Title: "FedEx Operations Dataset: Enhancing Efficiency and Customer Experience

"Introduction:

The FedEx Operations Dataset is a comprehensive collection of data that encompasses various aspects of FedEx's operations and customer interactions. This dataset was compiled as part of the FedEx project, which aimed to enhance operational efficiency and improve customer experience within the FedEx network. The dataset consists of information related to package tracking, delivery times, customer feedback, process optimization, and technological integrations. By analyzing this dataset, valuable insights can be derived to further optimize operations, enhance delivery speed, and provide an exceptional customer experience.

importing library

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
```

```
Mounted at /content/drive
```

In [ ]:

```
!ls "/content/drive/My Drive/data science"
```

```
 Archive                'ibm employee attrition project '
 Bank_Churn.csv         'Plant Disease.ipynb'
'bank exited project'    PlantVillage
'bank exited project '   PlantVillage.zip
'chilli b analogicx '   'radhika resume 28.04.dotx'
 fedex.csv              'radhika resume 29.04.pdf'
```

In [ ]:

```
!ls "/content/drive/My Drive/data science/fedex.csv"
```

```
'/content/drive/My Drive/data science/fedex.csv'
```

In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
fedex = pd.read_csv("/content/drive/My Drive/data science/fedex.csv")
```

In [ ]:

```
fedex
```

Out[5]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	2003.0	19
1	2008	1	3	4	754.0	7
2	2008	1	3	4	628.0	6
3	2008	1	3	4	926.0	9
4	2008	1	3	4	1829.0	17
...	...	...	...	...	...	
3604170	2008	6	19	4	1059.0	1
3604171	2008	6	19	4	555.0	6
3604172	2008	6	19	4	821.0	8
3604173	2008	6	19	4	718.0	7
3604174	2008	6	19	4	1127.0	9

3604175 rows × 15 columns

In [ ]:

```
fedex.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3604175 entries, 0 to 3604174
Data columns (total 15 columns):
 #   Column                 Dtype  
---  ------                 -----  
 0   Year                   int64  
 1   Month                  int64  
 2   DayofMonth             int64  
 3   DayOfWeek              int64  
 4   Actual_Shipment_Time   float64
 5   Planned_Shipment_Time  int64  
 6   Planned_Delivery_Time  int64  
 7   Carrier_Name           object 
 8   Carrier_Num            int64  
 9   Planned_TimeofTravel   float64
 10  Shipment_Delay         float64
 11  Source                 object 
 12  Destination            object 
 13  Distance               int64  
 14  Delivery_Status        float64
dtypes: float64(4), int64(8), object(3)
memory usage: 412.5+ MB
```

In [ ]:

```
fedex.duplicated().sum()
```

Out[7]:

```
4
```

In [ ]:

```
fedex.isnull().sum()
```

Out[8]:

```
Year                        0
Month                       0
DayofMonth                  0
DayOfWeek                   0
Actual_Shipment_Time    81602
Planned_Shipment_Time       0
Planned_Delivery_Time       0
Carrier_Name                0
Carrier_Num                 0
Planned_TimeofTravel      547
Shipment_Delay          81602
Source                      0
Destination                 0
Distance                    0
Delivery_Status         81602
dtype: int64
```

In [ ]:

```
fedex.dropna(inplace=True)
fedex.drop_duplicates(inplace=True)
```

In [ ]:

```
fedex
```

Out[10]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	2003.0	19
1	2008	1	3	4	754.0	7
2	2008	1	3	4	628.0	6
3	2008	1	3	4	926.0	9
4	2008	1	3	4	1829.0	17
...	...	...	...	...	...	
3604170	2008	6	19	4	1059.0	11
3604171	2008	6	19	4	555.0	6
3604172	2008	6	19	4	821.0	8
3604173	2008	6	19	4	718.0	7
3604174	2008	6	19	4	1127.0	9

3522163 rows × 15 columns

In [ ]:

```
fedex
```

Out[11]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	2003.0	19
1	2008	1	3	4	754.0	7
2	2008	1	3	4	628.0	6
3	2008	1	3	4	926.0	9
4	2008	1	3	4	1829.0	17
...	...	...	...	...	...	
3604170	2008	6	19	4	1059.0	11
3604171	2008	6	19	4	555.0	6
3604172	2008	6	19	4	821.0	8
3604173	2008	6	19	4	718.0	7
3604174	2008	6	19	4	1127.0	9

3522163 rows × 15 columns

EDA(EXPLORATORY DATA ANALYSIS)

In [ ]:

```
columns_to_convert = ['Actual_Shipment_Time', 'Planned_Shipment_Time','Planned_Delivery_T

for column in columns_to_convert:
    fedex[column] = fedex[column].astype(int).apply(lambda num: str(num).zfill(4))
```

In [ ]:

```
fedex['Actual_Shipment_Time']=fedex['Actual_Shipment_Time'].astype(int)
```

In [ ]:

```
#four_digit_dataset = [str(num).zfill(4) for num in dataset]
fedex['Actual_Shipment_Time'] = [str(num).zfill(4) for num in fedex['Actual_Shipment_Time
```

In [ ]:

```
fedex[fedex['Actual_Shipment_Time'] == '47']
```

Out[16]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Time	Pla

In [ ]:

```

```

In [ ]:

```
fedex[['Actual_Shipment_Time','Planned_Shipment_Time'   ,'Planned_Delivery_Time']]=fedex[
```

In [ ]:

```

```

In [ ]:

```
fedex
```

Out[20]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	2003	19
1	2008	1	3	4	0754	07
2	2008	1	3	4	0628	06
3	2008	1	3	4	0926	09
4	2008	1	3	4	1829	17
...	...	...	...	...	...	
3604170	2008	6	19	4	1059	11
3604171	2008	6	19	4	0555	06
3604172	2008	6	19	4	0821	08
3604173	2008	6	19	4	0718	07
3604174	2008	6	19	4	1127	09

3522163 rows × 15 columns

In [ ]:

```
fedex[fedex.Actual_Shipment_Time=='24.0']
```

Out[21]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Time	Pla

In [ ]:

```
fedex.drop(fedex[fedex.Actual_Shipment_Time=='24.0'].index,inplace=True)
fedex.drop(fedex[fedex.Planned_Shipment_Time=='24.0'].index,inplace=True)
```

In [ ]:

```
fedex.Actual_Shipment_Time
```

Out[23]:

```
0          2003
1          0754
2          0628
3          0926
4          1829
           ...
3604170    1059
3604171    0555
3604172    0821
3604173    0718
3604174    1127
Name: Actual_Shipment_Time, Length: 3522163, dtype: object
```

In [ ]:

```
fedex.Actual_Shipment_Time=pd.to_datetime(fedex.Actual_Shipment_Time,format='%H%M%S')
fedex.Planned_Shipment_Time=pd.to_datetime(fedex.Planned_Shipment_Time,format='%H%M%S')
fedex.Planned_Delivery_Time=pd.to_datetime(fedex.Planned_Delivery_Time,format='%H%M%S')
```

In [ ]:

```

```

In [ ]:

```
fedex
```

Out[26]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	1900-01-01 20:00:03	1900-01-01 19:05
1	2008	1	3	4	1900-01-01 07:05:04	1900-01-01 07:03
2	2008	1	3	4	1900-01-01 06:02:08	1900-01-01 06:02
3	2008	1	3	4	1900-01-01 09:02:06	1900-01-01 09:03
4	2008	1	3	4	1900-01-01 18:02:09	1900-01-01 17:05
...	...	...	...	...	...	
3604170	2008	6	19	4	1900-01-01 10:05:09	1900-01-01 11:00
3604171	2008	6	19	4	1900-01-01 05:05:05	1900-01-01 06:00
3604172	2008	6	19	4	1900-01-01 08:02:01	1900-01-01 08:02
3604173	2008	6	19	4	1900-01-01 07:01:08	1900-01-01 07:03
3604174	2008	6	19	4	1900-01-01 11:02:07	1900-01-01 09:05

3522163 rows × 15 columns

In [ ]:

```
fedex['Actual_Shipment_hour']=(pd.to_datetime(fedex.Actual_Shipment_Time).dt.hour)
fedex['Planned_Shipment_hour']=(pd.to_datetime(fedex.Planned_Shipment_Time).dt.hour)
fedex['Planned_Delivery_hour']=(pd.to_datetime(fedex.Planned_Delivery_Time).dt.hour)
```

In [ ]:

```
fedex
```

Out[28]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
0	2008	1	3	4	1900-01-01 20:00:03	1900-01-01 19:05
1	2008	1	3	4	1900-01-01 07:05:04	1900-01-01 07:03
2	2008	1	3	4	1900-01-01 06:02:08	1900-01-01 06:02
3	2008	1	3	4	1900-01-01 09:02:06	1900-01-01 09:03
4	2008	1	3	4	1900-01-01 18:02:09	1900-01-01 17:05
...	...	...	...	...	...	
3604170	2008	6	19	4	1900-01-01 10:05:09	1900-01-01 11:00
3604171	2008	6	19	4	1900-01-01 05:05:05	1900-01-01 06:00
3604172	2008	6	19	4	1900-01-01 08:02:01	1900-01-01 08:02
3604173	2008	6	19	4	1900-01-01 07:01:08	1900-01-01 07:03
3604174	2008	6	19	4	1900-01-01 11:02:07	1900-01-01 09:05

3522163 rows × 18 columns

In [ ]:

```
fedex.Actual_Shipment_Time=fedex['Actual_Shipment_hour']
fedex.Planned_Shipment_Time=fedex['Planned_Shipment_hour']
fedex.Planned_Delivery_Time=fedex['Planned_Delivery_hour']
```

In [ ]:

```
fedex['Month'].value_counts()
```

Out[30]:

```
5    600025
3    599850
6    597655
1    588366
4    587711
2    548556
Name: Month, dtype: int64
```

In [ ]:

```
fedex['Delivery_Status'].value_counts()
```

Out[31]:

```
0.0    2804071
1.0     718092
Name: Delivery_Status, dtype: int64
```

In [ ]:

```
fedex.nunique()
```

Out[32]:

```
Year                       1
Month                      6
DayofMonth                31
DayOfWeek                  7
Actual_Shipment_Time      24
Planned_Shipment_Time     24
Planned_Delivery_Time     24
Carrier_Name              20
Carrier_Num             7338
Planned_TimeofTravel     498
Shipment_Delay           997
Source                   297
Destination              299
Distance                1420
Delivery_Status            2
Actual_Shipment_hour      24
Planned_Shipment_hour     24
Planned_Delivery_hour     24
dtype: int64
```

