
9.119911e+00
std      1.757864e+00        2.151060e+01     4.645213e+01
5.142576e+00
min      1.000000e+00        1.000000e+00     4.400000e+01
```

```
0.000000e+00
25%      2.000000e+00        1.100000e+01     9.000000e+01
5.000000e+00
50%      3.000000e+00        2.300000e+01     1.430000e+02
1.000000e+01
75%      4.000000e+00        3.900000e+01     1.720000e+02
1.400000e+01
max      1.500000e+01        1.090000e+02     3.370000e+02
1.600000e+01

       AIRPORT_FLIGHTS_MONTH  ...     PLANE_AGE  DEPARTING_AIRPORT  \
count           6.489062e+06  ...  6.489062e+06       6.489062e+06
mean            1.268458e+04  ...  1.153211e+01       4.282200e+01
std             8.839796e+03  ...  6.935706e+00       2.728344e+01
min             1.100000e+03  ...  0.000000e+00       0.000000e+00
25%             5.345000e+03  ...  5.000000e+00       1.700000e+01
50%             1.156200e+04  ...  1.200000e+01       4.200000e+01
75%             1.761500e+04  ...  1.700000e+01       6.500000e+01
max             3.525600e+04  ...  3.200000e+01       9.500000e+01

           LATITUDE     LONGITUDE  PREVIOUS_AIRPORT          PRCP  \
count  6.489062e+06  6.489062e+06      6.489062e+06  6.489062e+06
mean   3.670581e+01 -9.425515e+01      1.861398e+02  1.037063e-01
std    5.500804e+00  1.790952e+01      8.663547e+01  3.432134e-01
min    1.844000e+01 -1.593460e+02      0.000000e+00  0.000000e+00
25%    3.343600e+01 -1.063770e+02      1.230000e+02  0.000000e+00
50%    3.750500e+01 -8.790600e+01      2.160000e+02  0.000000e+00
75%    4.077900e+01 -8.093600e+01      2.420000e+02  2.000000e-02
max    6.116900e+01 -6.600200e+01      3.550000e+02  1.163000e+01

               SNOW          SNWD          TMAX          AWND
count  6.489062e+06  6.489062e+06  6.489062e+06  6.489062e+06
mean   3.159310e-02  9.152397e-02  7.146846e+01  8.341329e+00
std    3.170163e-01  7.281285e-01  1.835333e+01  3.607604e+00
min    0.000000e+00  0.000000e+00 -1.000000e+01  0.000000e+00
25%    0.000000e+00  0.000000e+00  5.900000e+01  5.820000e+00
50%    0.000000e+00  0.000000e+00  7.400000e+01  7.830000e+00
75%    0.000000e+00  0.000000e+00  8.600000e+01  1.029000e+01
max    1.720000e+01  2.520000e+01  1.150000e+02  3.378000e+01

[8 rows x 26 columns]
```

```
fd.isnull().sum()
```

```
MONTH                            0
DAY_OF_WEEK                      0
DEP_DEL15                        0
DEP_TIME_BLK                     0
DISTANCE_GROUP                   0
SEGMENT_NUMBER                   0
```

```
CONCURRENT_FLIGHTS               0
NUMBER_OF_SEATS                  0
CARRIER_NAME                     0
AIRPORT_FLIGHTS_MONTH            0
AIRLINE_FLIGHTS_MONTH            0
AIRLINE_AIRPORT_FLIGHTS_MONTH    0
AVG_MONTHLY_PASS_AIRPORT         0
AVG_MONTHLY_PASS_AIRLINE         0
FLT_ATTENDANTS_PER_PASS          0
GROUND_SERV_PER_PASS             0
PLANE_AGE                        0
DEPARTING_AIRPORT                0
LATITUDE                         0
LONGITUDE                        0
PREVIOUS_AIRPORT                 0
PRCP                             0
SNOW                             0
SNWD                             0
TMAX                             0
AWND                             0
dtype: int64
```

we can also write fd.isna().sum() to get the missing value in the dataset

```
fd.isnull()
```

```
         MONTH  DAY_OF_WEEK  DEP_DEL15  DEP_TIME_BLK
DISTANCE_GROUP  \
0        False        False      False         False           False

1        False        False      False         False           False

2        False        False      False         False           False

3        False        False      False         False           False

4        False        False      False         False           False

...        ...          ...        ...           ...             ...

6489057  False        False      False         False           False

6489058  False        False      False         False           False

6489059  False        False      False         False           False

6489060  False        False      False         False           False

6489061  False        False      False         False           False
```

```
         SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS
CARRIER_NAME  \
0                 False               False            False
False
1                 False               False            False
False
2                 False               False            False
False
3                 False               False            False
False
4                 False               False            False
False
...                 ...                 ...              ...
...
6489057           False               False            False
False
6489058           False               False            False
False
6489059           False               False            False
False
6489060           False               False            False
False
6489061           False               False            False
False

         AIRPORT_FLIGHTS_MONTH  ...  PLANE_AGE  DEPARTING_AIRPORT
LATITUDE  \
0                        False  ...      False              False
False
1                        False  ...      False              False
False
2                        False  ...      False              False
False
3                        False  ...      False              False
False
4                        False  ...      False              False
False
...                        ...  ...        ...                ...
...
6489057                  False  ...      False              False
False
6489058                  False  ...      False              False
False
6489059                  False  ...      False              False
False
6489060                  False  ...      False              False
False
6489061                  False  ...      False              False
False
```

```
         LONGITUDE  PREVIOUS_AIRPORT   PRCP   SNOW   SNWD   TMAX
AWND
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
...            ...               ...    ...    ...    ...    ...    ..
.
6489057      False             False  False  False  False  False
False
6489058      False             False  False  False  False  False
False
6489059      False             False  False  False  False  False
False
6489060      False             False  False  False  False  False
False
6489061      False             False  False  False  False  False
False

[6489062 rows x 26 columns]
```

we can use the fillna() to fill the missing value of particular column with the help of mean value of that column

```
fd.fillna(fd.mean(), inplace=True)
```

Check Correlation

```
fd.corr()

                                  MONTH  DAY_OF_WEEK  DEP_DEL15
DEP_TIME_BLK  \
MONTH                          1.000000     0.006727  -0.019049     -
0.000650
DAY_OF_WEEK                    0.006727     1.000000  -0.000199
0.005468
DEP_DEL15                     -0.019049    -0.000199   1.000000
0.167281
DEP_TIME_BLK                  -0.000650     0.005468   0.167281
1.000000
DISTANCE_GROUP                -0.002561     0.013550   0.016289     -
0.026919
SEGMENT_NUMBER                 0.016712    -0.029812   0.117528
```

```
0.743527
CONCURRENT_FLIGHTS               0.022951    -0.027214   0.009028
0.055996
NUMBER_OF_SEATS                  0.003155     0.009300   0.011845      -
0.021984
CARRIER_NAME                     0.000090    -0.001988   0.016082
0.012839
AIRPORT_FLIGHTS_MONTH            0.036913    -0.001725   0.026740
0.107640
AIRLINE_FLIGHTS_MONTH            0.038884    -0.006282   0.003528      -
0.006798
AIRLINE_AIRPORT_FLIGHTS_MONTH    0.018836    -0.002327   0.013711
0.113420
AVG_MONTHLY_PASS_AIRPORT        -0.002490     0.000075   0.024383
0.108532
AVG_MONTHLY_PASS_AIRLINE        -0.004709    -0.004934   0.001257      -
0.016789
FLT_ATTENDANTS_PER_PASS          0.000873     0.000519  -0.002114      -
0.005063
GROUND_SERV_PER_PASS            -0.004257     0.001153  -0.016736      -
0.023117
PLANE_AGE                       -0.017344    -0.005785   0.006220
0.008759
DEPARTING_AIRPORT                0.000293     0.003262  -0.007315      -
0.042453
LATITUDE                         0.012913    -0.011503   0.000490      -
0.035409
LONGITUDE                       -0.004548    -0.006120   0.027097      -
0.004765
PREVIOUS_AIRPORT                -0.003549     0.005966  -0.013342      -
0.081345
PRCP                            -0.005133     0.018205   0.080277      -
0.005099
SNOW                            -0.053596    -0.005883   0.050156      -
0.005931
SNWD                            -0.088596    -0.009878   0.026129      -
0.004901
TMAX                             0.173454     0.007513  -0.008936
0.017053
AWND                            -0.119272     0.001785   0.050947
0.004311

                               DISTANCE_GROUP  SEGMENT_NUMBER  \
MONTH                               -0.002561        0.016712
DAY_OF_WEEK                          0.013550       -0.029812
DEP_DEL15                            0.016289        0.117528
DEP_TIME_BLK                        -0.026919        0.743527
DISTANCE_GROUP                       1.000000       -0.237415
SEGMENT_NUMBER                      -0.237415        1.000000
```

CONCURRENT_FLIGHTS	-0.035572	0.014240
NUMBER_OF_SEATS	0.447485	-0.202832
CARRIER_NAME	-0.054989	0.075980
AIRPORT_FLIGHTS_MONTH	-0.013700	0.042633
AIRLINE_FLIGHTS_MONTH	-0.003279	0.027261
AIRLINE_AIRPORT_FLIGHTS_MONTH	-0.035004	0.056441
AVG_MONTHLY_PASS_AIRPORT	0.037927	0.025410
AVG_MONTHLY_PASS_AIRLINE	0.126656	-0.057568
FLT_ATTENDANTS_PER_PASS	0.167134	-0.098353
GROUND_SERV_PER_PASS	0.266503	-0.183047
PLANE_AGE	-0.138314	0.076003
DEPARTING_AIRPORT	0.104267	-0.036197
LATITUDE	-0.011182	-0.034347
LONGITUDE	-0.159929	-0.070426
PREVIOUS_AIRPORT	0.081160	-0.125193
PRCP	-0.012478	-0.016279
SNOW	-0.000485	-0.014711
SNWD	-0.007388	-0.006888
TMAX	0.003697	0.029219
AWND	0.023392	-0.024107

	CONCURRENT_FLIGHTS	NUMBER_OF_SEATS	\
MONTH	0.022951	0.003155	
DAY_OF_WEEK	-0.027214	0.009300	
DEP_DEL15	0.009028	0.011845	
DEP_TIME_BLK	0.055996	-0.021984	
DISTANCE_GROUP	-0.035572	0.447485	
SEGMENT_NUMBER	0.014240	-0.202832	
CONCURRENT_FLIGHTS	1.000000	-0.054131	
NUMBER_OF_SEATS	-0.054131	1.000000	
CARRIER_NAME	-0.132342	-0.049387	
AIRPORT_FLIGHTS_MONTH	0.849023	0.003574	
AIRLINE_FLIGHTS_MONTH	-0.043085	0.308104	
AIRLINE_AIRPORT_FLIGHTS_MONTH	0.582488	0.161800	
AVG_MONTHLY_PASS_AIRPORT	0.808717	0.063401	
AVG_MONTHLY_PASS_AIRLINE	-0.017998	0.557014	
FLT_ATTENDANTS_PER_PASS	0.122495	0.204501	
GROUND_SERV_PER_PASS	0.103521	0.343823	
PLANE_AGE	0.036077	-0.102969	
DEPARTING_AIRPORT	-0.364696	0.058096	
LATITUDE	0.018230	-0.136366	
LONGITUDE	0.118662	-0.161854	
PREVIOUS_AIRPORT	0.009078	0.051274	
PRCP	-0.015074	-0.014044	
SNOW	-0.017250	-0.008511	
SNWD	-0.027321	-0.015707	
TMAX	0.026031	0.061254	
AWND	0.059272	-0.018506	

```
                                            CARRIER_NAME  
AIRPORT_FLIGHTS_MONTH  ...  \
MONTH                                           0.000090  
0.036913  ...   
DAY_OF_WEEK                                    -0.001988               -
0.001725  ...   
DEP_DEL15                                       0.016082  
0.026740  ...   
DEP_TIME_BLK                                    0.012839  
0.107640  ...   
DISTANCE_GROUP                                 -0.054989               -
0.013700  ...   
SEGMENT_NUMBER                                  0.075980  
0.042633  ...   
CONCURRENT_FLIGHTS                             -0.132342  
0.849023  ...   
NUMBER_OF_SEATS                                -0.049387  
0.003574  ...   
CARRIER_NAME                                    1.000000               -
0.134015  ...   
AIRPORT_FLIGHTS_MONTH                          -0.134015  
1.000000  ...   
AIRLINE_FLIGHTS_MONTH                           0.227784               -
0.019744  ...   
AIRLINE_AIRPORT_FLIGHTS_MONTH                  -0.150674  
0.648140  ...   
AVG_MONTHLY_PASS_AIRPORT                       -0.120333  
0.967896  ...   
AVG_MONTHLY_PASS_AIRLINE                        0.062683  
0.012987  ...   
FLT_ATTENDANTS_PER_PASS                        -0.023047  
0.148205  ...   
GROUND_SERV_PER_PASS                           -0.171594  
0.107796  ...   
PLANE_AGE                                       0.043921  
0.027185  ...   
DEPARTING_AIRPORT                               0.126469               -
0.402696  ...   
LATITUDE                                       -0.021838  
0.019066  ...   
LONGITUDE                                      -0.057003  
0.091296  ...   
PREVIOUS_AIRPORT                                0.065909  
0.011825  ...   
PRCP                                           -0.013202               -
0.010380  ...   
SNOW                                            0.012916               -
0.009837  ...   
SNWD                                           -0.002202               -
```

```
0.032679  ...
TMAX                                  -0.000003
0.038819  ...
AWND                                   0.022458
0.071717  ...

                               PLANE_AGE  DEPARTING_AIRPORT  LATITUDE  \
MONTH                          -0.017344           0.000293  0.012913
DAY_OF_WEEK                    -0.005785           0.003262 -0.011503
DEP_DEL15                       0.006220          -0.007315  0.000490
DEP_TIME_BLK                    0.008759          -0.042453 -0.035409
DISTANCE_GROUP                 -0.138314           0.104267 -0.011182
SEGMENT_NUMBER                  0.076003          -0.036197 -0.034347
CONCURRENT_FLIGHTS              0.036077          -0.364696  0.018230
NUMBER_OF_SEATS                -0.102969           0.058096 -0.136366
CARRIER_NAME                    0.043921           0.126469 -0.021838
AIRPORT_FLIGHTS_MONTH           0.027185          -0.402696  0.019066
AIRLINE_FLIGHTS_MONTH           0.170682           0.036970 -0.027490
AIRLINE_AIRPORT_FLIGHTS_MONTH   0.115853          -0.348771 -0.066772
AVG_MONTHLY_PASS_AIRPORT        0.026380          -0.351535 -0.027297
AVG_MONTHLY_PASS_AIRLINE        0.216519           0.018572 -0.073788
FLT_ATTENDANTS_PER_PASS         0.198599          -0.046073 -0.022480
GROUND_SERV_PER_PASS            0.194035           0.004346 -0.036737
PLANE_AGE                       1.000000          -0.037521 -0.025906
DEPARTING_AIRPORT              -0.037521           1.000000  0.077955
LATITUDE                       -0.025906           0.077955  1.000000
LONGITUDE                       0.025182          -0.253362  0.124157
PREVIOUS_AIRPORT               -0.028820           0.018096  0.015349
PRCP                            0.006147          -0.026787  0.019578
```

```
SNOW                           -0.000623           0.007797  0.084096
SNWD                            0.002179          -0.034180  0.144921
TMAX                            0.005179           0.008240 -0.361213
AWND                           -0.001391          -0.060679  0.073188

                               LONGITUDE  PREVIOUS_AIRPORT  
PRCP  \
MONTH                          -0.004548         -0.003549 -0.005133
DAY_OF_WEEK                    -0.006120          0.005966  0.018205
DEP_DEL15                       0.027097         -0.013342  0.080277
DEP_TIME_BLK                   -0.004765         -0.081345 -0.005099
DISTANCE_GROUP                 -0.159929          0.081160 -0.012478
SEGMENT_NUMBER                 -0.070426         -0.125193 -0.016279
CONCURRENT_FLIGHTS              0.118662          0.009078 -0.015074
NUMBER_OF_SEATS                -0.161854          0.051274 -0.014044
CARRIER_NAME                   -0.057003          0.065909 -0.013202
AIRPORT_FLIGHTS_MONTH           0.091296          0.011825 -0.010380
AIRLINE_FLIGHTS_MONTH          -0.080197          0.008588 -0.008108
AIRLINE_AIRPORT_FLIGHTS_MONTH   0.071855          0.036431 -0.000840
AVG_MONTHLY_PASS_AIRPORT        0.027809          0.023237 -0.015256
AVG_MONTHLY_PASS_AIRLINE       -0.057783          0.006114 -0.005727
FLT_ATTENDANTS_PER_PASS         0.004883         -0.000169  0.002044
GROUND_SERV_PER_PASS           -0.099493          0.017294 -0.006006
PLANE_AGE                       0.025182         -0.028820  0.006147
DEPARTING_AIRPORT              -0.253362          0.018096 -0.026787
LATITUDE                        0.124157          0.015349  0.019578
LONGITUDE                       1.000000         -0.099495  0.096599
```

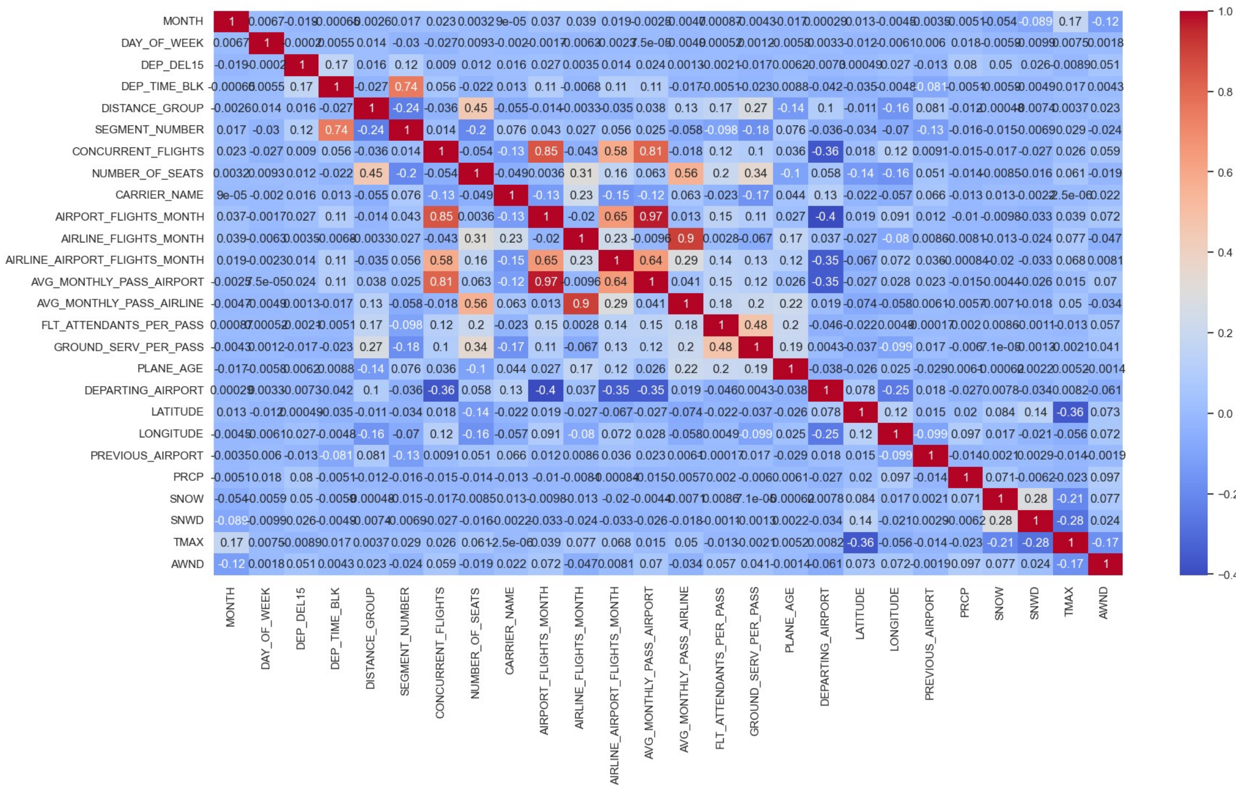
PREVIOUS_AIRPORT	-0.099495	1.000000	-0.013710
PRCP	0.096599	-0.013710	1.000000
SNOW	0.017035	0.002081	0.070900
SNWD	-0.020941	0.002901	-0.006215
TMAX	-0.056073	-0.014185	-0.022785
AWND	0.072104	-0.001912	0.096856

	SNOW	SNWD	TMAX	AWND
MONTH	-0.053596	-0.088596	0.173454	-0.119272
DAY_OF_WEEK	-0.005883	-0.009878	0.007513	0.001785
DEP_DEL15	0.050156	0.026129	-0.008936	0.050947
DEP_TIME_BLK	-0.005931	-0.004901	0.017053	0.004311
DISTANCE_GROUP	-0.000485	-0.007388	0.003697	0.023392
SEGMENT_NUMBER	-0.014711	-0.006888	0.029219	-0.024107
CONCURRENT_FLIGHTS	-0.017250	-0.027321	0.026031	0.059272
NUMBER_OF_SEATS	-0.008511	-0.015707	0.061254	-0.018506
CARRIER_NAME	0.012916	-0.002202	-0.000003	0.022458
AIRPORT_FLIGHTS_MONTH	-0.009837	-0.032679	0.038819	0.071717
AIRLINE_FLIGHTS_MONTH	-0.012901	-0.024357	0.076706	-0.047277
AIRLINE_AIRPORT_FLIGHTS_MONTH	-0.019745	-0.032805	0.067799	0.008097
AVG_MONTHLY_PASS_AIRPORT	-0.004358	-0.025802	0.014953	0.069800
AVG_MONTHLY_PASS_AIRLINE	-0.007138	-0.018442	0.049718	-0.033686
FLT_ATTENDANTS_PER_PASS	0.008563	-0.001053	-0.012604	0.057262
GROUND_SERV_PER_PASS	0.000071	-0.001275	-0.002075	0.041470
PLANE_AGE	-0.000623	0.002179	0.005179	-0.001391
DEPARTING_AIRPORT	0.007797	-0.034180	0.008240	-0.060679
LATITUDE	0.084096	0.144921	-0.361213	0.073188

```
LONGITUDE                                0.017035 -0.020941 -0.056073  0.072104
PREVIOUS_AIRPORT                         0.002081  0.002901 -0.014185 -0.001912
PRCP                                     0.070900 -0.006215 -0.022785  0.096856
SNOW                                     1.000000  0.284735 -0.207118  0.077381
SNWD                                     0.284735  1.000000 -0.280950  0.023618
TMAX                                    -0.207118 -0.280950  1.000000 -0.173037
AWND                                     0.077381  0.023618 -0.173037  1.000000

[26 rows x 26 columns]
```

Visualization of Correlation data using Matplotlib (Multivariate Analysis)

```python
# show correlation in a heatmap

%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns

# show the correlation in a plt figure
def show_correlation(fd):
    plt.figure(figsize=(20, 10))
    sns.set(style='whitegrid', context='notebook')
    sns.heatmap(fd.corr(), annot=True, square=False, cmap='coolwarm')
    plt.show()

# show the correlation
show_correlation(fd)
```



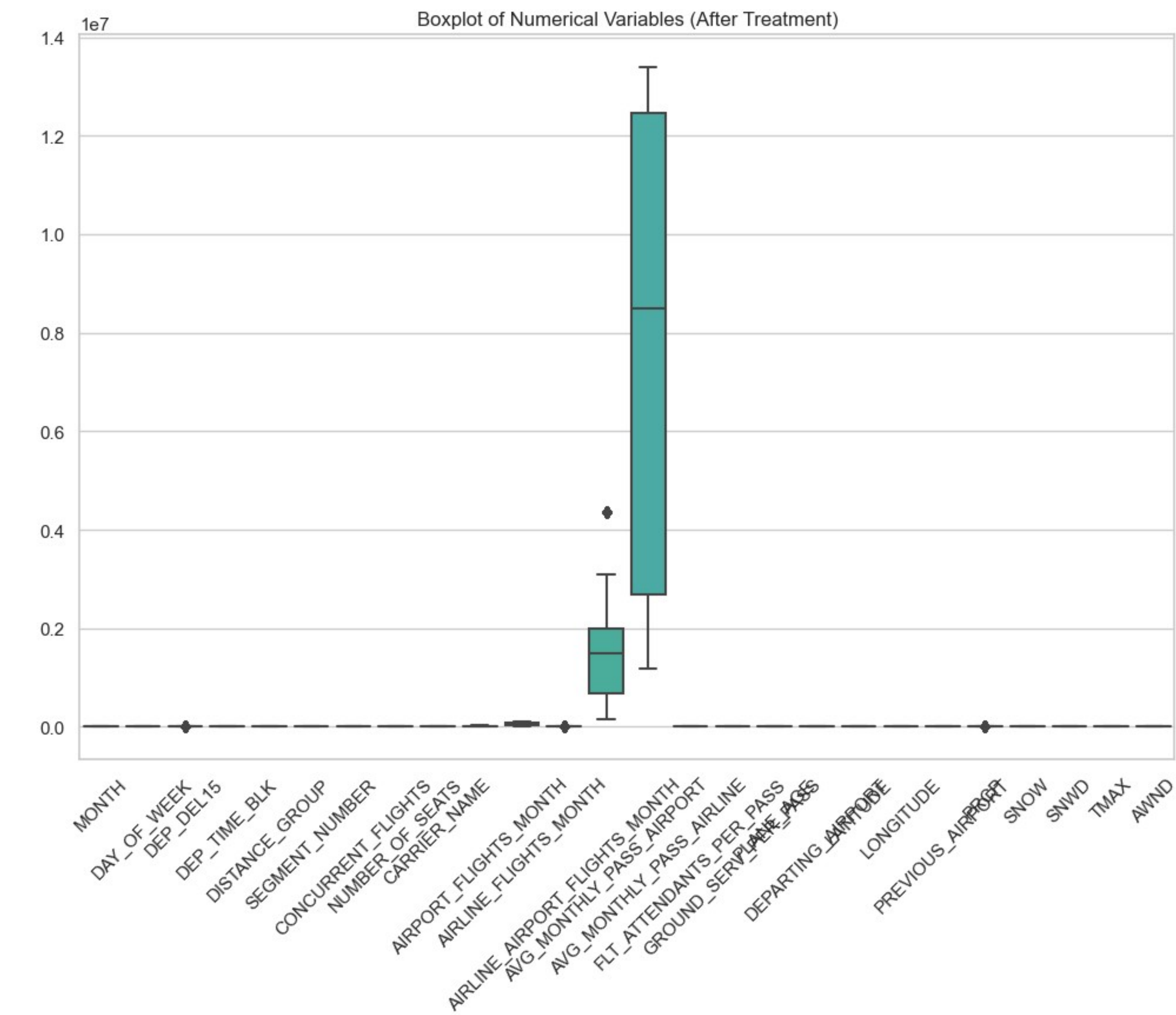
Outlier dectation and treatment

```
# Function to detect outliers using IQR method
def detect_outliers_iqr(fd, column):
    Q1 = fd[column].quantile(0.25)
    Q3 = fd[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = fd[(fd[column] < lower_bound) | (fd[column] >
upper_bound)]
    return outliers

# Detect outliers in numerical columns using IQR method
numerical_columns = fd.select_dtypes(include=['int64',
'float64']).columns
outliers = pd.DataFrame(columns=fd.columns)
for column in numerical_columns:
    outliers = pd.concat([outliers, detect_outliers_iqr(fd, column)])

print("Outliers detected using IQR method:")
print(outliers)

# Function to treat outliers using Winsorization
def treat_outliers_winsorization(fd, columns, lower_percentile=0.05,
upper_percentile=0.95):
```

```python
    for column in columns:
        lower_limit = fd[column].quantile(lower_percentile)
        upper_limit = fd[column].quantile(upper_percentile)
        fd[column] = np.where(fd[column] < lower_limit, lower_limit,
fd[column])
        fd[column] = np.where(fd[column] > upper_limit, upper_limit,
fd[column])

# Treat outliers using Winsorization
columns_to_treat = fd.select_dtypes(include=['int64',
'float64']).columns
treat_outliers_winsorization(fd, columns_to_treat)

# Visualize distributions after treatment
plt.figure(figsize=(12, 8))
sns.boxplot(data=fd)
plt.title('Boxplot of Numerical Variables (After Treatment)')
plt.xticks(rotation=45)
plt.show()
```

```
Outliers detected using IQR method:
        MONTH DAY_OF_WEEK DEP_DEL15 DEP_TIME_BLK DISTANCE_GROUP  \
7           1           7         1            0              7
10          1           7         1           18              6
15          1           7         1            2              4
24          1           7         1            5              3
36          1           7         1           17              9
...       ...         ...       ...          ...            ...
6487964    12           5         0           14              1
6487965    12           5         1           15              1
6487966    12           5         0           16              1
6487967    12           5         0           16              1
6487968    12           5         0           18              1

        SEGMENT_NUMBER CONCURRENT_FLIGHTS NUMBER_OF_SEATS CARRIER_NAME
\
7                    1                 10             186            8

10                   1                 17             180            8

15                   1                 29             181            0

24                   1                 29             142           16

36                   1                  9             162           10

...                ...                ...             ...          ...