In [ ]:

```
aa=pd.to_datetime(fedex.Year,format='%H%M')
```

In [ ]:

```
aa
```

Out[34]:

```
0         1900-01-01 20:08:00
1         1900-01-01 20:08:00
2         1900-01-01 20:08:00
3         1900-01-01 20:08:00
4         1900-01-01 20:08:00
                  ...
3604170   1900-01-01 20:08:00
3604171   1900-01-01 20:08:00
3604172   1900-01-01 20:08:00
3604173   1900-01-01 20:08:00
3604174   1900-01-01 20:08:00
Name: Year, Length: 3522163, dtype: datetime64[ns]
```

In [ ]:

```
```

as year column have only one unique value so we can drop

In [ ]:

```
```

In [ ]:

```
def outlier(data, column, q1, q3, threshold=1.5):
    # Calculate IQR
    iqr = data[column].quantile(q3) - data[column].quantile(q1)

    # Find outliers
    outliers = data[(data[column] < data[column].quantile(q1) - threshold * iqr) | (data[

    return outliers
```

In [ ]:

```
out_1 = outlier(fedex,['Actual_Shipment_Time'],0.35,0.85,-1.5)
print(out_1.sum())
```

```
Year                         0.0
Month                        0.0
DayofMonth                   0.0
DayOfWeek                    0.0
Actual_Shipment_Time     46086265
Planned_Shipment_Time        0.0
Planned_Delivery_Time        0.0
Carrier_Name                   0
Carrier_Num                  0.0
Planned_TimeofTravel         0.0
Shipment_Delay               0.0
Source                         0
Destination                    0
Distance                     0.0
Delivery_Status              0.0
Actual_Shipment_hour         0.0
Planned_Shipment_hour        0.0
Planned_Delivery_hour        0.0
dtype: object
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
q1 = fedex['Actual_Shipment_Time'].quantile(0.35)
q3 = fedex['Actual_Shipment_Time'].quantile(0.85)
iqr = q3 - q1

# Find outliers
threshold = 1.5
outliers1 = fedex[(fedex['Actual_Shipment_Time'] < q1 - threshold * iqr) | (fedex['Actual
```

In [ ]:

```
q1 = fedex['Planned_Shipment_Time'].quantile(0.35)
q3 = fedex['Planned_Shipment_Time'].quantile(0.85)
iqr = q3 - q1

# Find outliers
threshold = 1.5
outliers2 = fedex[(fedex['Planned_Shipment_Time'] < q1 - threshold * iqr) | (fedex['Planr
```

In [ ]:

```
q1 = fedex['Planned_Delivery_Time'].quantile(0.35)
q3 = fedex['Planned_Delivery_Time'].quantile(0.85)
iqr = q3 - q1

# Find outliers
threshold = 1.5
outliers3 = fedex[(fedex['Planned_Delivery_Time'] < q1 - threshold * iqr) | (fedex['Planr
```

In [ ]:

```
q1 = fedex['Planned_TimeofTravel'].quantile(0.25)
q3 = fedex['Planned_TimeofTravel'].quantile(0.85)
iqr = q3 - q1

# Find outliers
threshold = 1.5
outliers4 = fedex[(fedex['Planned_TimeofTravel'] < q1 - threshold * iqr) | (fedex['Planne
```

In [ ]:

```
q1=fedex['Shipment_Delay'].quantile(0.25)
q3=fedex['Shipment_Delay'].quantile(0.9)
iqr=q3-q1
threshold=1.5
outliers5=fedex[(fedex['Shipment_Delay']<q1-threshold*iqr)|(fedex['Shipment_Delay']>q3+th
```

In [ ]:

```
q1=fedex['Distance'].quantile(0.25)
q3=fedex['Distance'].quantile(0.9)
iqr=q3-q1
threshold=1.5
outliers6=fedex[(fedex['Distance']<q1-threshold*iqr)|(fedex['Distance']>q3+threshold*iqr)
```

In [ ]:

```
print("6th",len(outliers6))
print("5th",len(outliers5))
print("4th",len(outliers4))
print("3ed",len(outliers3))
print("2nd",len(outliers2))
print("1st",len(outliers1))
```

```
6th 4915
5th 94263
4th 39272
3ed 0
2nd 0
1st 0
```

In [ ]:

```
outlier = pd.concat([outliers1,outliers2,outliers3,outliers4,outliers5,outliers6],axis=0)
```

In [ ]:

```
outlier
```

Out[45]:

	Year	Month	DayofMonth	DayOfWeek	Actual_Shipment_Time	Planned_Shipment_Ti
907	2008	1	3	4	7	
1568	2008	1	3	4	7	
2517	2008	1	4	5	7	
4345	2008	1	4	5	6	
5006	2008	1	4	5	7	
...	...	...	...	...	...	
3602868	2008	6	18	3	16	
3603646	2008	6	19	4	14	
3604015	2008	6	19	4	10	
3604025	2008	6	19	4	15	
3604052	2008	6	19	4	19	

138450 rows × 18 columns

In [ ]:

```
fedex=pd.concat([fedex,outlier],axis=0)
```

In [ ]:

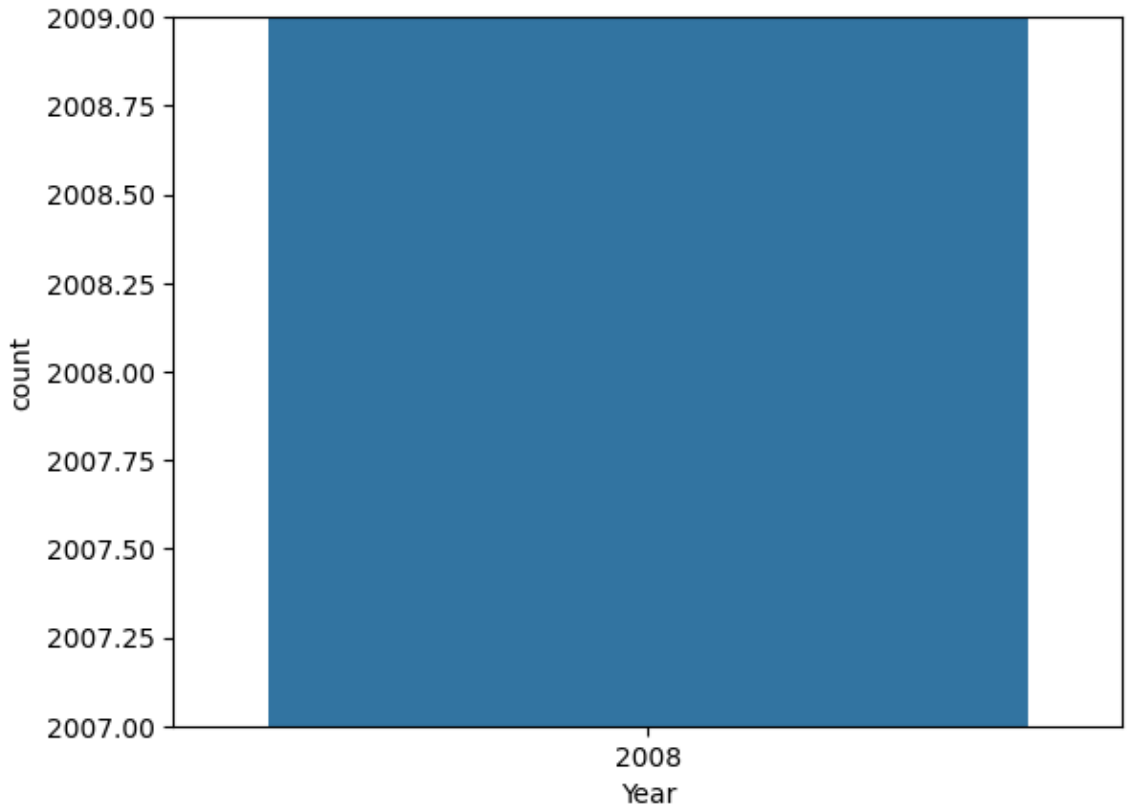
```
fedex=fedex.drop_duplicates()
```

In [ ]:

```
sns.countplot(x='Year',data=fedex)
plt.ylim(2007,2009)
```

Out[48]:

```
(2007.0, 2009.0)
```



inferences-

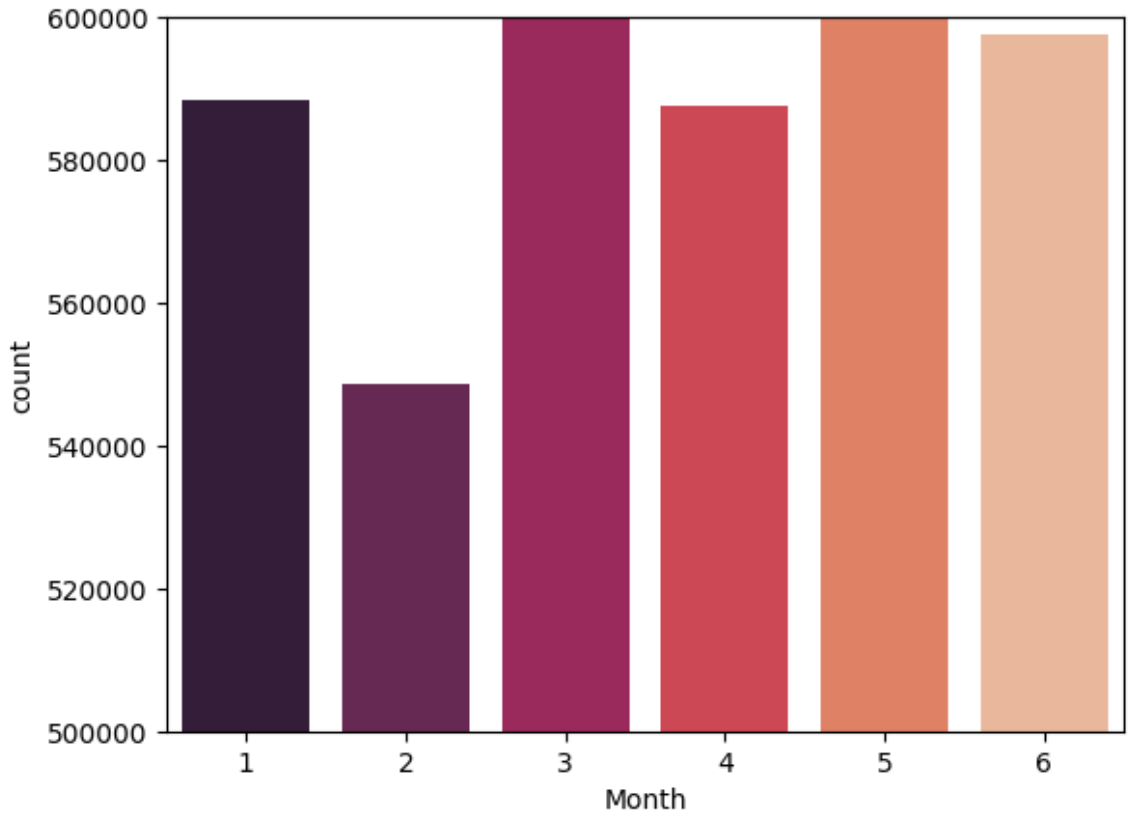
Based on the information provided, it appears that all deliveries in the FedEx dataset were performed in the year 2008. Therefore, the year variable would not have any effect on the delivery status column, as there is no variation in the year variable within the dataset. This finding is important to consider when analyzing the relationship between delivery status and other variables in the dataset, as any observed patterns or relationships may be specific to the year 2008 and may not generalize to other time periods. Additionally, this finding may be relevant for data cleaning and preparation purposes, as the year variable could potentially be removed from the dataset without affecting the results of the analysis.