6487964             10                  1             123            9
```

```
6487965             11                  2             123             9

6487966             12                  4             123             9

6487967             12                  4             123             9

6487968             13                  4             123             9


        AIRPORT_FLIGHTS_MONTH  ... PLANE_AGE DEPARTING_AIRPORT
LATITUDE  \
7                       13056  ...         3                44
36.080
10                      13056  ...         3                44
36.080
15                      13056  ...         4                44
36.080
24                      13056  ...        19                44
36.080
36                      13056  ...        15                44
36.080
...                       ...  ...       ...               ...       ..
.
6487964                  1318  ...        15                39
21.979
6487965                  1318  ...        18                39
21.979
6487966                  1318  ...        15                39
21.979
6487967                  1318  ...        18                39
21.979
6487968                  1318  ...        18                39
21.979

        LONGITUDE  PREVIOUS_AIRPORT  PRCP SNOW SNWD  TMAX   AWND
7        -115.152               216  0.00  0.0  0.0  65.0   2.91
10       -115.152               216  0.00  0.0  0.0  65.0   2.91
15       -115.152               216  0.00  0.0  0.0  65.0   2.91
24       -115.152               216  0.00  0.0  0.0  65.0   2.91
36       -115.152               216  0.00  0.0  0.0  65.0   2.91
...           ...               ...   ...  ...  ...   ...    ...
6487964  -159.346               136  0.44  0.0  0.0  78.0  25.72
6487965  -159.346               136  0.44  0.0  0.0  78.0  25.72
6487966  -159.346               136  0.44  0.0  0.0  78.0  25.72
6487967  -159.346               136  0.44  0.0  0.0  78.0  25.72
6487968  -159.346               136  0.44  0.0  0.0  78.0  25.72

[4932359 rows x 26 columns]
```



Principal Component Analysis

In the context of Principal Component Analysis (PCA), it's common practice to standardize the features before applying PCA. Here's why StandardScaler from sklearn.preprocessing and PCA from sklearn.decomposition are used:

StandardScaler: PCA is sensitive to the scale of the features. If the features have different scales, the ones with larger scales will dominate the ones with smaller scales when calculating principal components. Standardizing the features ensures that each feature has a mean of 0 and a standard deviation of 1, putting them on the same scale. This preprocessing step is crucial for PCA to work effectively and make meaningful comparisons between variables.

PCA: The PCA algorithm itself is implemented in the PCA class from the sklearn.decomposition module. This class provides methods for fitting the PCA model to the data, transforming the data into the principal components, and accessing various attributes such as the explained variance ratio and the principal components themselves. By using PCA from sklearn.decomposition, you can easily apply PCA to your dataset and analyze the results.

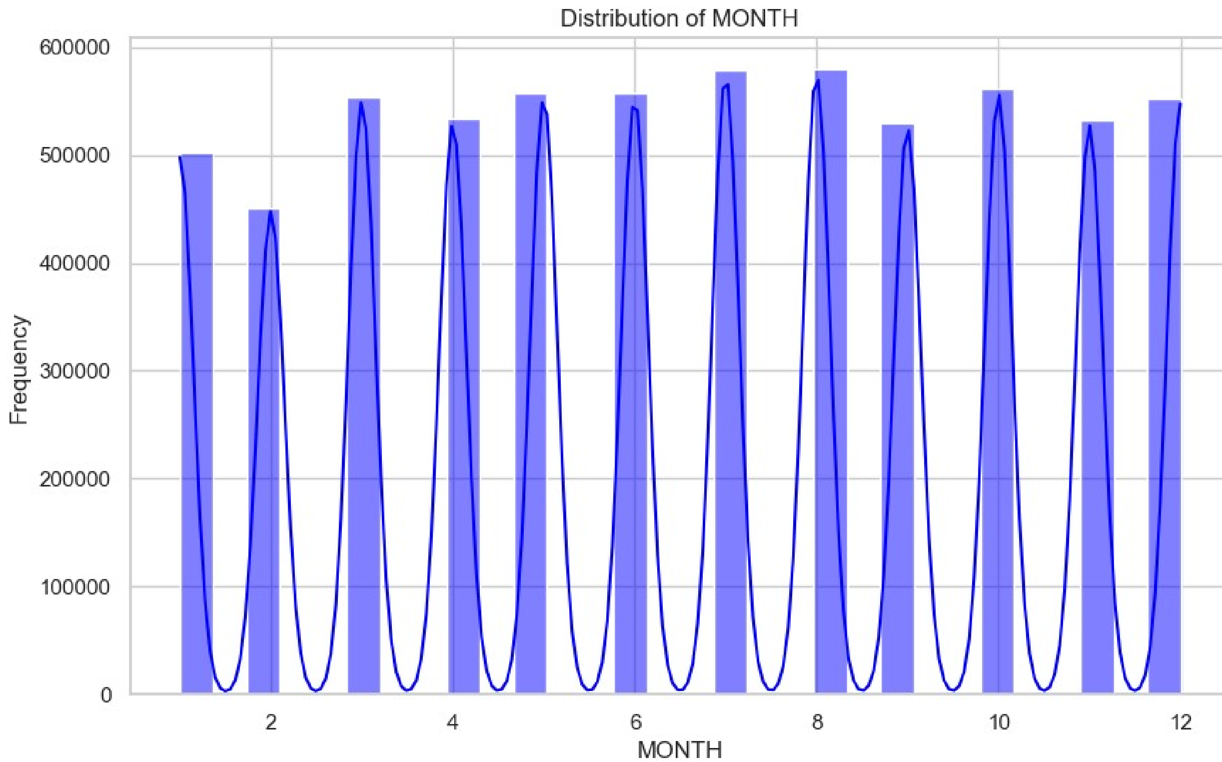
```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Standardize the features
scaler = StandardScaler()
scaled_features = scaler.fit_transform(fd)

# Perform PCA
pca = PCA()
pca.fit(scaled_features)

# Explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_

# Cumulative explained variance
cumulative_explained_variance = explained_variance_ratio.cumsum()

# Determine the number of components to retain
num_components_to_retain = sum(cumulative_explained_variance < 0.95) +
1  # Retain components explaining at least 95% of variance

print("Number of components to retain:", num_components_to_retain)

# Apply PCA transformation
pca = PCA(n_components=num_components_to_retain)
pca_features = pca.fit_transform(scaled_features)

# Convert PCA features to DataFrame
pca_df = pd.DataFrame(data=pca_features, columns=[f"PC{i+1}" for i in
range(num_components_to_retain)])

# Concatenate PCA features with original DataFrame
fd_with_pca = pd.concat([fd, pca_df], axis=1)

# Display the DataFrame with PCA features
print("DataFrame with PCA features:")
print(fd_with_pca.head())
```

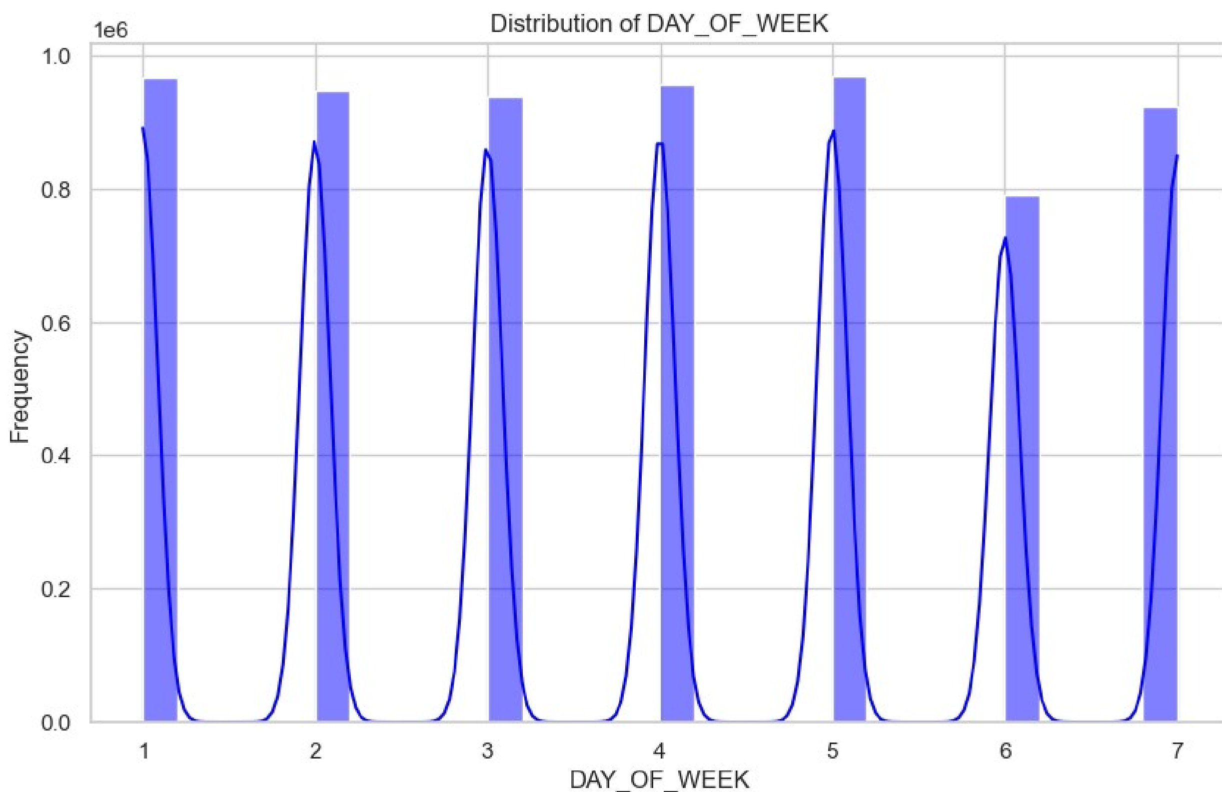
```
Number of components to retain: 18
DataFrame with PCA features:
   MONTH  DAY_OF_WEEK  DEP_DEL15  DEP_TIME_BLK  DISTANCE_GROUP  \
0    1.0          7.0        0.0           3.0             2.0
1    1.0          7.0        0.0           2.0             7.0
2    1.0          7.0        0.0           1.0             7.0
3    1.0          7.0        0.0           1.0             9.0
4    1.0          7.0        0.0           1.0             7.0

   SEGMENT_NUMBER  CONCURRENT_FLIGHTS  NUMBER_OF_SEATS
```

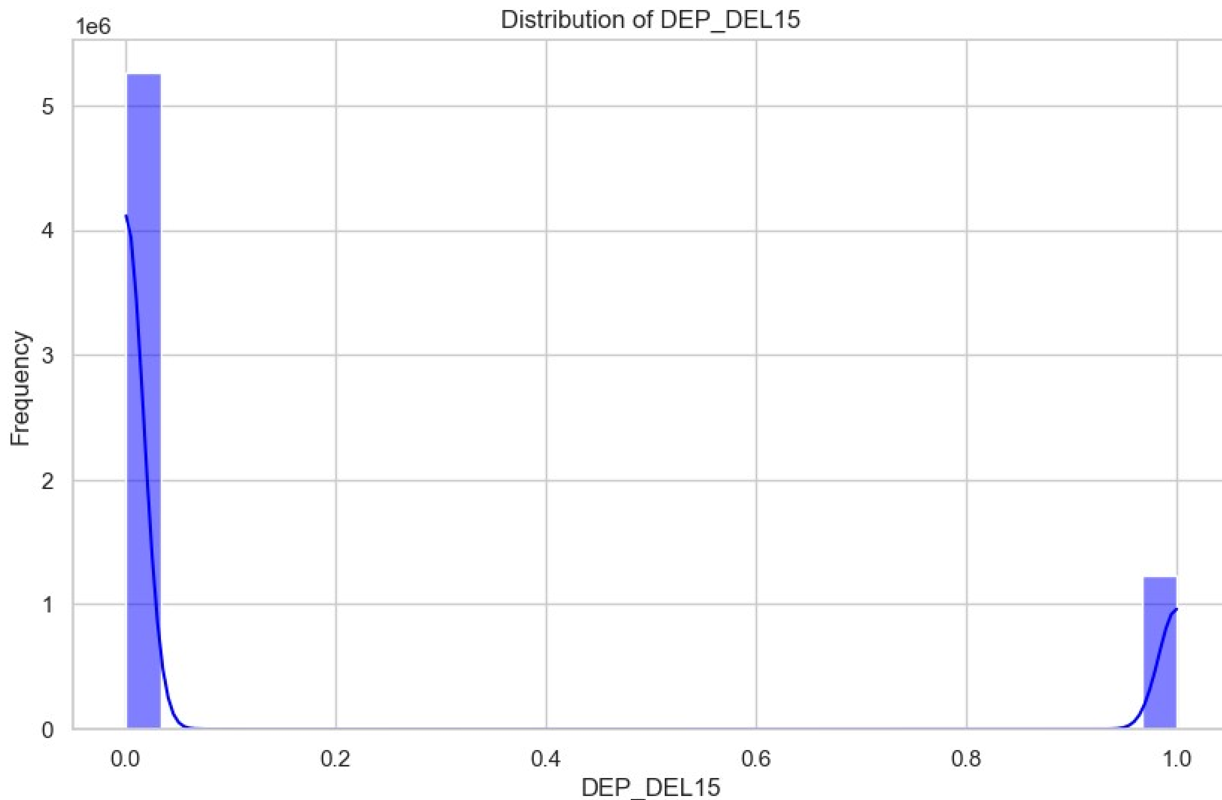
```
CARRIER_NAME  \
0             1.0                  25.0             143.0          14.0

1             1.0                  29.0             191.0           6.0

2             1.0                  27.0             191.0           6.0

3             1.0                  27.0             180.0           6.0

4             1.0                  10.0             182.0          15.0


   AIRPORT_FLIGHTS_MONTH  ...       PC9      PC10      PC11      PC12  
\
0                13056.0  ... -0.873042 -1.905185  0.654694  0.773188  

1                13056.0  ... -0.931177 -1.882322  0.783077  0.552134  

2                13056.0  ... -0.897892 -1.936484  1.134451  0.990293  

3                13056.0  ... -0.882538 -1.833846  0.671061  0.488233  

4                13056.0  ... -0.851744 -1.682603  0.118620  0.177374  


       PC13      PC14      PC15      PC16      PC17      PC18  
0 -0.440829  1.288903  0.589640  0.734997  0.423752  0.047473  
1 -0.464591  1.263954  0.298821 -0.042924  0.570927  0.687310  
2 -0.323958  1.011973  0.088184 -1.238702  0.391251  0.202072  
3 -0.596920  1.364600  0.094538 -0.613912  0.609237  1.041586  
4 -0.958284  1.865276  0.573383 -0.507441  0.478492 -0.953162  

[5 rows x 44 columns]
```

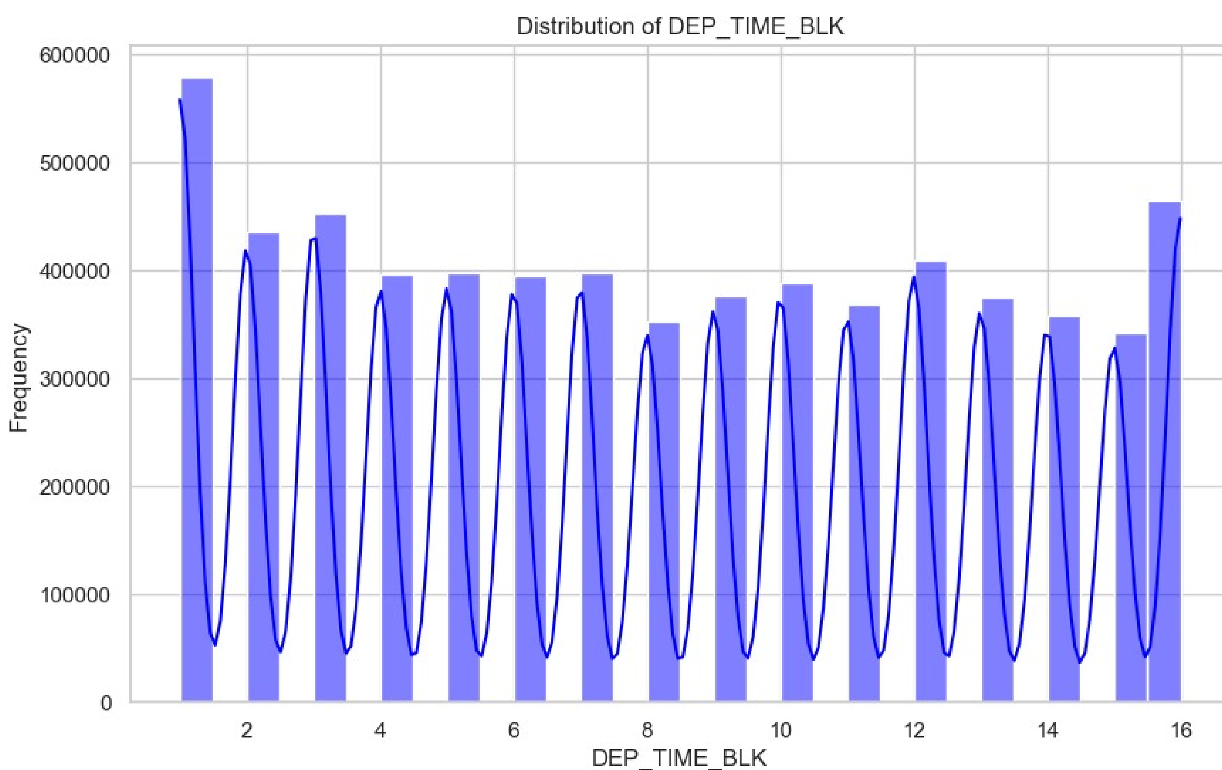
Univariate Analysis

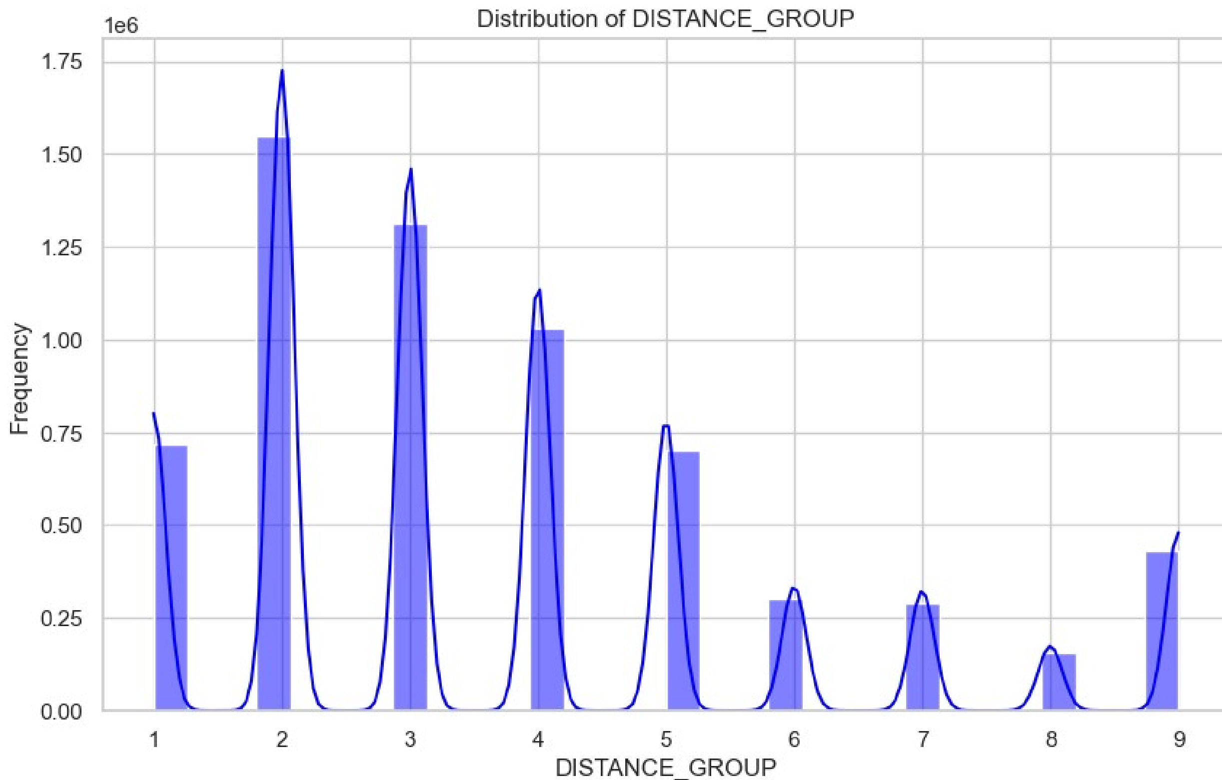
```python
# Example univariate analysis for a numerical variable
def univariate_analysis_numerical(fd):
    numerical_columns = fd.select_dtypes(include=['int64', 
'float64']).columns
    for column in numerical_columns:
        plt.figure(figsize=(10, 6))
        sns.histplot(fd[column], kde=True, color='blue', bins=30)
        plt.title(f'Distribution of {column}')
        plt.xlabel(column)
        plt.ylabel('Frequency')
        plt.show()

univariate_analysis_numerical(fd)
```

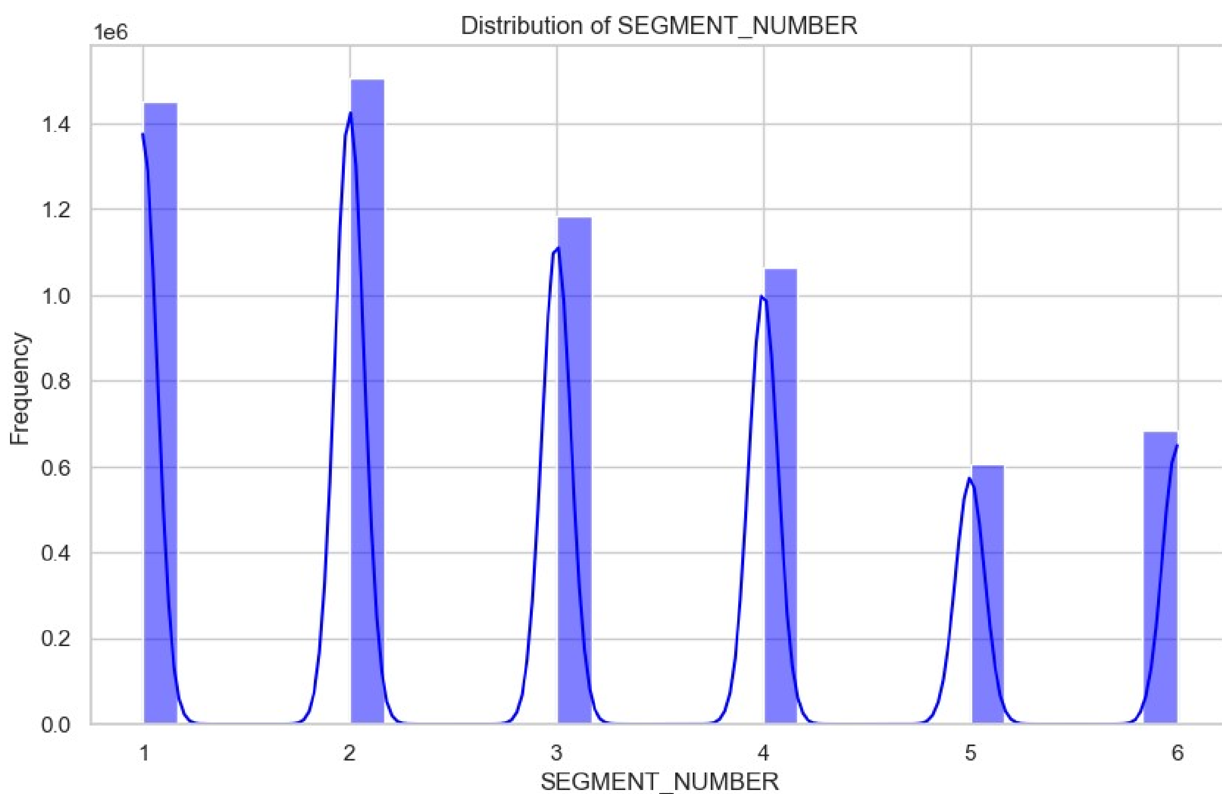

Distribution of MONTH
Frequency
MONTH
600000
500000
400000
300000
200000
100000
0
2
4
6
8
10
12

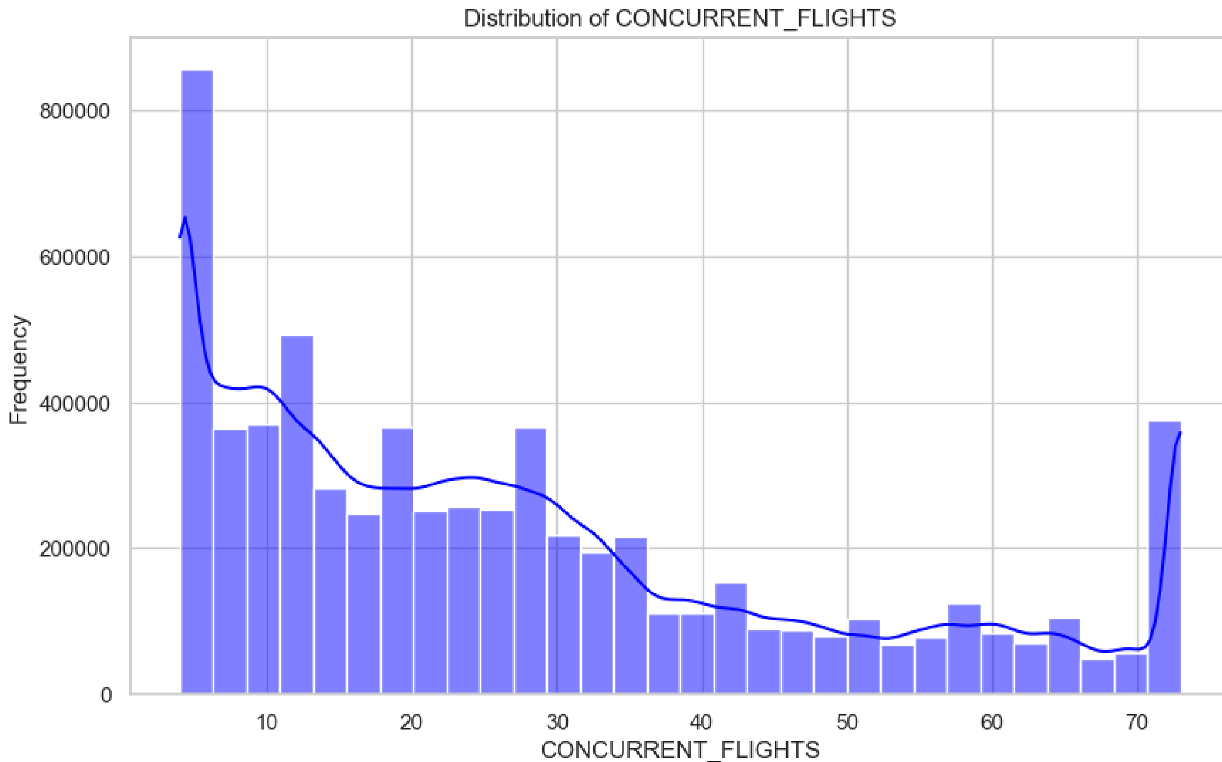

Distribution of DAY_OF_WEEK
1e6
Frequency
DAY_OF_WEEK
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
4
5
6
7

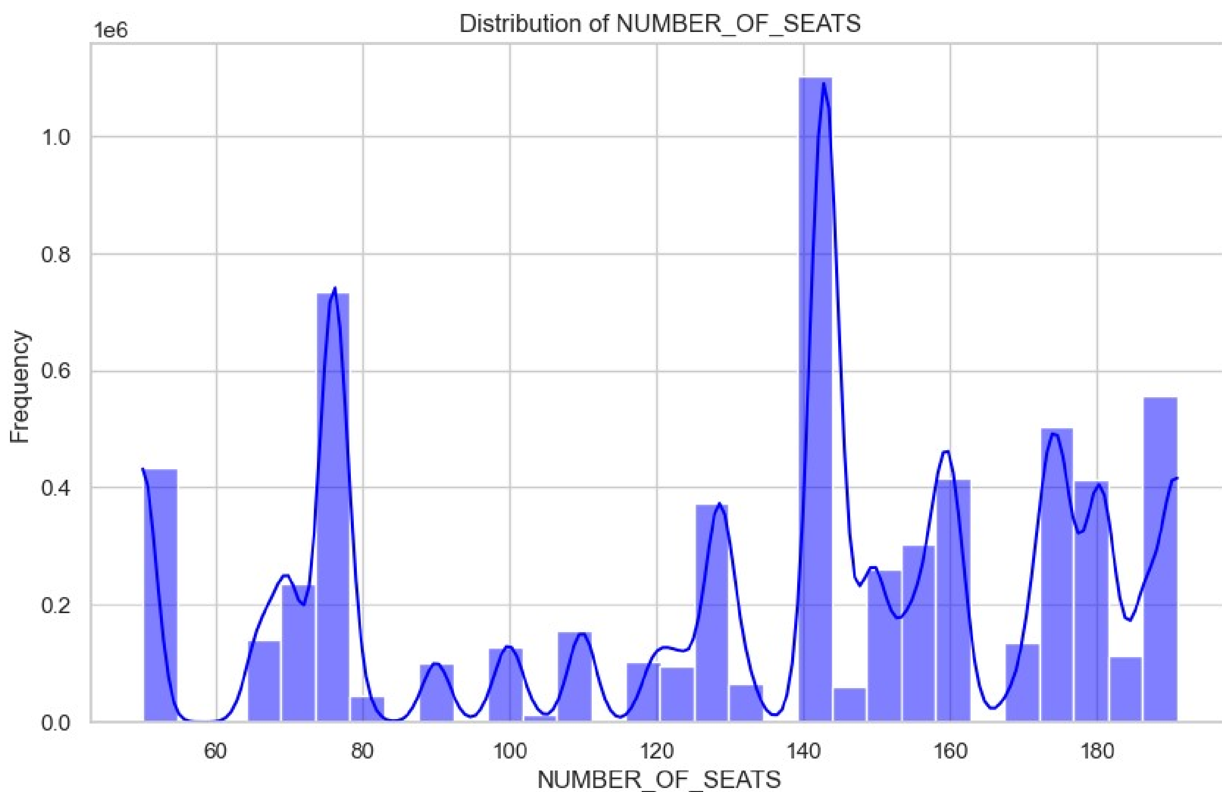

Distribution of DEP_DEL15
1e6
5
4
3
2
1
0
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
DEP_DEL15

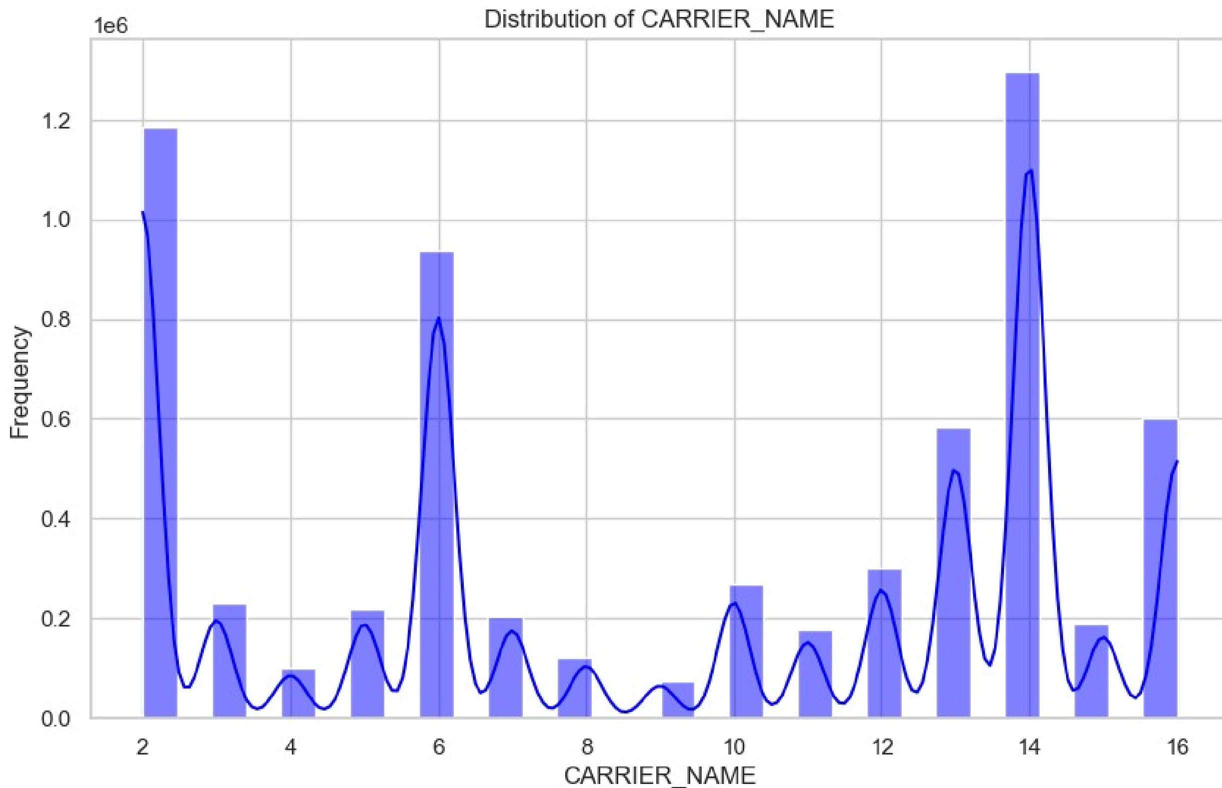

Distribution of DEP_TIME_BLK
600000
500000
400000
300000
200000
100000
0
Frequency
2
4
6
8
10
12
14
16
DEP_TIME_BLK

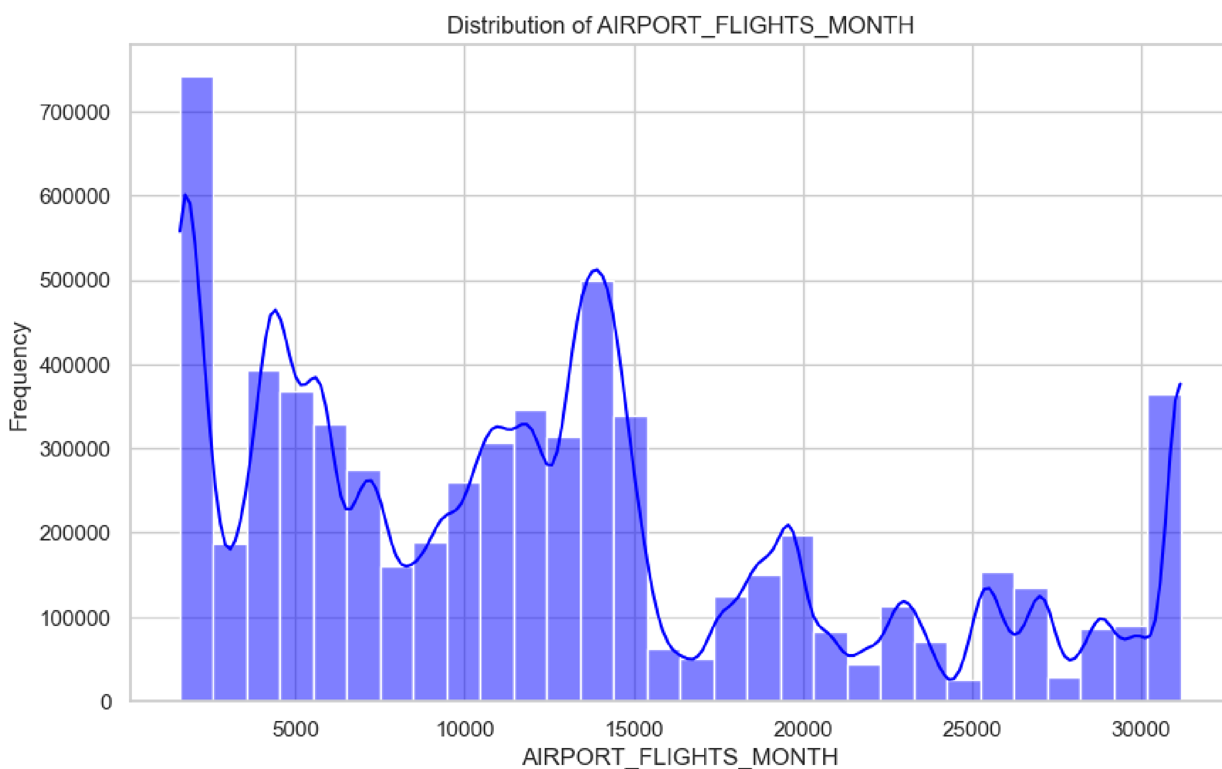

Distribution of DISTANCE_GROUP
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Frequency
1
2
3
4
5
6
7
8
9
DISTANCE_GROUP

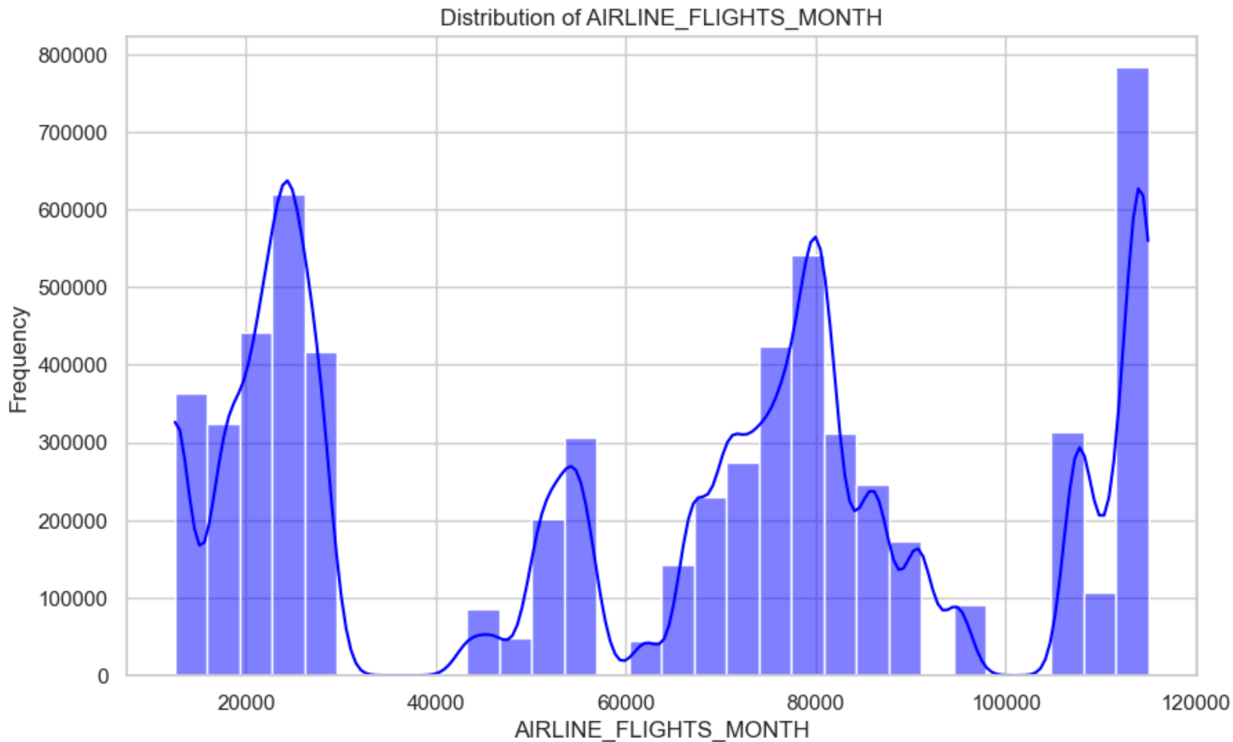

Distribution of SEGMENT_NUMBER
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
1
2
3
4
5
6
SEGMENT_NUMBER

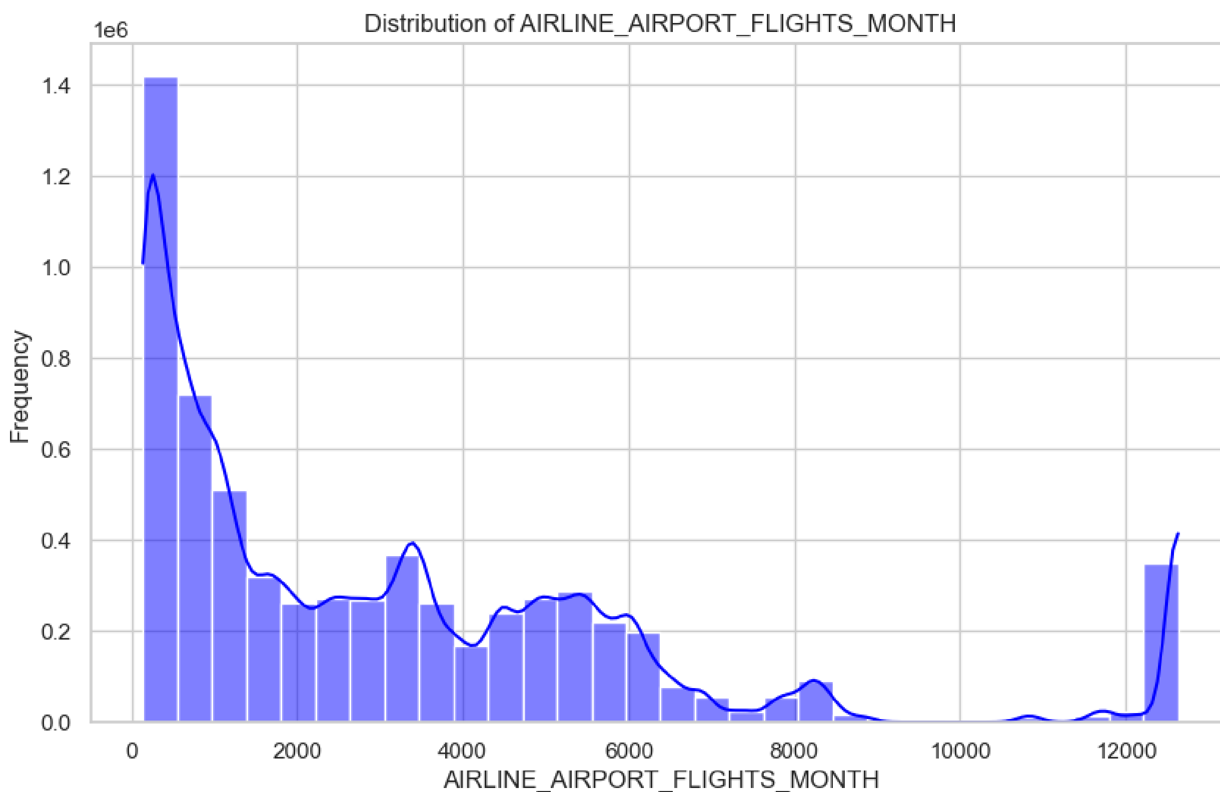

Distribution of CONCURRENT_FLIGHTS
Frequency
0