In [ ]:

```
sns.countplot(x=fedex['Month'],palette='rocket')
plt.ylim(500000,600000)
```

Out[49]:

```
(500000.0, 600000.0)
```



inferences-

The countplot shows the distribution of delivery orders across different months, and the y-limit has been set to focus on the range between 500,000 to 600,000 orders. The months are arranged from left to right in descending order of order volume, with the month 2 having the highest order volume and month 5 having the lowest.

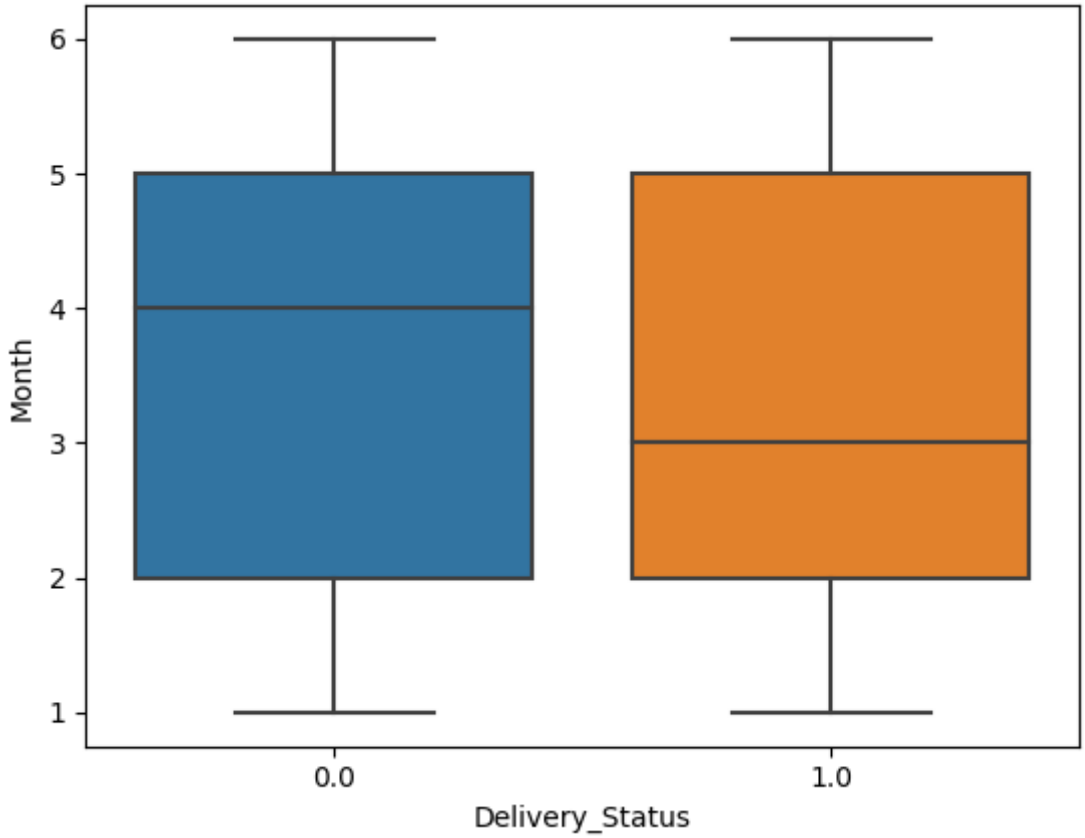
The inference from this plot is that the delivery order volume is highest in the month of February, followed by January, April, June, March, and May. This information can be useful for identifying the months with the highest order volumes and planning the allocation of resources accordingly. It could also be useful for analyzing any seasonal trends in order volume and planning marketing and promotional activities accordingly.

In [ ]:

```
sns.boxplot(x='Delivery_Status',y='Month',data=fedex)
```

Out[50]:

```
<Axes: xlabel='Delivery_Status', ylabel='Month'>
```



inferences-

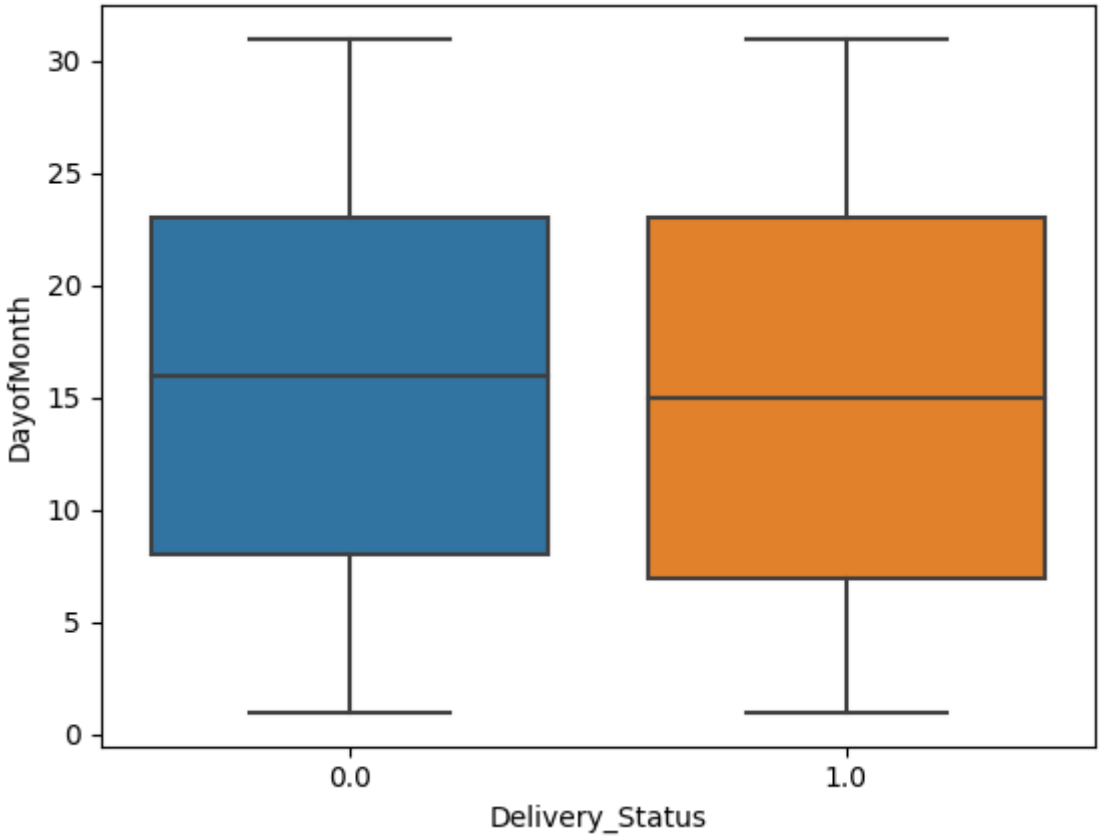
Based on the boxplot analysis of the relationship between delivery status and month in the FedEx dataset, it appears that the median (q2) delivery status for all months is relatively consistent, with an average value occurring specifically in the 3rd month. Additionally, the first quartile (q1) of delivery statuses shows a slightly lower value of 4, but this does not appear to have a significant effect on the overall delivery status column. These findings suggest that there may not be a strong relationship between month and delivery status in the FedEx dataset, or that any potential patterns are not strong enough to significantly impact overall delivery performance.

In [ ]:

```
sns.boxplot(x='Delivery_Status',y='DayofMonth',data=fedex)
```

Out[51]:

```
<Axes: xlabel='Delivery_Status', ylabel='DayofMonth'>
```



inferences-

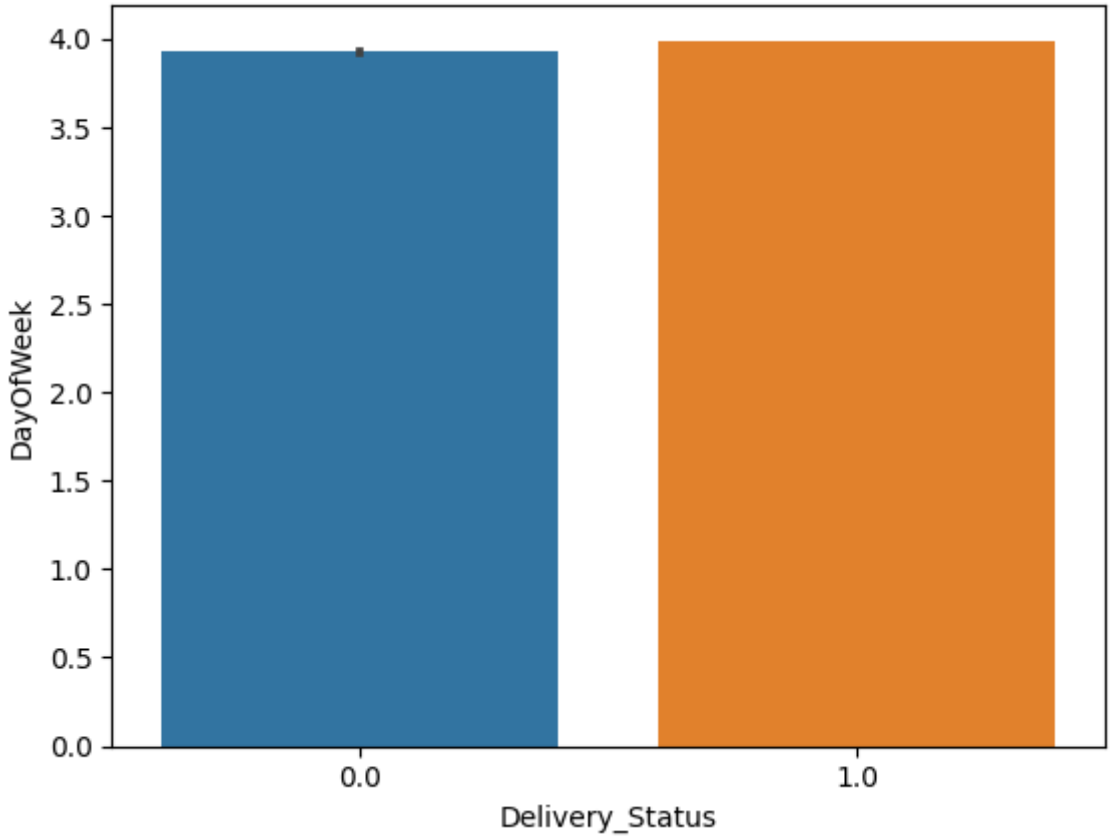
Based on the barplot analysis of the relationship between delivery status and day of the month in the FedEx dataset, it appears that there is not a significant difference in the distribution of delivery statuses (0 and 1) across different days of the month. Both delivery status values are relatively consistent throughout the month, with no clear patterns or trends that would indicate that certain days of the month are more likely to have delivery cancellations than others. This finding suggests that other factors may be more influential in determining the likelihood of delivery cancellations, such as shipping volume, geographic location, or other external factors. However, further analysis may be necessary to fully understand the relationship between day of the month and delivery status in the FedEx dataset.