200000
400000
600000
800000
10
20
30
40
50
60
70
CONCURRENT_FLIGHTS

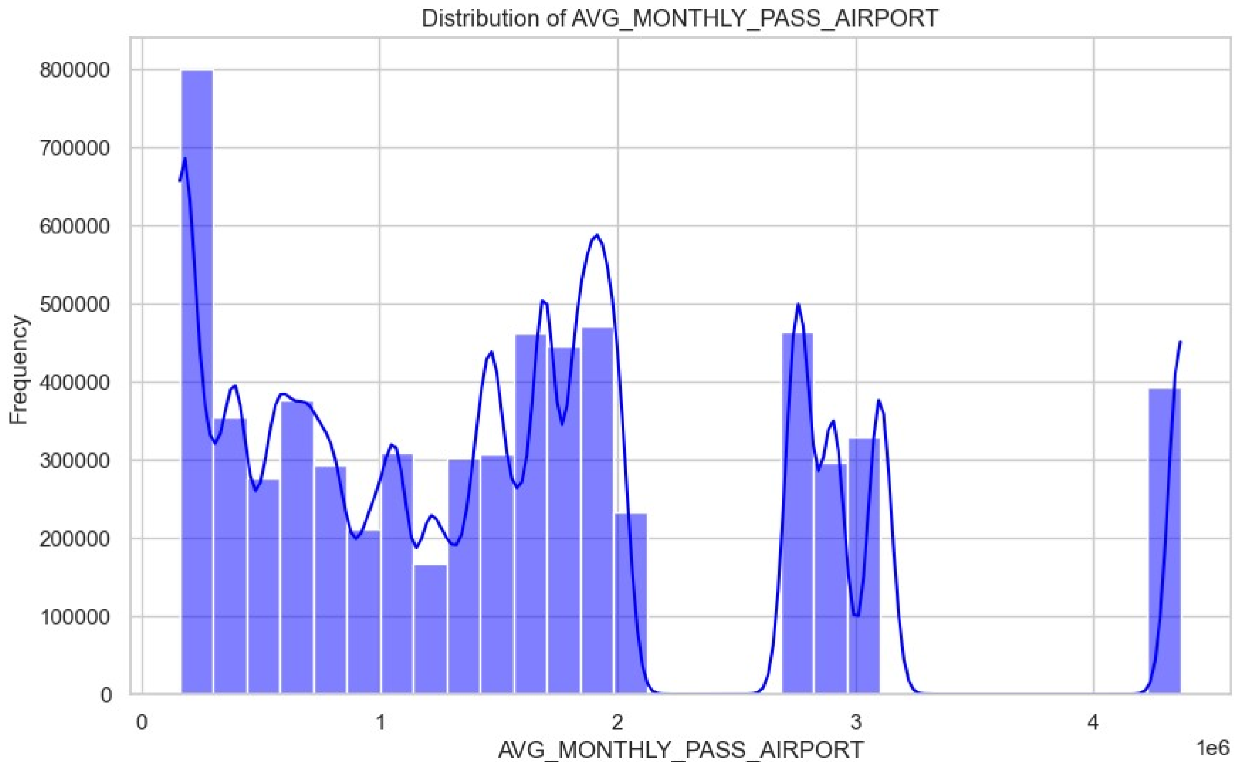

Distribution of NUMBER_OF_SEATS
1e6
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
60
80
100
120
140
160
180
NUMBER_OF_SEATS

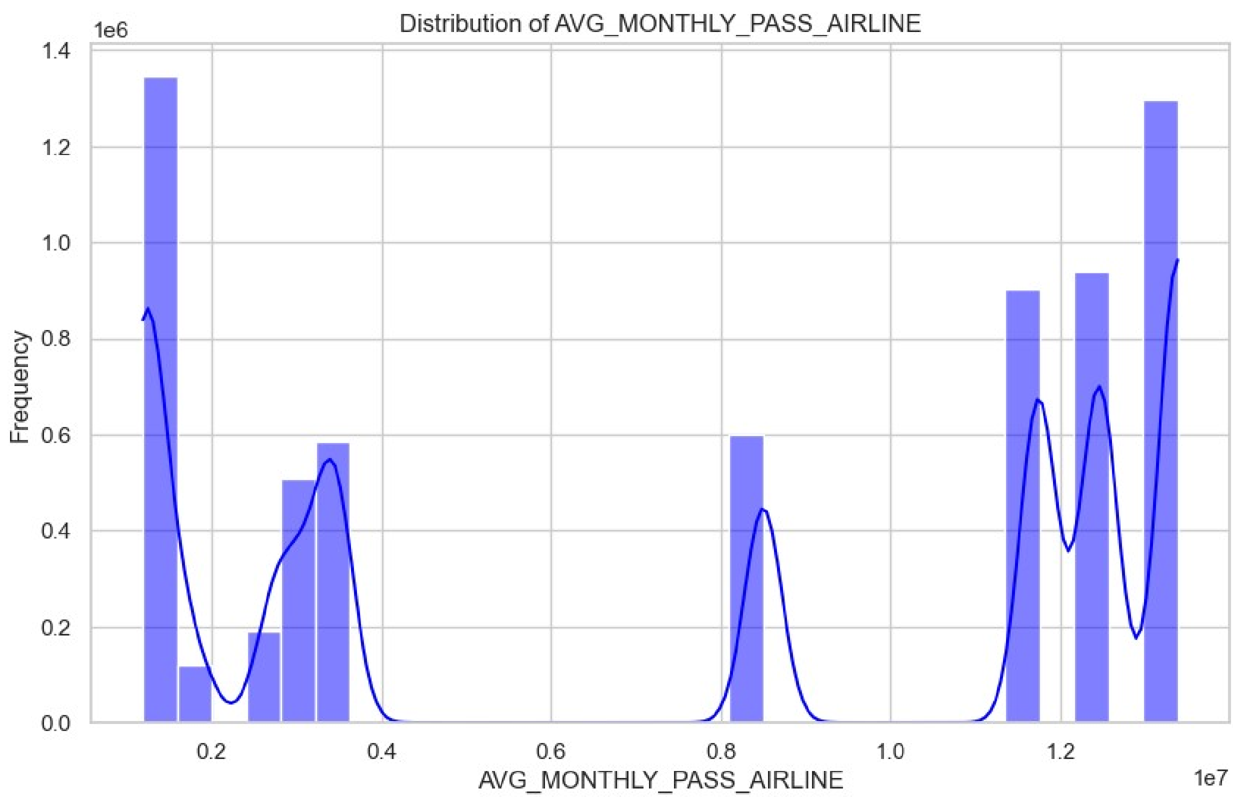

Distribution of CARRIER_NAME
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
2
4
6
8
10
12
14
16
CARRIER_NAME

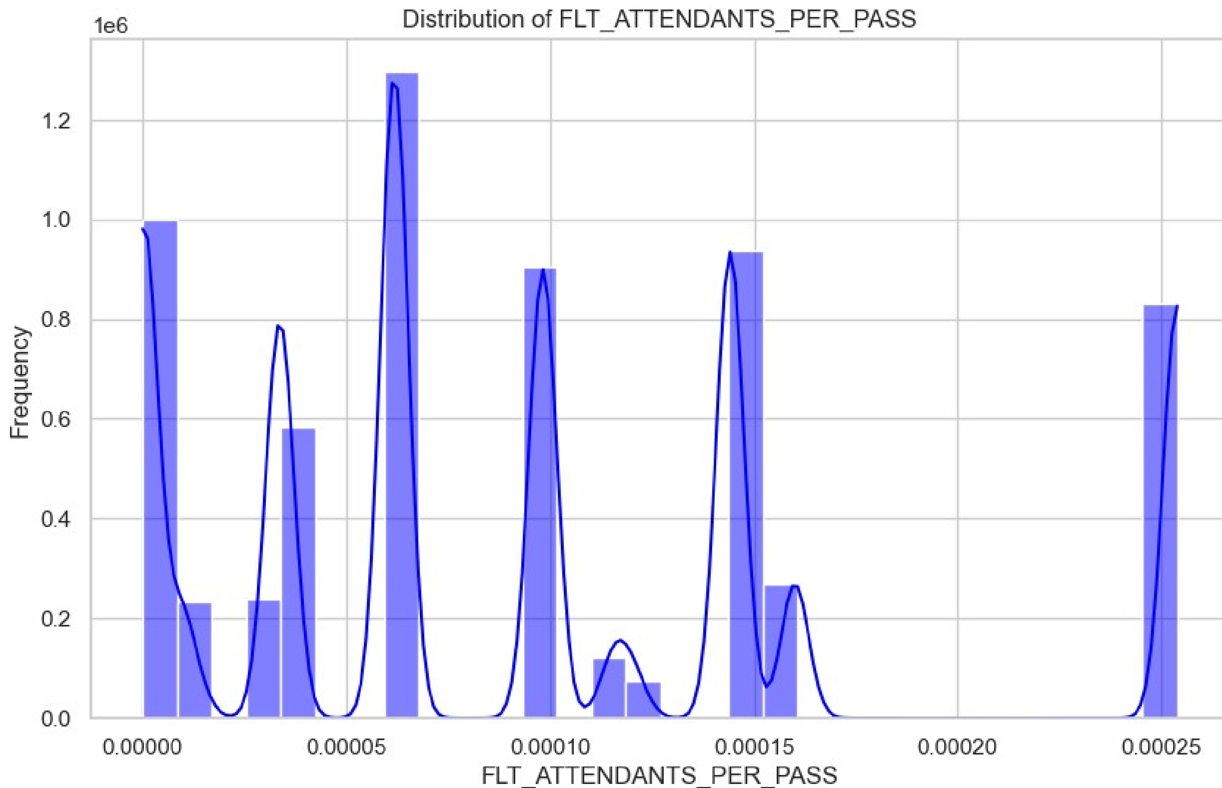

Distribution of AIRPORT_FLIGHTS_MONTH
700000
600000
500000
400000
300000
200000
100000
0
Frequency
5000
10000
15000
20000
25000
30000
AIRPORT_FLIGHTS_MONTH

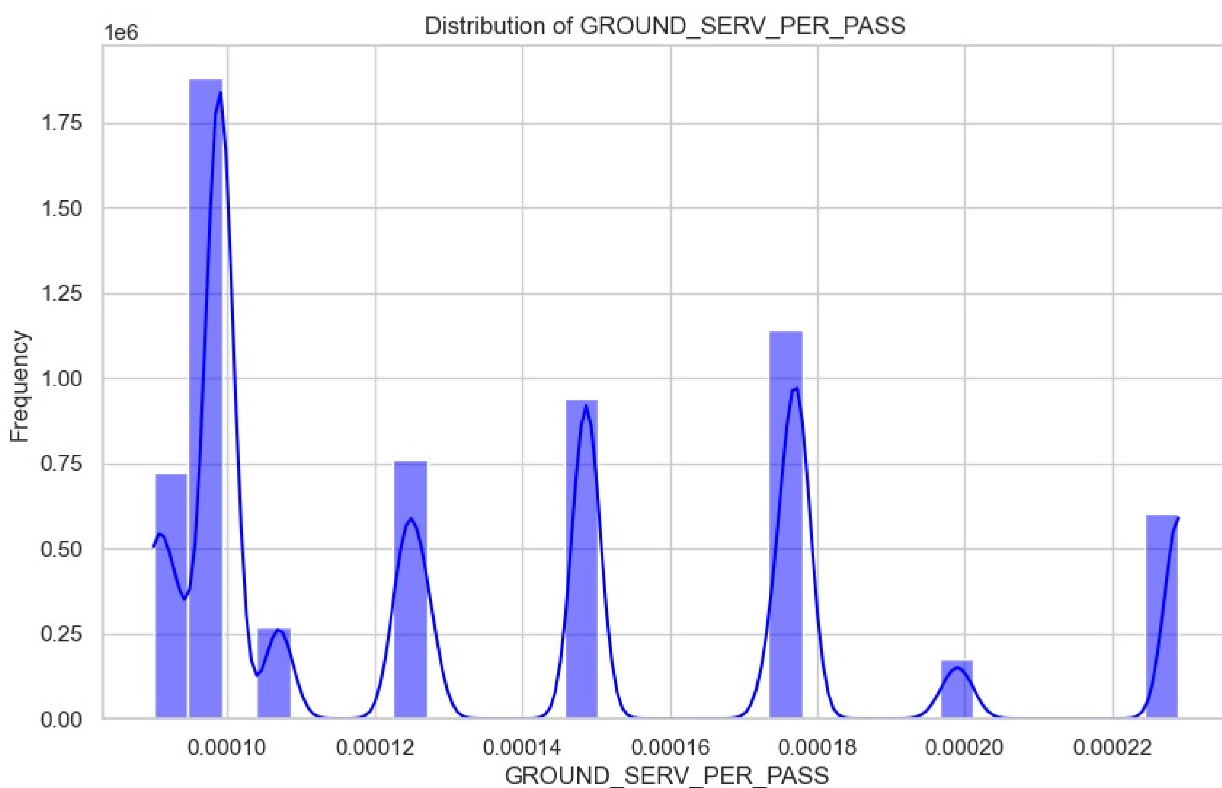

Distribution of AIRLINE_FLIGHTS_MONTH
800000
700000
600000
500000
400000
300000
200000
100000
0
Frequency
20000
40000
60000
80000
100000
120000
AIRLINE_FLIGHTS_MONTH

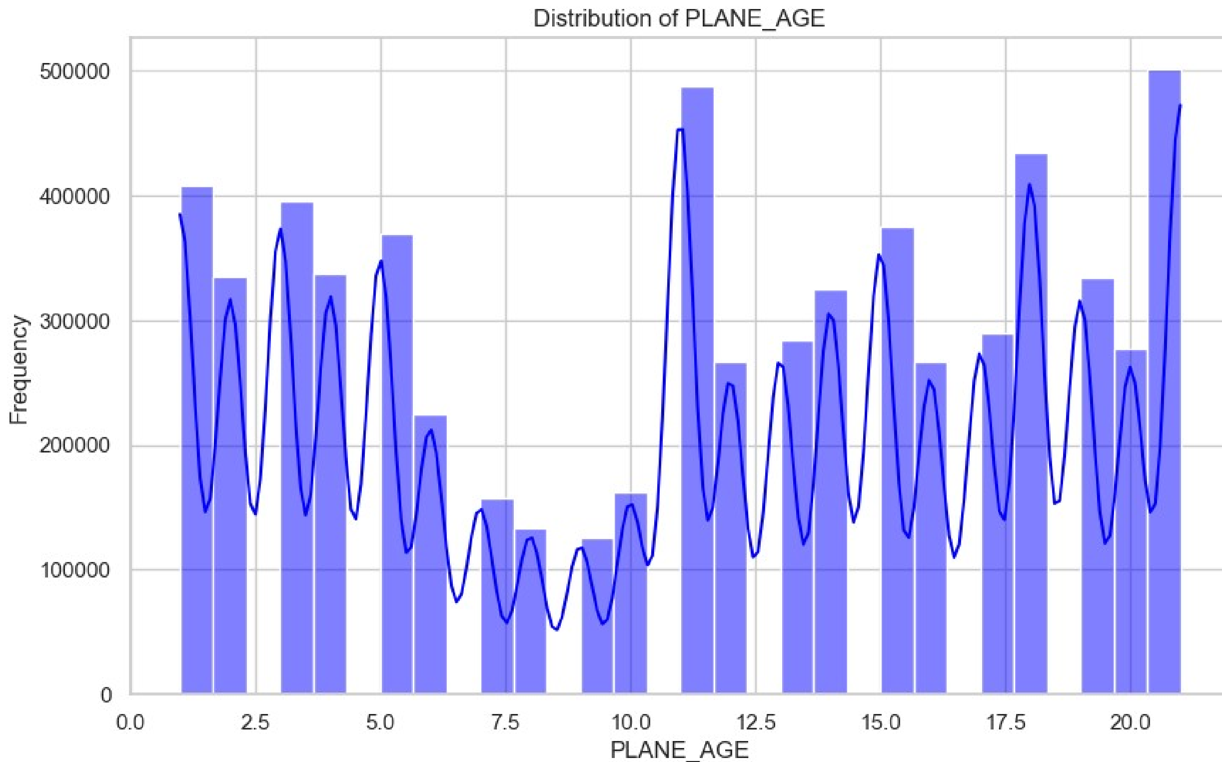

Distribution of AIRLINE_AIRPORT_FLIGHTS_MONTH
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
0
2000
4000
6000
8000
10000
12000
AIRLINE_AIRPORT_FLIGHTS_MONTH

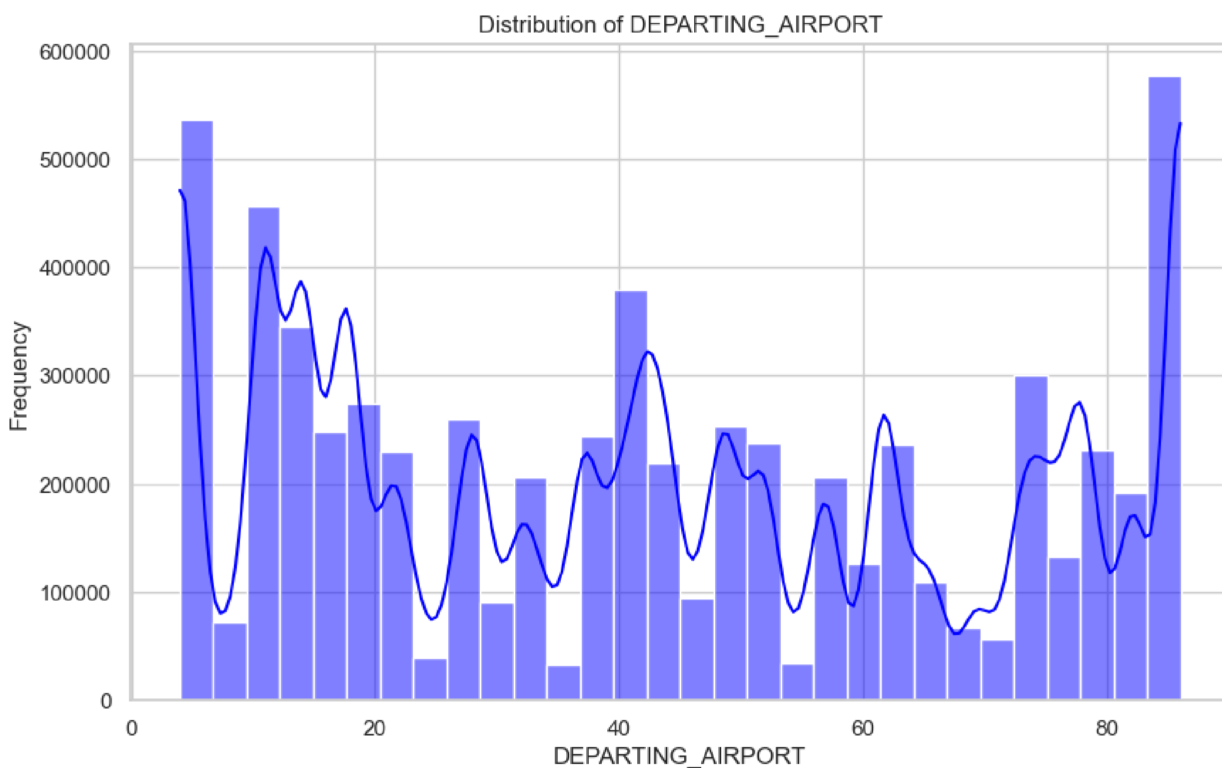