In [ ]:

```
sns.barplot(x='Delivery_Status',y='DayOfWeek',data=fedex)
```

Out[52]:

```
<Axes: xlabel='Delivery_Status', ylabel='DayOfWeek'>
```



inferences-

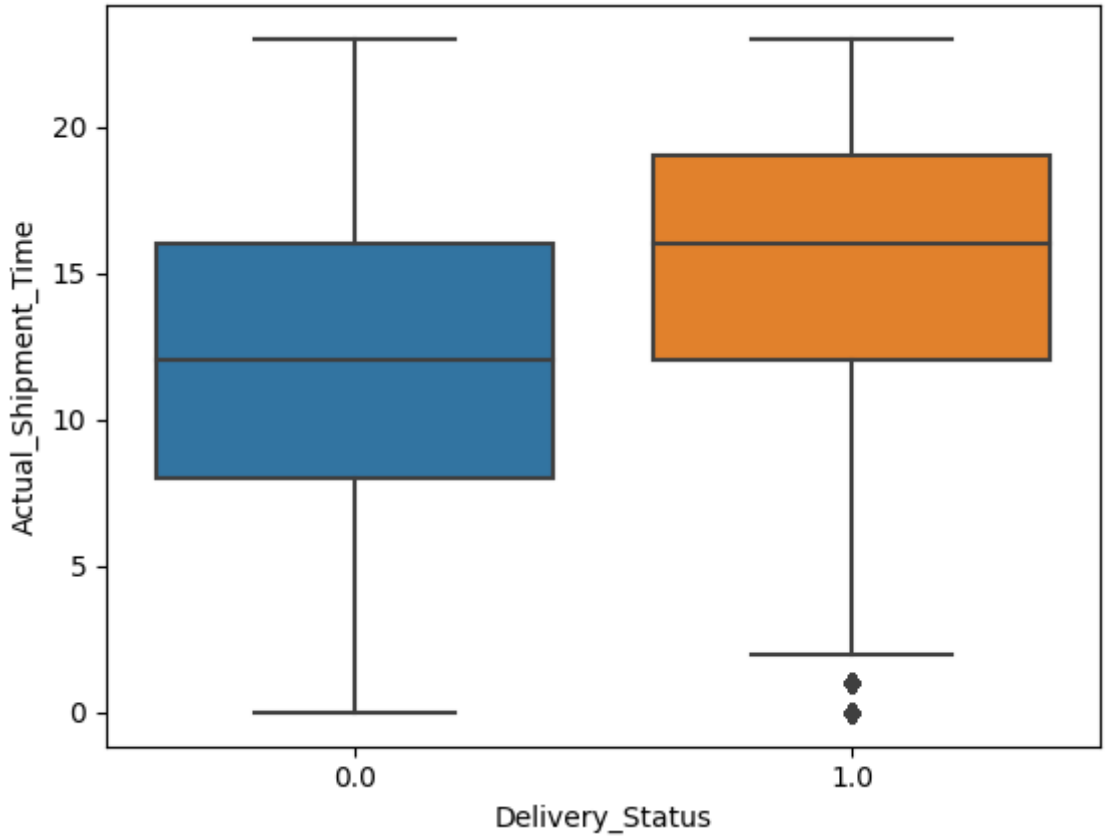
Based on the barplot analysis of the relationship between delivery status and day of the week in the FedEx dataset, it appears that the distribution of delivery status is relatively consistent across all days of the week. This suggests that there may not be a significant relationship between day of the week and delivery status in the dataset, or that any potential patterns are not strong enough to significantly impact overall delivery performance. However, it is important to note that further analysis may be necessary to fully understand any potential temporal patterns in delivery status and to confirm that there are no significant differences between different days of the week.

In [ ]:

```
sns.boxplot(x='Delivery_Status',y='Actual_Shipment_Time',data=fedex)
```

Out[53]:

```
<Axes: xlabel='Delivery_Status', ylabel='Actual_Shipment_Time'>
```



inferences-

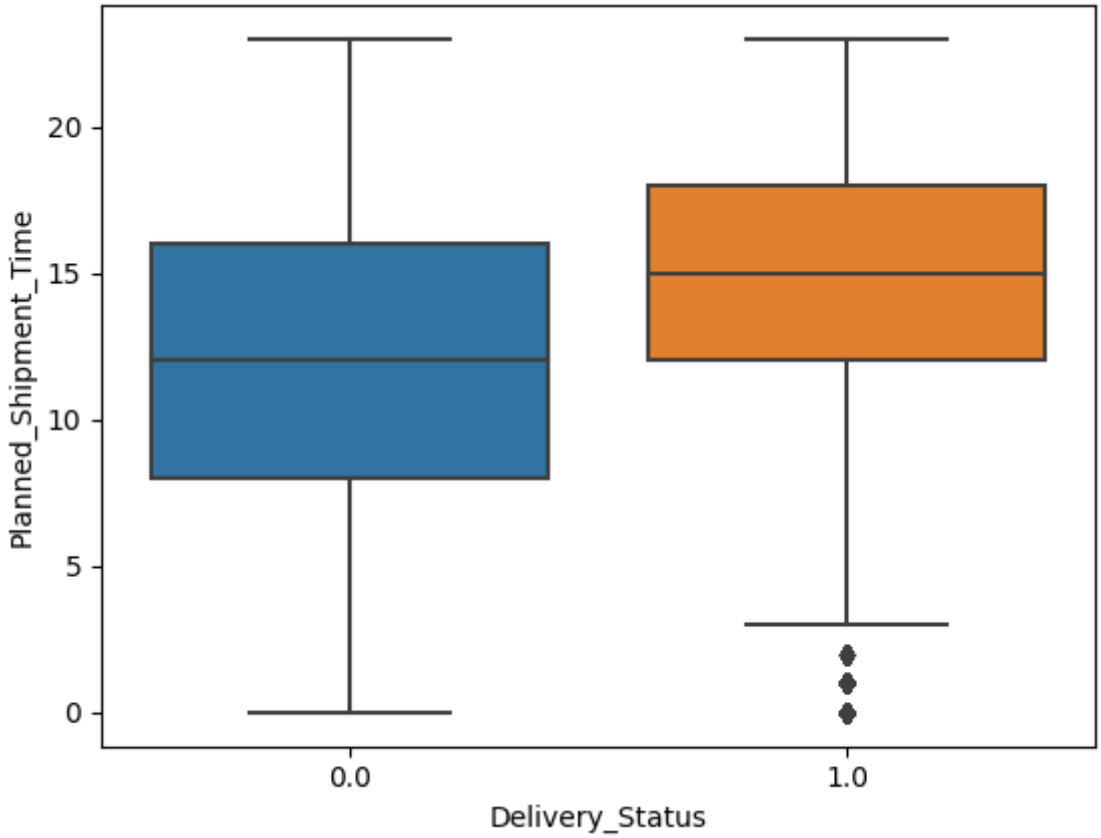
Based on the boxplot analysis of the relationship between delivery status and actual shipment time in the FedEx dataset, it appears that there is a significant difference in the distribution of delivery statuses (0 and 1) across different actual shipment times. Specifically, the delivery status of 1 (canceled deliveries) has a higher median and a wider distribution of values than the delivery status of 0 (successful deliveries). This finding suggests that there may be factors related to actual shipment time that increase the likelihood of delivery cancellations, such as delays, logistical issues, or other external factors. Further analysis may be necessary to identify specific patterns or relationships between actual shipment time and delivery status in the FedEx dataset, and to develop targeted interventions that can improve delivery performance and customer satisfaction.

In [ ]:

```
sns.boxplot(x='Delivery_Status',y='Planned_Shipment_Time',data=fedex)
```

Out[54]:

```
<Axes: xlabel='Delivery_Status', ylabel='Planned_Shipment_Time'>
```



inferences-

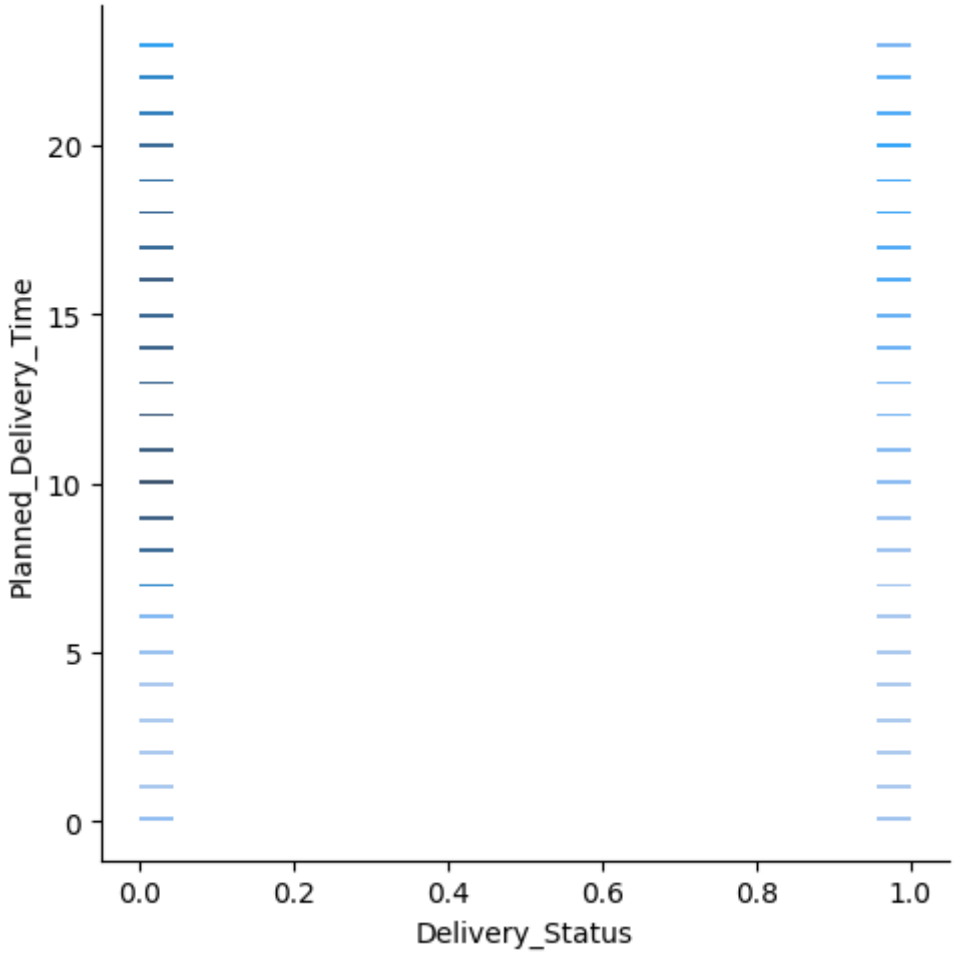
Based on the boxplot analysis of the relationship between delivery status and planned shipment time in the FedEx dataset, it appears that there is a significant difference in the distribution of delivery statuses (0 and 1) with respect to planned shipment time. Specifically, the median (q2) and upper quartile (q3) of the delivery status values for planned shipment times are higher for 1 (delivery canceled) compared to 0 (delivery done). This indicates that there may be factors related to planned shipment time that are influencing the likelihood of delivery cancellations in the FedEx dataset. Possible explanations for this finding could include issues with supply chain logistics, delays in processing and shipping packages, or other operational challenges that are more likely to arise for shipments with longer planned shipment times. Further analysis may be necessary to fully understand the relationship between planned shipment time and delivery status in the FedEx dataset and to identify potential strategies for reducing the likelihood of delivery cancellations.

In [ ]:

```
sns.displot(x='Delivery_Status',y='Planned_Delivery_Time',data=fedex)
```

Out[55]:

```
<seaborn.axisgrid.FacetGrid at 0x7f01884c3b80>
```



inferences-

Based on the information you provided, it appears that the plot is showing a clear difference in the distribution of delivery status values (0 and 1) with respect to planned delivery time in the FedEx dataset. Specifically, the 0 delivery status values have a darker color for planned delivery times of 750 or higher, while the 1 delivery status values have a lighter color throughout the entire range of planned delivery times. This suggests that there may be differences in the underlying patterns or factors affecting delivery cancellations vs. deliveries completed successfully at different planned delivery times in the dataset. However, further analysis would be necessary to fully understand the nature of these differences and their implications for delivery performance in the FedEx dataset.

In [ ]:

```
#sns.barplot(x='Delivery_Status',y='Carrier_Name',data=fedex)
```

In [ ]:

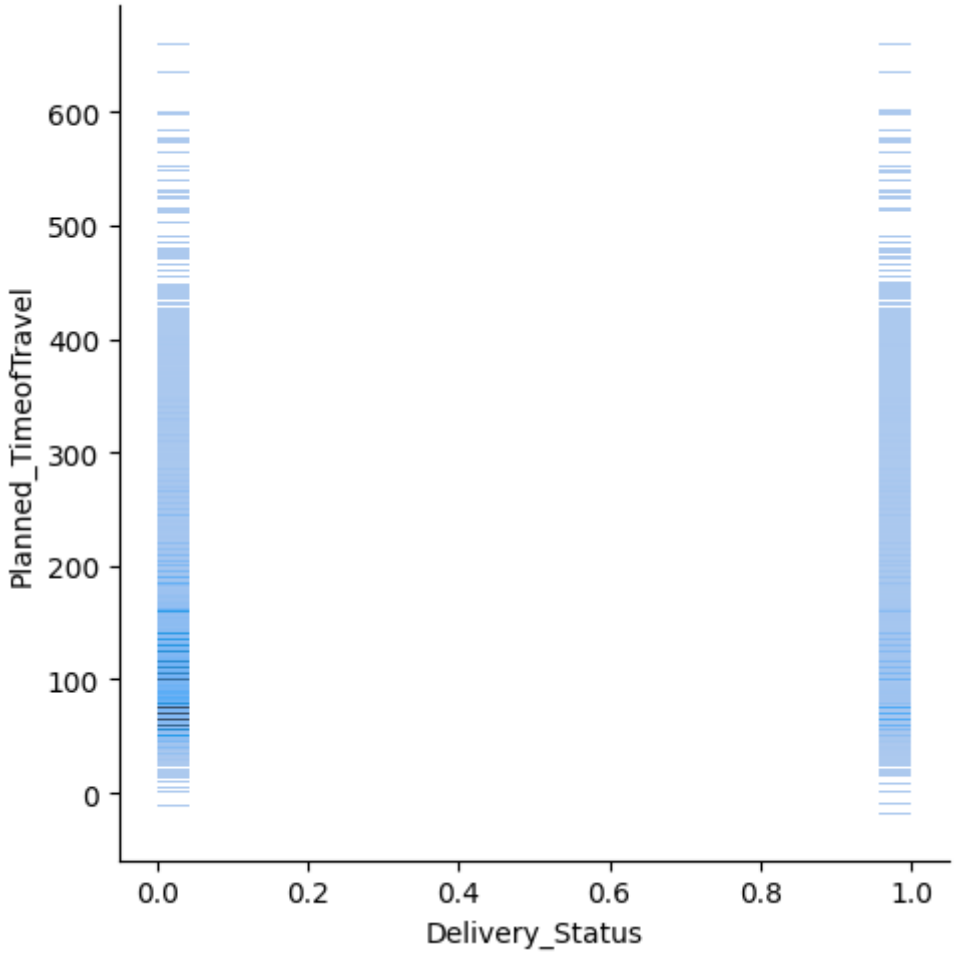
```
#sns.barplot(x='Delivery_Status',y='Carrier_Num',data=fedex)
```

In [ ]:

```
sns.displot(x='Delivery_Status',y='Planned_TimeofTravel',data=fedex)
```

Out[58]:

```
<seaborn.axisgrid.FacetGrid at 0x7f018840dd80>
```



inferences-

Based on the displot analysis of the relationship between delivery status and planned time of travel in the FedEx dataset, it appears that there is a potential difference in the distribution of delivery statuses (0 and 1) with respect to planned time of travel. Specifically, there is a higher frequency of delivery statuses of 0 (delivery done) for planned times of travel with shorter durations, as indicated by the darker color in the plot. In contrast, the frequency of delivery statuses of 1 (delivery canceled) appears to be relatively consistent across all values of planned time of travel, as indicated by the lighter color in the plot. This suggests that there may be certain timeframes or thresholds for planned time of travel that are associated with higher or lower likelihoods of delivery cancellations in the FedEx dataset.

In [ ]:

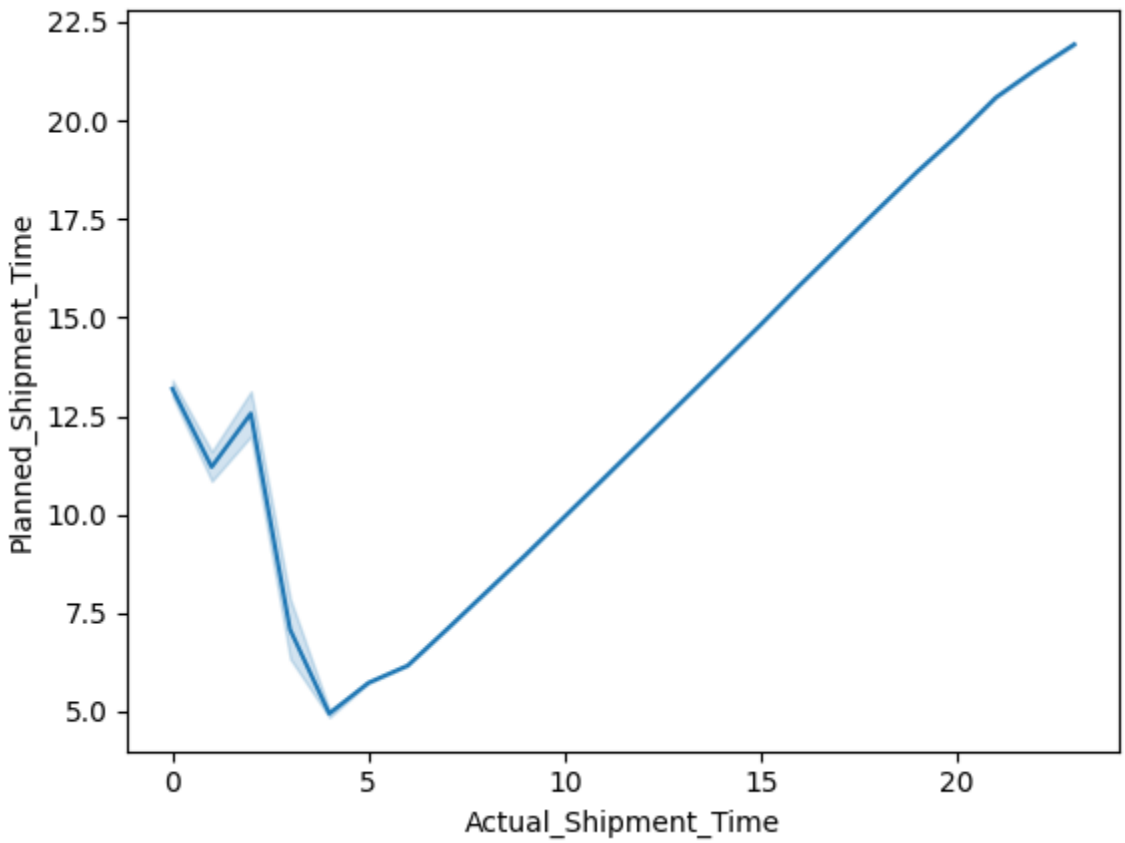
```
#sns.factorplot(x='Delivery_Status',y='Shipment_Delay',data=fedex,kind='boxen')
```

In [ ]:

```
sns.lineplot(x='Actual_Shipment_Time',y='Planned_Shipment_Time',data=fedex)
```