Distribution of AVG_MONTHLY_PASS_AIRPORT
Frequency
800000
700000
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
1e6
AVG_MONTHLY_PASS_AIRPORT

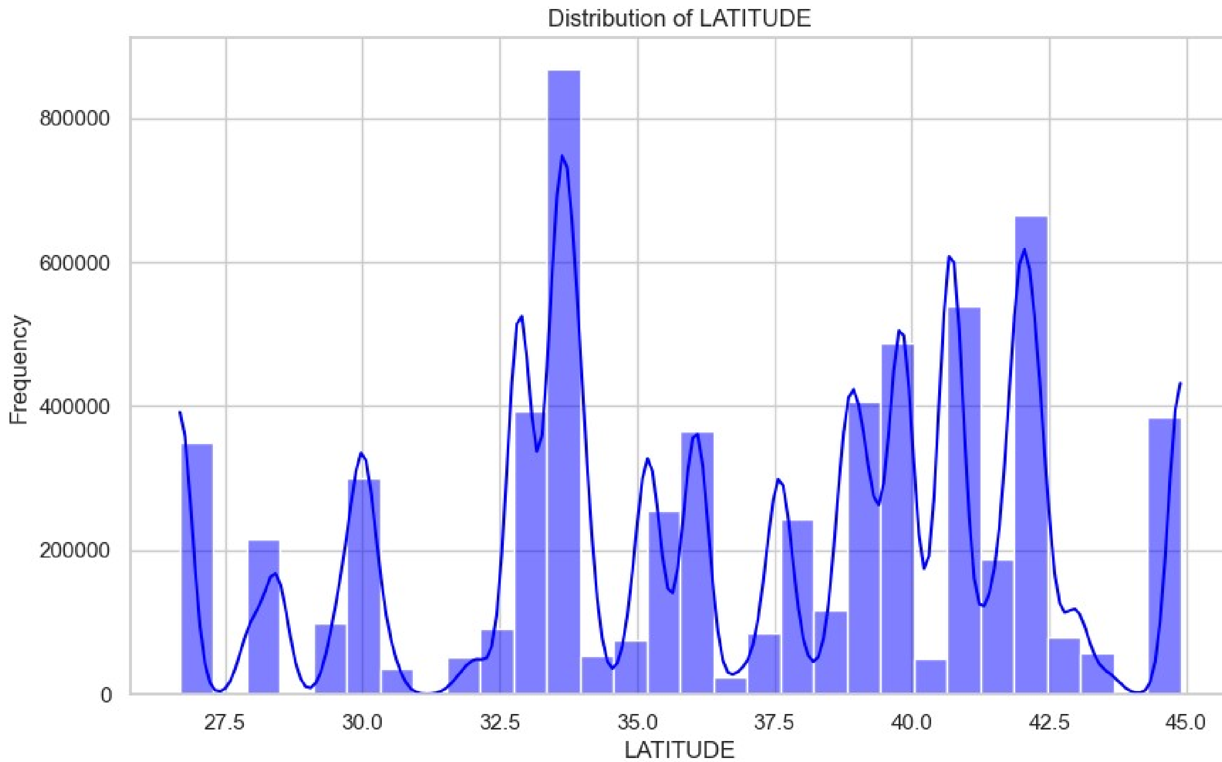

Distribution of AVG_MONTHLY_PASS_AIRLINE
1e6
Frequency
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e7
AVG_MONTHLY_PASS_AIRLINE

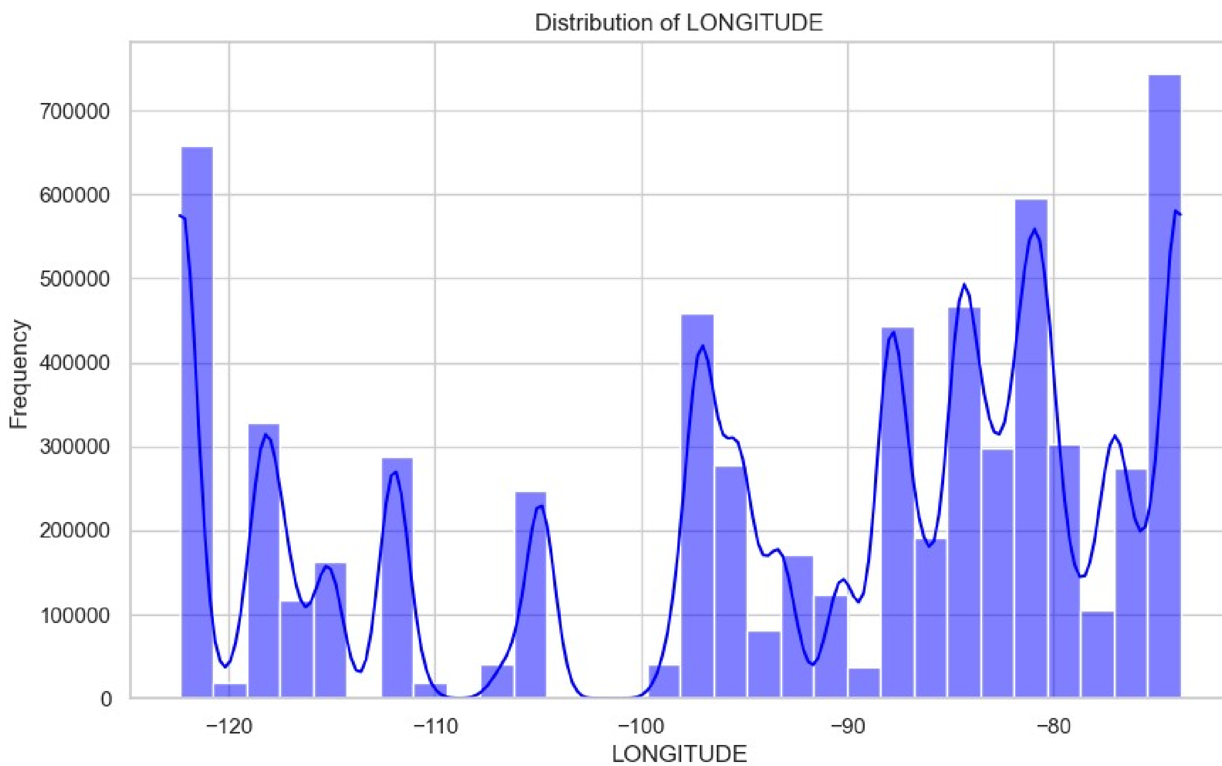

Distribution of FLT_ATTENDANTS_PER_PASS
1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
0.00000
0.00005
0.00010
0.00015
0.00020
0.00025
FLT_ATTENDANTS_PER_PASS

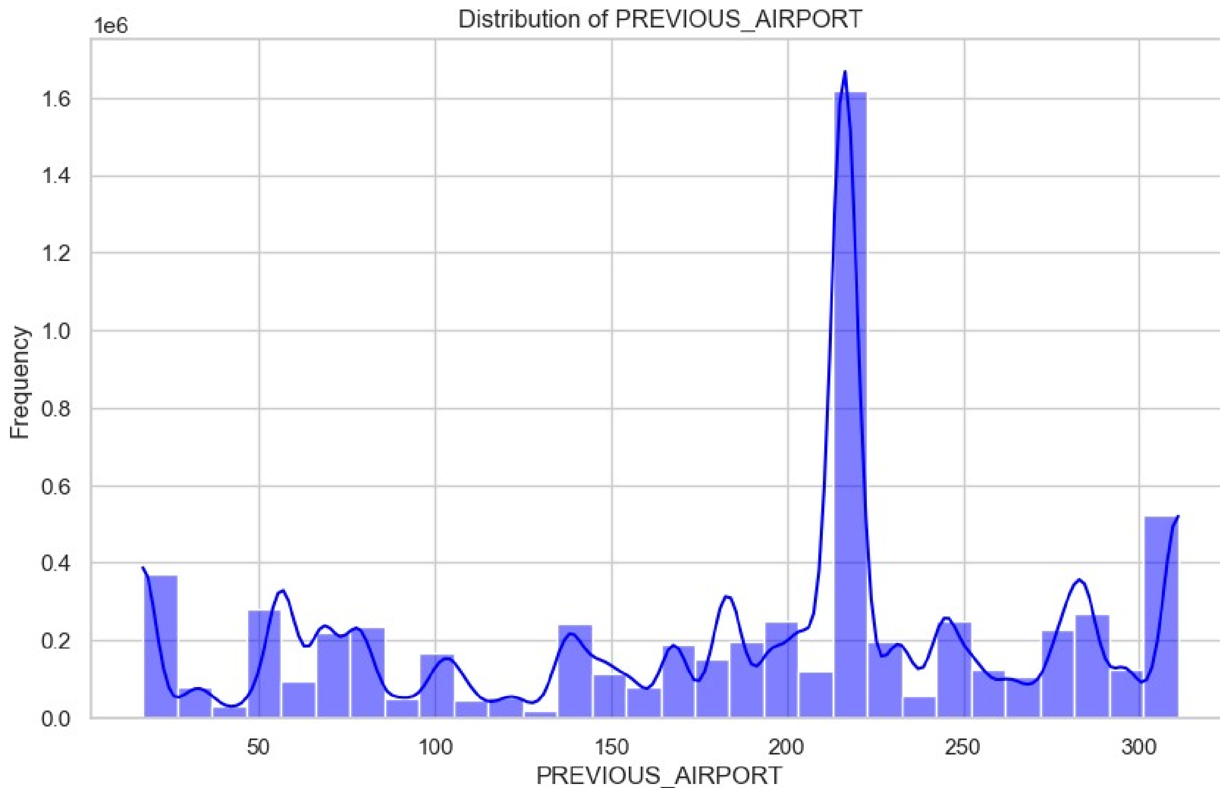

Distribution of GROUND_SERV_PER_PASS
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Frequency
0.00010
0.00012
0.00014
0.00016
0.00018
0.00020
0.00022
GROUND_SERV_PER_PASS

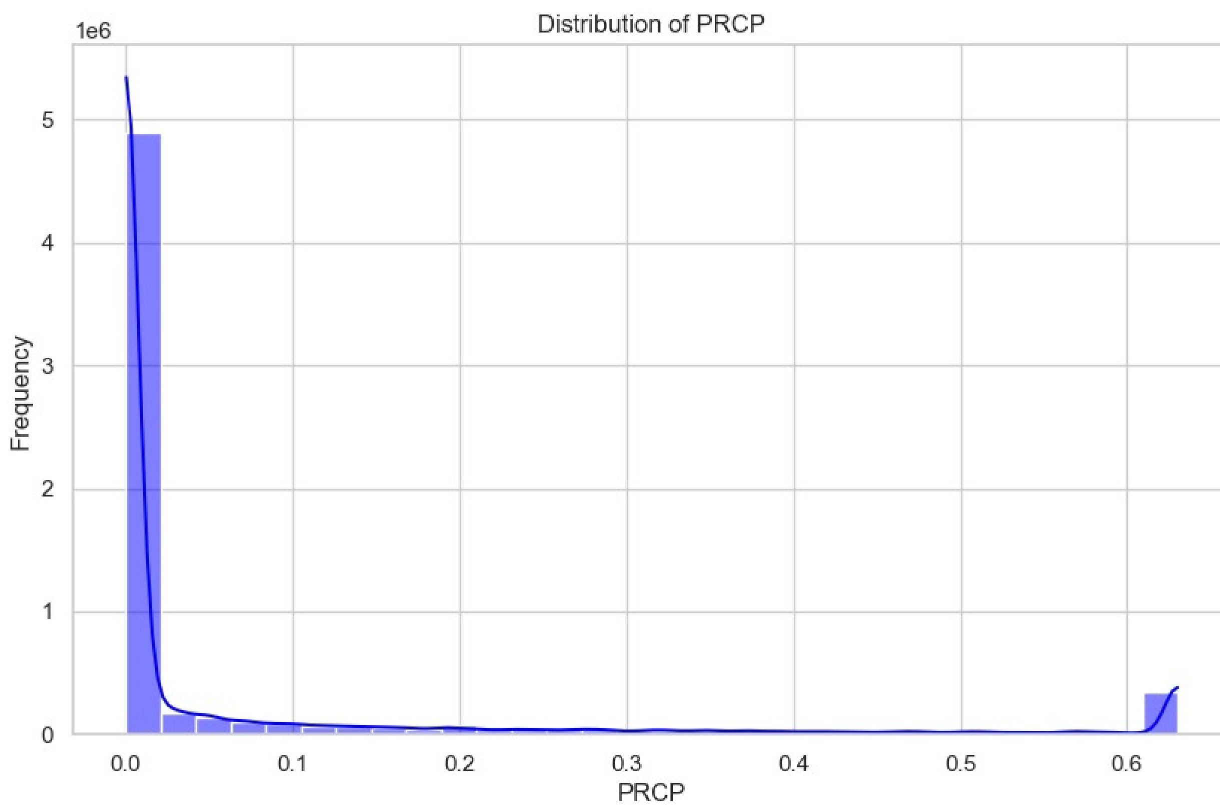

Distribution of PLANE_AGE
Frequency
0
100000
200000
300000
400000
500000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
PLANE_AGE

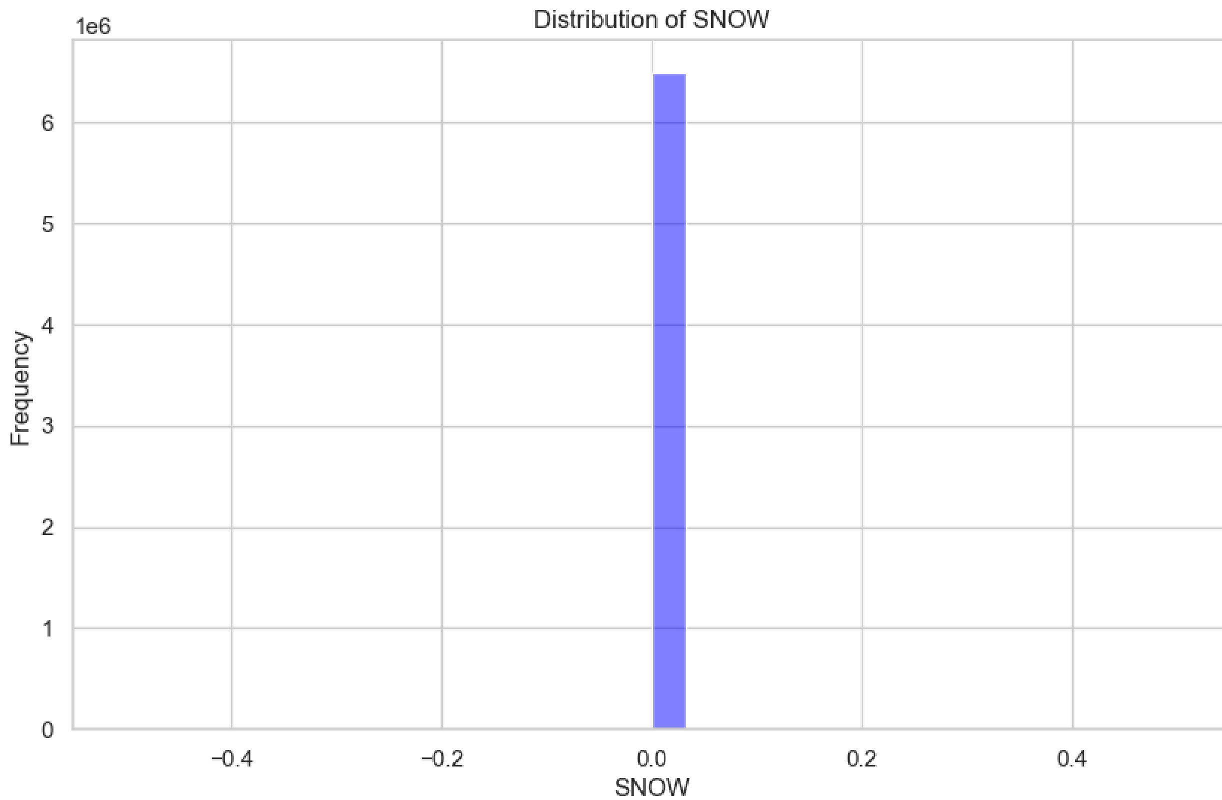

Distribution of DEPARTING_AIRPORT
Frequency
0
100000
200000
300000
400000
500000
600000