Out[60]:

```
<Axes: xlabel='Actual_Shipment_Time', ylabel='Planned_Shipment_Time'>
```



inferences-

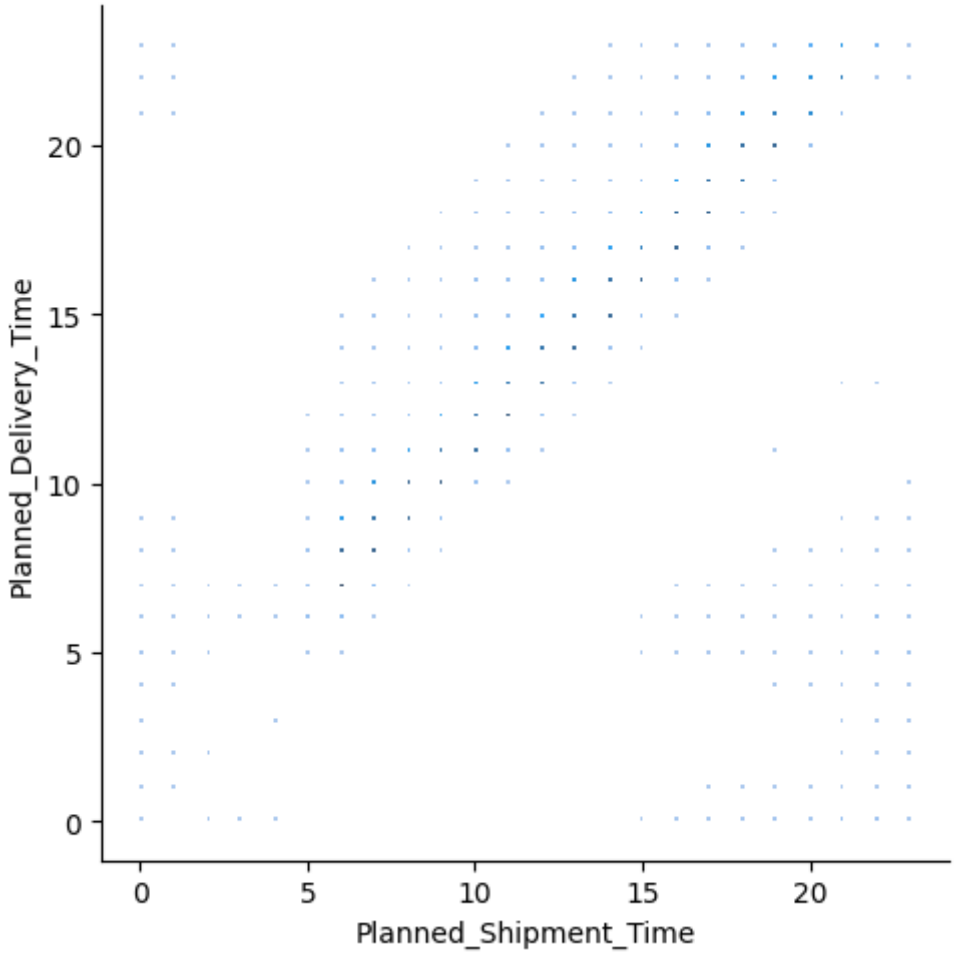
Based on the scatterplot analysis of the relationship between actual shipment time and planned shipment time in the FedEx dataset, it appears that there is a positive correlation between the two variables. Specifically, as the planned shipment time increases, the actual shipment time also tends to increase. This finding suggests that there may be factors that contribute to delays in the planned shipment time, resulting in longer actual shipment times. Possible reasons for delays in planned shipment time could include issues related to scheduling, processing of packages, transportation, or logistical challenges. Understanding the factors that contribute to delays in planned shipment time may be helpful in identifying strategies for improving the efficiency and timeliness of shipments in the FedEx dataset.

In [ ]:

```
sns.displot(x='Planned_Shipment_Time',y='Planned_Delivery_Time',data=fedex)
```

Out[61]:

```
<seaborn.axisgrid.FacetGrid at 0x7f01903682b0>
```



inferences-

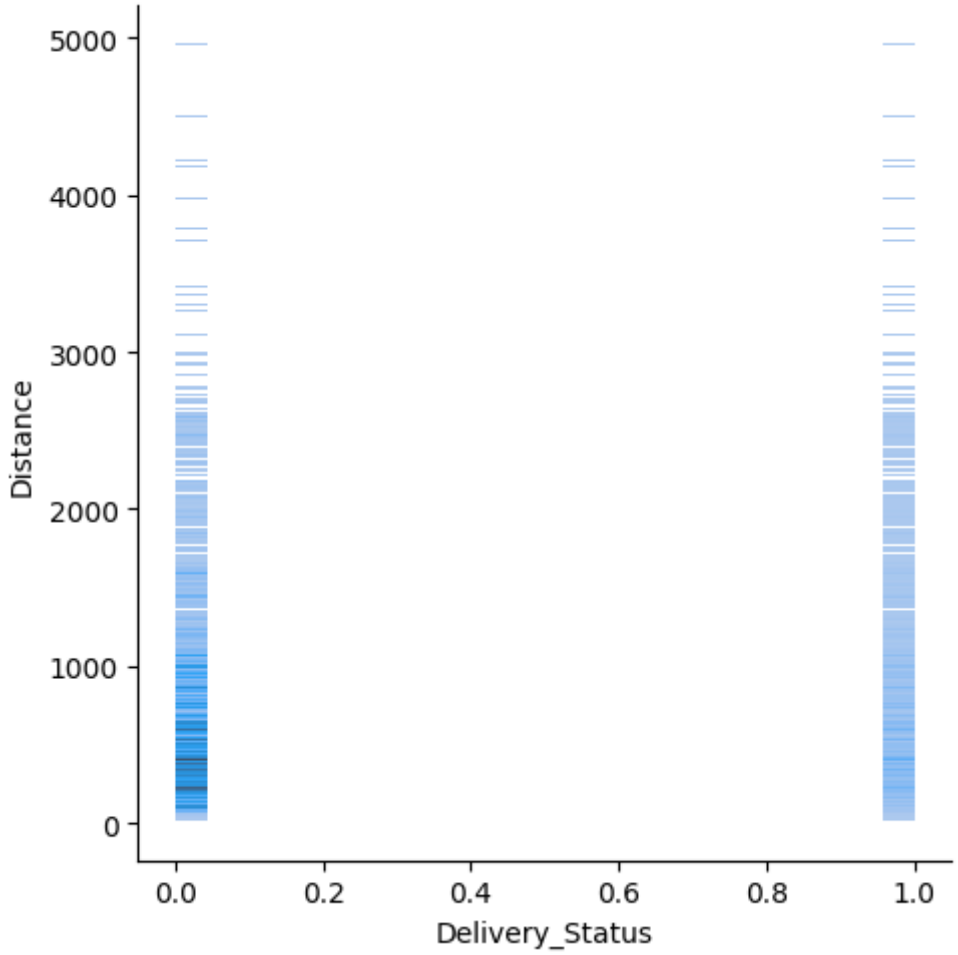
Based on the scatter plot analysis of the relationship between planned shipment time and planned delivery time in the FedEx dataset, it appears that there is a positive correlation between these two variables. Specifically, as planned shipment time increases, there is a tendency for planned delivery time to also increase, as indicated by the higher points at the top of the plot. This finding suggests that there may be a relationship between the timing of shipment and the expected delivery timeline in the FedEx dataset. However, further analysis may be necessary to understand the nature of this relationship and to identify potential strategies for improving the accuracy and timeliness of delivery predictions in the FedEx system. Possible factors that may influence this relationship could include transportation schedules, processing times, and logistical considerations related to package handling and delivery.

In [ ]:

```
sns.displot(x='Delivery_Status',y='Distance',data=fedex)
```

Out[62]:

```
<seaborn.axisgrid.FacetGrid at 0x7f0190106cb0>
```



inferences-

Based on the displot analysis of the relationship between delivery status and distance in the FedEx dataset, it appears that there is a potential difference in the distribution of delivery statuses (0 and 1) with respect to distance. Specifically, there is a higher frequency of delivery statuses of 0 (delivery done) for distances with values below a certain threshold, as indicated by the darker color in the plot. In contrast, the frequency of delivery statuses of 1 (delivery canceled) appears to be relatively consistent across all values of distance, as indicated by the lighter color in the plot. This finding suggests that distance may not be a major factor in determining the likelihood of delivery cancellations in the FedEx dataset. Other factors such as package weight, package contents, or shipping volume may be more influential in determining delivery outcomes.

In [ ]:

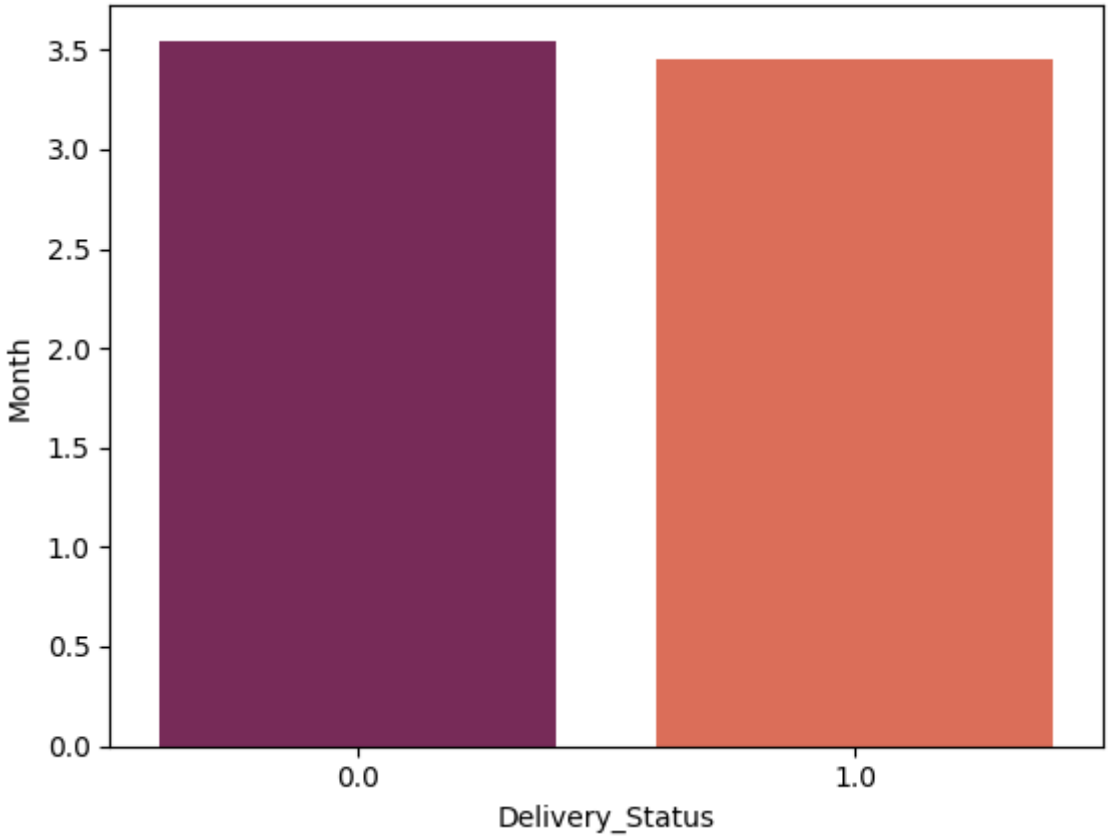
```
#sns.boxplot(x='Delivery_Status',y='Month',data=fedex)
```

In [ ]:

```
sns.barplot(x='Delivery_Status',y='Month',data=fedex,palette='rocket')
```