0
20
40
60
80
DEPARTING_AIRPORT

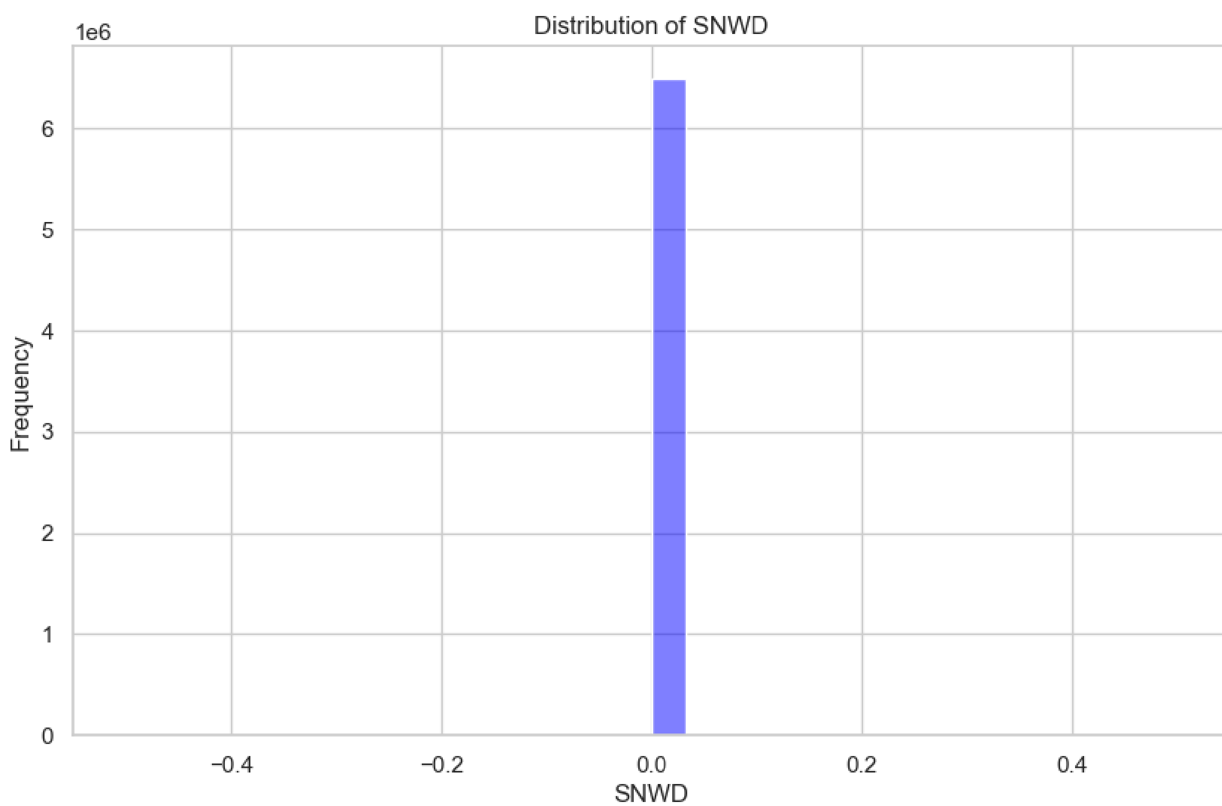

Distribution of LATITUDE
Frequency
0
200000
400000
600000
800000
27.5
30.0
32.5
35.0
37.5
40.0
42.5
45.0
LATITUDE

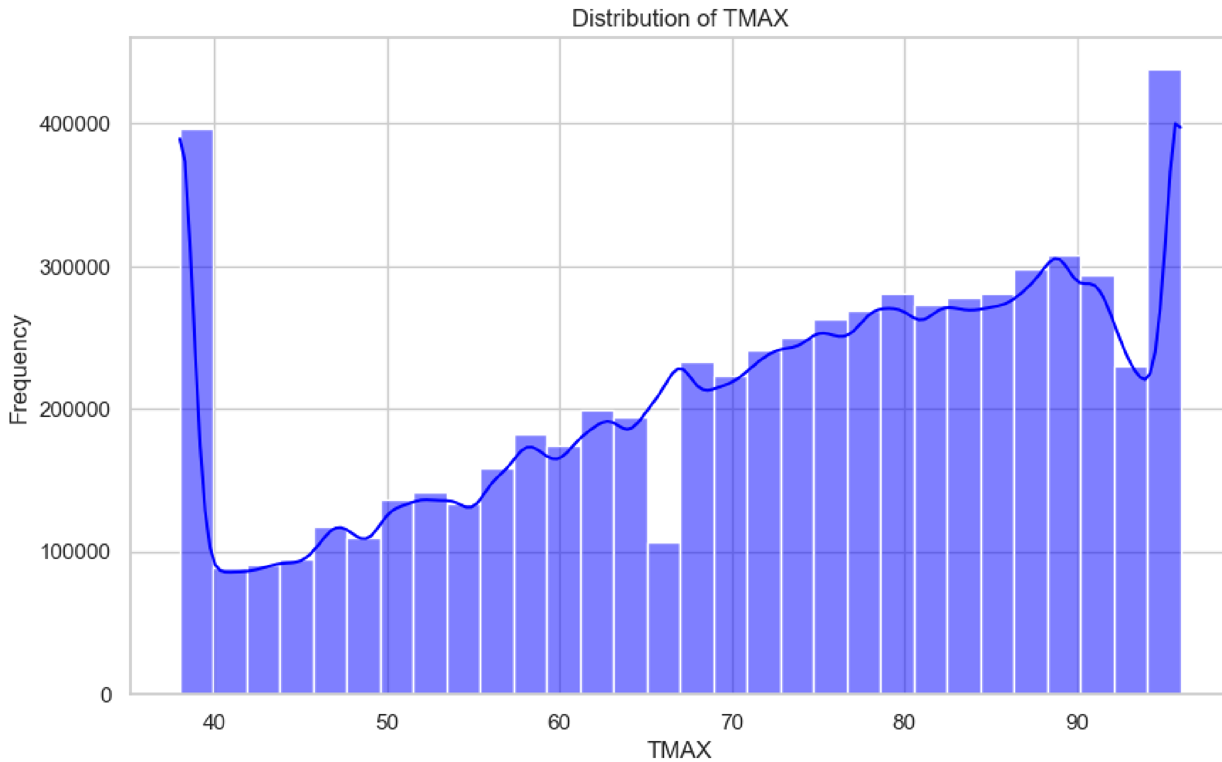

Distribution of LONGITUDE
Frequency
0
100000
200000
300000
400000
500000
600000
700000
−120
−110
−100
−90
−80
LONGITUDE

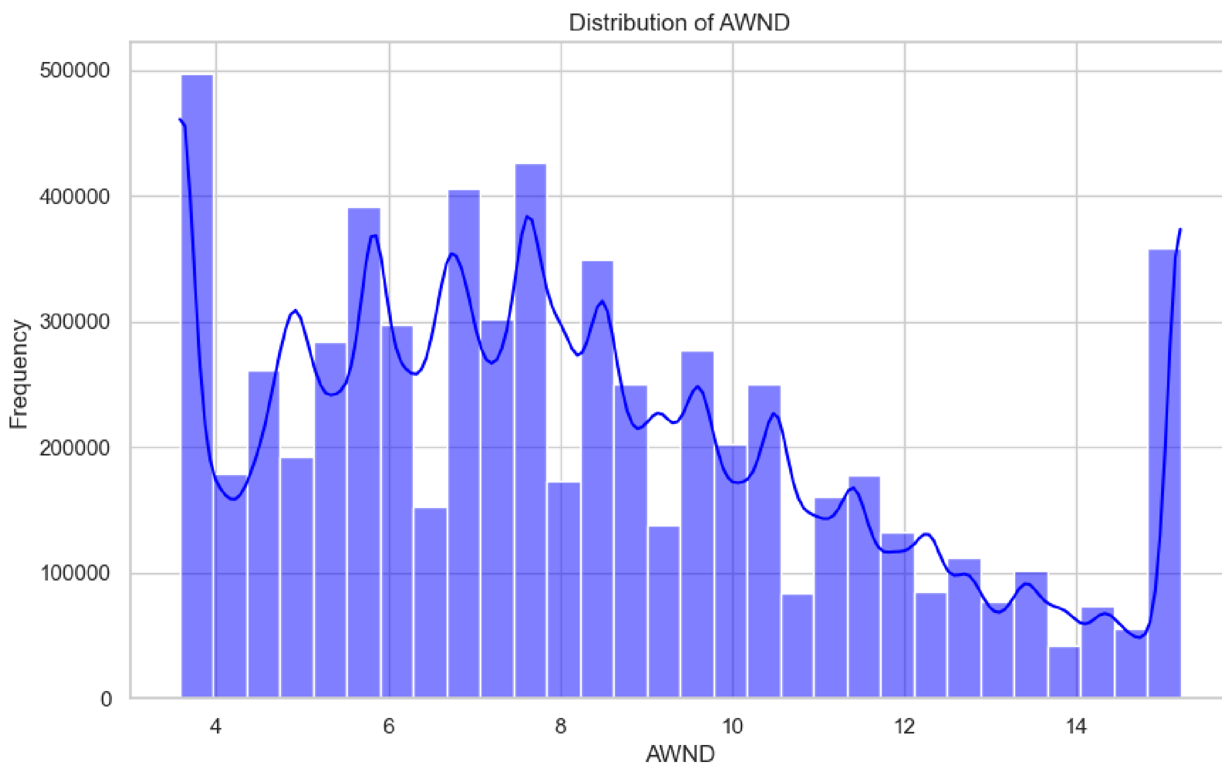

Distribution of PREVIOUS_AIRPORT
1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
50
100
150
200
250
300
PREVIOUS_AIRPORT


Distribution of PRCP
1e6
5
4
3
2
1
0
Frequency
0.0
0.1
0.2
0.3
0.4
0.5
0.6
PRCP


Distribution of SNOW
1e6
Frequency
0
1
2
3
4
5
6
−0.4
−0.2
0.0
0.2
0.4
SNOW

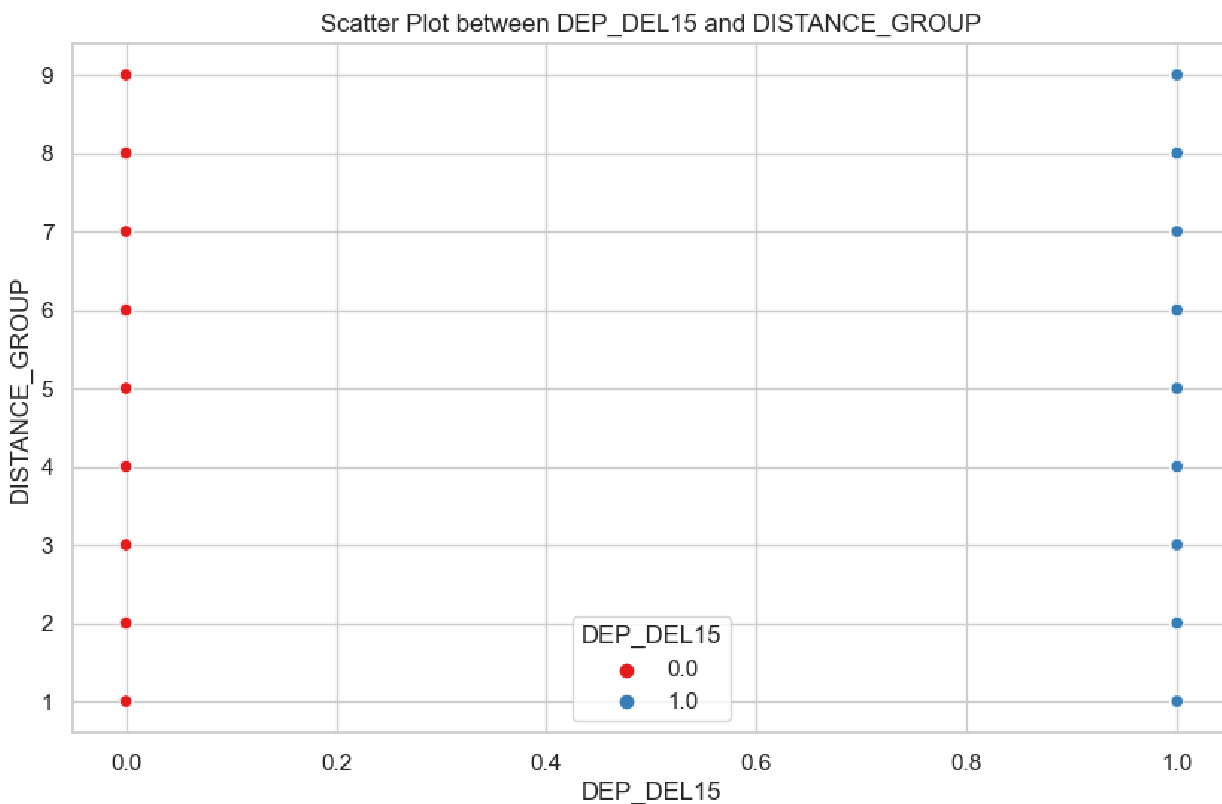

Distribution of SNWD
1e6
Frequency
0
1
2
3
4
5
6
−0.4
−0.2
0.0
0.2
0.4
SNWD


Distribution of TMAX
Frequency
TMAX
0
100000
200000
300000
400000
40
50
60
70
80
90


Distribution of AWND
Frequency
AWND
0
100000
200000
300000
400000
500000
4
6
8
10
12
14

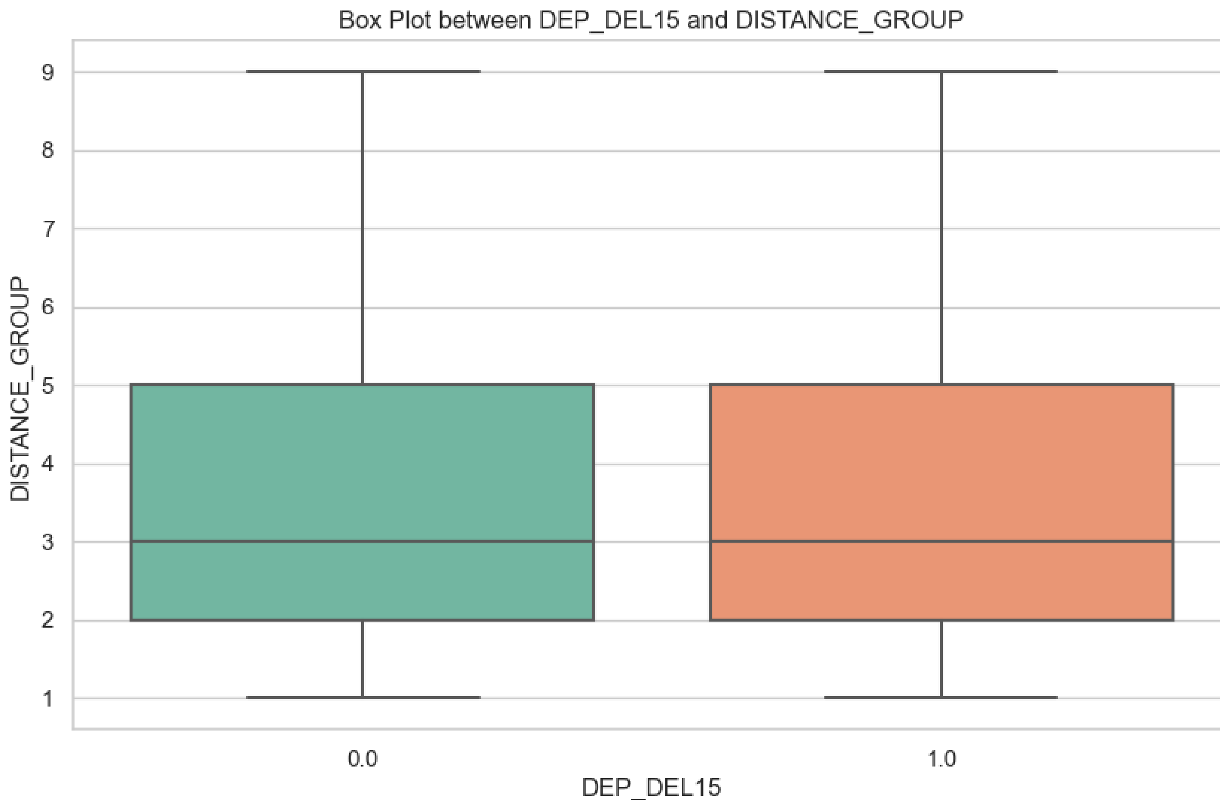
Bivariate Analysis using Scatter plot