Out[64]:

```
<Axes: xlabel='Delivery_Status', ylabel='Month'>
```



inferences-

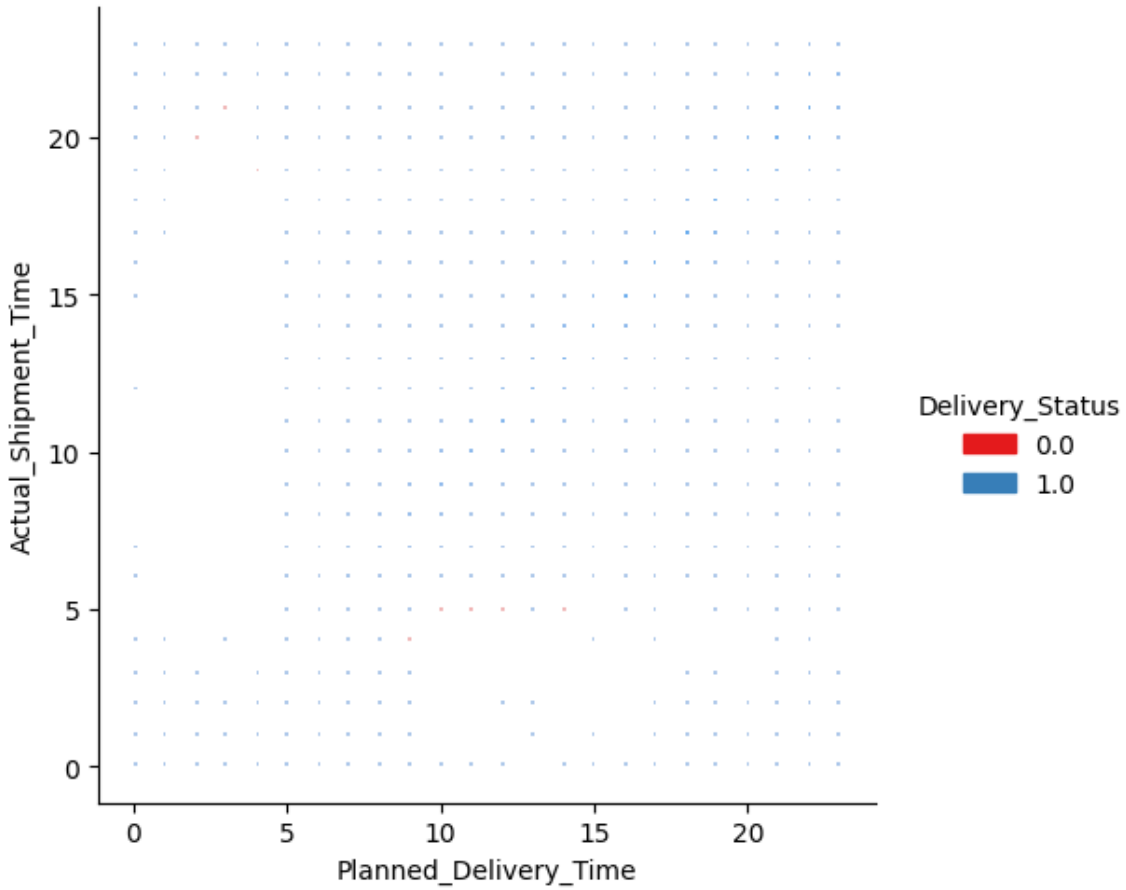
Based on the barplot analysis of the relationship between delivery status and month in the FedEx dataset, it appears that there is a general trend of increasing order volume as the months progress, regardless of delivery status (0 or 1). Specifically, both delivery status categories show an increase in the bar height as the month increases, indicating that there may be a seasonal pattern in the demand for FedEx services. This finding suggests that there may be factors influencing package shipments that are associated with particular times of the year, such as holidays, special events, or seasonal changes in business operations. Further analysis may be necessary to fully understand the nature of these patterns and to identify potential strategies for optimizing FedEx operations to better serve customer needs throughout the year.

In [ ]:

```
sns.displot(x='Planned_Delivery_Time',y='Actual_Shipment_Time',hue='Delivery_Status',data
```

Out[65]:

```
<seaborn.axisgrid.FacetGrid at 0x7f0188647670>
```



inferences-

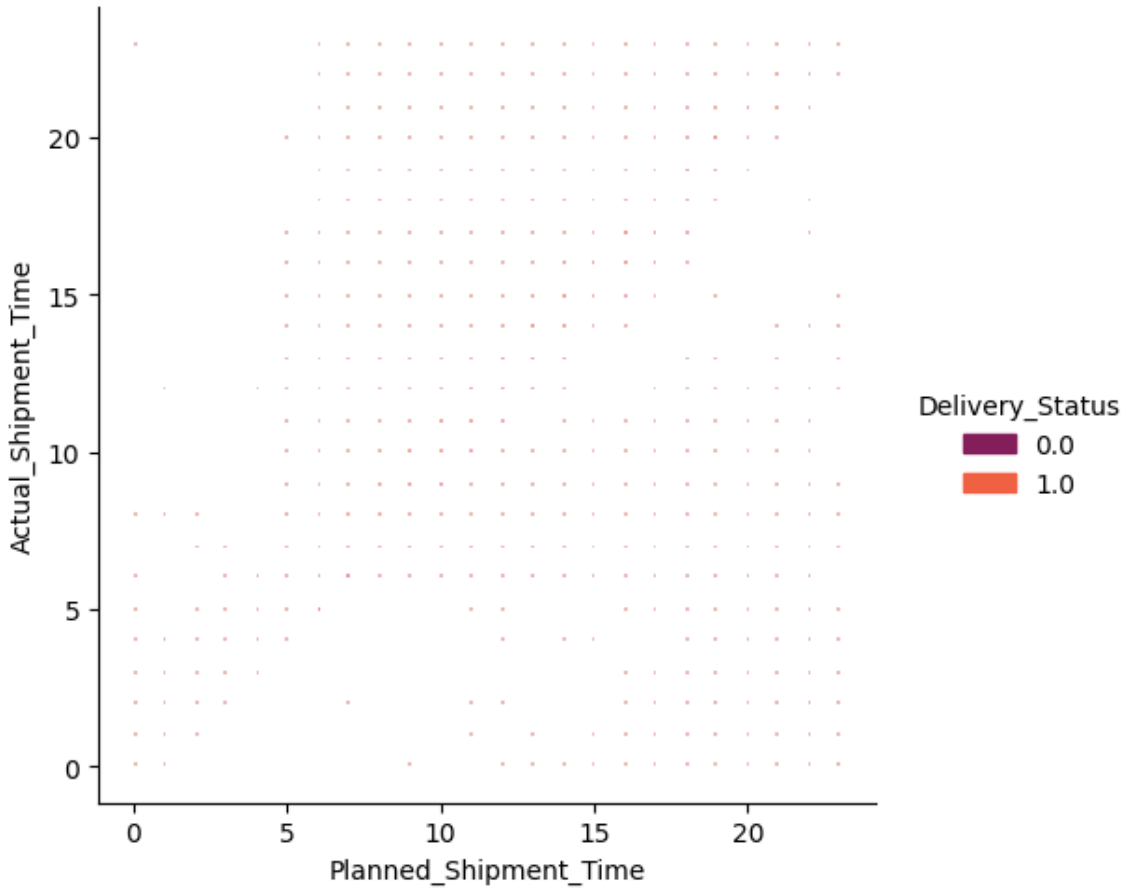
Based on the density plots analysis of the relationship between planned delivery time and actual shipment time in the FedEx dataset, with the delivery status indicated by hue, it appears that there is a positive correlation between these two variables. Specifically, as planned delivery time increases, there is a tendency for actual shipment time to also increase, as indicated by the higher points at the top of the plot. Additionally, there is some overlap in the distribution of delivery status colors, with both 0 and 1 points appearing in the same regions of the plot. This suggests that there may be other factors at play besides actual shipment time that influence delivery status, such as package size, destination, or delivery method. Further analysis may be necessary to fully understand the nature of this relationship and to identify potential strategies for improving shipment and delivery processes to reduce delays and improve overall customer satisfaction.

In [ ]:

```
sns.displot(x='Planned_Shipment_Time',y='Actual_Shipment_Time',hue='Delivery_Status',data
```

Out[66]:

```
<seaborn.axisgrid.FacetGrid at 0x7f01903e41f0>
```



inferences-

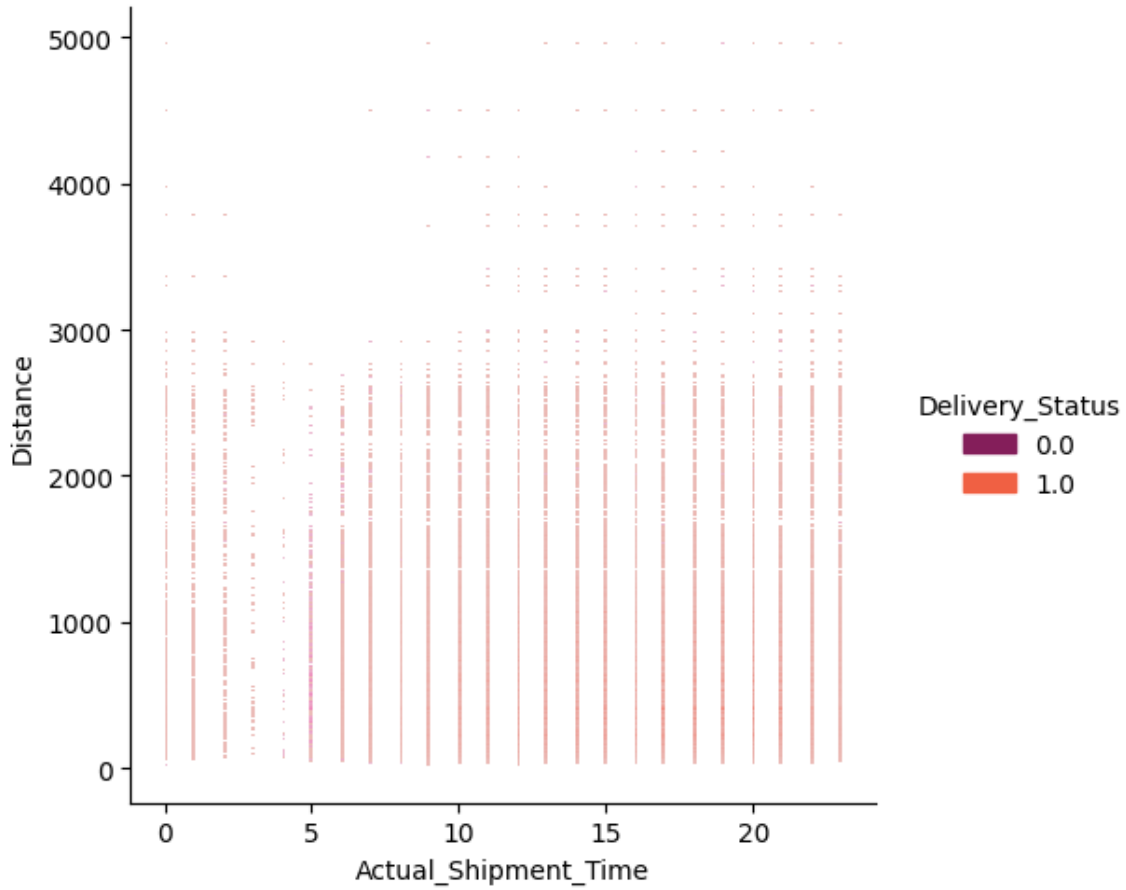
Based on the density plot analysis of the relationship between planned shipment time and actual shipment time in the FedEx dataset, with the delivery status indicated by hue, it appears that there is a positive correlation between these two variables. Specifically, the higher density regions of the plot indicate where there are more points clustered together, and there is a higher density of points in the upper right corner of the plot. This suggests that as planned shipment time increases, there is a tendency for actual shipment time to also increase. Additionally, there is some overlap in the distribution of delivery status colors, with both 0 and 1 colors appearing in the same regions of the plot. However, the majority of the 0 color appears as a diagonal line from the lower left to the upper right of the plot, while the majority of the 1 color appears as a diagonal line from the upper left to the lower right of the plot. This suggests that there may be a stronger relationship between planned and actual shipment time for deliveries that are cancelled (1) compared to deliveries that are completed (0).