```
def bivariate_analysis(fd):
    # Scatter plot between 'DEP_DEL15' and 'DISTANCE_GROUP'
    plt.figure(figsize=(10, 6))
    sns.scatterplot(data=fd, x='DEP_DEL15', y='DISTANCE_GROUP',
hue='DEP_DEL15', palette='Set1')
    plt.title('Scatter Plot between DEP_DEL15 and DISTANCE_GROUP')
    plt.xlabel('DEP_DEL15')
    plt.ylabel('DISTANCE_GROUP')
    plt.show()

bivariate_analysis(fd)
```




```
def bivariate_analysis(fd):
    # Box plot between 'DEP_DEL15' and 'DISTANCE_GROUP'
    plt.figure(figsize=(10, 6))
    sns.boxplot(data=fd, x='DEP_DEL15', y='DISTANCE_GROUP',
palette='Set2')
    plt.title('Box Plot between DEP_DEL15 and DISTANCE_GROUP')
    plt.xlabel('DEP_DEL15')
    plt.ylabel('DISTANCE_GROUP')
    plt.show()
```

```
bivariate_analysis(fd)
```



Exploratory Data Analysis using Python Ydata Profiling

```
pip install ydata_profiling

from ydata_profiling import ProfileReport

# Generate the report
profile = ProfileReport(fd,title="Airline delay w/ weather")

# Save the report to .html
profile.to_file("FlightDelay_report.html")
```


{"model_id":"87fb526b64454b9cac8470c826f31e7e","version_major":2,"version_minor":0}

{"model_id":"83e0ce20c788430c91a37da830471be2","version_major":2,"version_minor":0}

{"model_id":"2a1135590e084140ad766e71a1449695","version_major":2,"version_minor":0}

{"model_id":"f43b87a69ac940d49c804bd2fa2bd032","version_major":2,"version_minor":0}

```
profile
```

```
<IPython.core.display.HTML object>
```