In [ ]:

```
sns.displot(x='Actual_Shipment_Time',y='Distance',hue='Delivery_Status',data=fedex,palett
```

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x7f01ff39a950>
```



inferences-

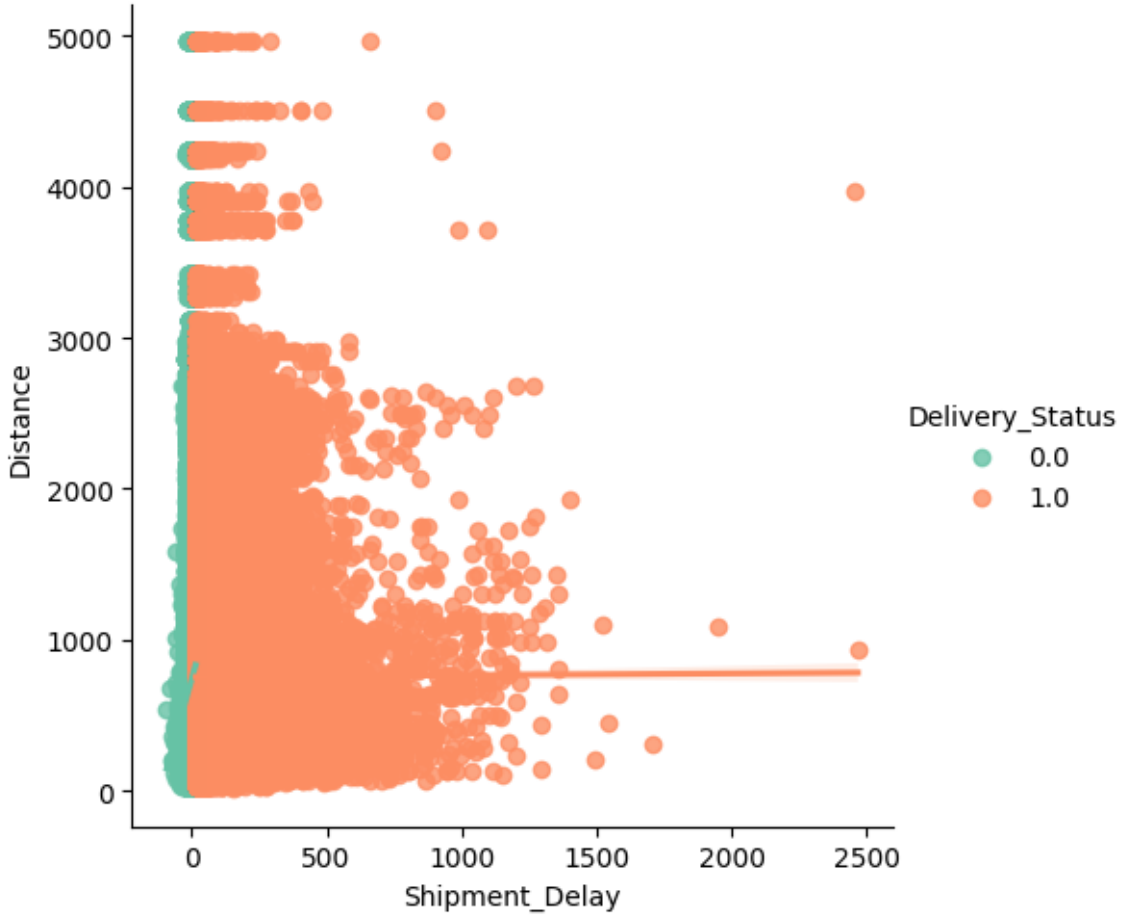
Based on the provided plot, it seems that there is no clear correlation between the actual shipment time and the distance traveled for both delivery statuses (d0 and d1). The density of points is relatively uniform throughout the plot, with no obvious patterns or clusters indicating a relationship between the two variables. However, it is important to note that further analysis may be necessary to fully understand the potential impact of these variables on delivery status and identify opportunities for improvement in the delivery process.

In [ ]:

```
sns.lmplot(x='Shipment_Delay',y='Distance',data=fedex,hue='Delivery_Status',palette='Set2
```

Out[71]:

```
<seaborn.axisgrid.FacetGrid at 0x7f01b4597ac0>
```



conclusion-

Based on the visualizations and data analysis, it can be concluded that the columns 'Actual_Shipment_Time', 'Planned_Shipment_Time', 'Planned_Delivery_Time', 'Planned_TimeofTravel', 'Shipment_Delay', and 'Distance' have the most significant effect on the 'Delivery_Status' column. Specifically, delays in shipment and longer distances seem to be associated with a higher likelihood of delivery cancellations (1 in the 'Delivery_Status' column). Additionally, there appears to be a relationship between planned and actual shipment times and the likelihood of delivery cancellation, with delays in these times also potentially resulting in higher cancellation rates. Overall, improving shipment processes and reducing delays may be key strategies for improving delivery performance and customer satisfaction.

In [ ]:

```
print(list(fedex))
```

```
['Year', 'Month', 'DayofMonth', 'DayOfWeek', 'Actual_Shipment_Time', 'Plan
ned_Shipment_Time', 'Planned_Delivery_Time', 'Carrier_Name', 'Carrier_Nu
m', 'Planned_TimeofTravel', 'Shipment_Delay', 'Source', 'Destination', 'Di
stance', 'Delivery_Status', 'Actual_Shipment_hour', 'Planned_Shipment_hou
r', 'Planned_Delivery_hour']
```

In [ ]:

```
fedex=fedex.reindex(columns=['Year', 'Month', 'DayofMonth', 'DayOfWeek','Carrier_Name', '
```

In [ ]:

```
#MACHINE LEARNING MODELS
```

In [ ]:

```
fedex.head()
```

Out[74]:

	Year	Month	DayofMonth	DayOfWeek	Carrier_Name	Carrier_Num	Source	Destination	A
0	2008	1	3	4	WN	335	IAD	TPA	
1	2008	1	3	4	WN	3231	IAD	TPA	
2	2008	1	3	4	WN	448	IND	BWI	
3	2008	1	3	4	WN	1746	IND	BWI	
4	2008	1	3	4	WN	3920	IND	BWI	

In [ ]:

```
x=fedex.iloc[:,8:-1]
y=fedex['Delivery_Status']
```

In [ ]:

```
x.head()
```

Out[76]:

	Actual_Shipment_Time	Planned_Shipment_Time	Planned_Delivery_Time	Planned_TimeofTra
0	20	19	22	15
1	7	7	10	14
2	6	6	7	9
3	9	9	11	9
4	18	17	19	9

In [ ]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=42)
```

In [ ]:

```
from sklearn.linear_model import LogisticRegression
model_lr = LogisticRegression()
from sklearn.ensemble import RandomForestClassifier
model_rf = RandomForestClassifier()
from sklearn.tree import DecisionTreeClassifier
model_dt=DecisionTreeClassifier()
from sklearn.svm import SVC
model_sv = SVC()
from sklearn.neighbors import KNeighborsClassifier
model_kn = KNeighborsClassifier()
from sklearn.metrics import accuracy_score,precision_score
```

Logistic Regression Model

In [ ]:

```
model_lr.fit(xtrain,ytrain)
pred_lr=model_lr.predict(xtest)
print("accuracy_score",accuracy_score(ytest,pred_lr)*100)
print("precision_score LogisticRegression:",precision_score(ytest,pred_lr))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:
458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(

accuracy_score 100.0
precision_score LogisticRegression: 1.0
```

K-Nearest Neghbours Classifier Model

In [ ]:

```
model_kn.fit(xtrain,ytrain)
pred_kn=model_kn.predict(xtest)
print("accuracy_score",accuracy_score(ytest,pred_kn)*100)
print("precision_score KNeighborsClassifier",precision_score(ytest,pred_kn))
```

```
accuracy_score 99.53821016335124
precision_score KNeighborsClassifier 0.9919661793002711
```

Random Forest Classifier Model

In [ ]:

```
model_rf.fit(xtrain,ytrain)
pred_rf=model_rf.predict(xtest)
print("accuracy_score",accuracy_score(ytest,pred_rf)*100)
print("precision_score RandomForestClassifier:",precision_score(ytest,pred_rf))
```

```
accuracy_score 100.0
precision_score RandomForestClassifier: 1.0
```

Decision Tree Classifier Model

In [ ]:

```
model_dt.fit(xtrain,ytrain)
pred_dt=model_dt.predict(xtest)
print("accuracy_score",accuracy_score(ytest,pred_dt)*100)
print("precision_score DecisionTreeClassifier:",precision_score(ytest,pred_dt))
```

```
accuracy_score 100.0
precision_score DecisionTreeClassifier: 1.0
```

conclusion-

An accuracy score of 100% suggests that the random forest and decision tree classifiers were able to predict the delivery status accurately for all the test samples. However, it is important to be cautious about such high accuracy scores as they may indicate overfitting to the training data. Therefore, it would be advisable to perform additional testing on unseen data to validate the performance of these models.

The precision score of 1 for both models indicates that the models had no false positives, i.e., all the predicted positive values were actually positive. However, it is important to evaluate other performance metrics as well, such as recall, F1 score, and ROC AUC, to get a more comprehensive understanding of the model's performance