Classification of Flights as Delayed

Obtain Data

Import Libs and Read Data

In [2]:
```
import pandas as pd


data = pd.read_csv('./data/2018.csv')
```

Columns

FL_DATE - Date of Flight
OP_CARRIER - Flight Carrier
OP_CARRIER_FL_NUM - Flight Carrier Identifier
ORIGIN- Start Airport
DEST- Destination Airport
CRS_DEP_TIME - Computer Reservation System (CRS) Departure Time
DEP_TIME - Actual Departure Time
DEP_DELAY - Dep Time minus CRS Dep Time in Min
TAXI_OUT - Time To taxi
WHEELS_OFF - Time Wheels in Air
WHEELS_ON - Time Wheels on Ground
TAXI_IN - Time To taxi
CRS_ARR_TIME - Computer Reservation System (CRS) Arrival Time
ARR_TIME - Actual Arrival Time
ARR_DELAY - ARR_Time minus CRS_ARR_TIME in Min
CANCELLED - Flight Cancelled or not
CANCELLATION_CODE - Cancell Code
DIVERTED - Flight Was diverted or Not
CRS_ELAPSED_TIME -CRS scheduled Flight Time
ACTUAL_ELAPSED_TIME - Actual Flight Time
AIR_TIME - Time in the Air
DISTANCE - Distance of Flight

CARRIER_DELAY - Carrier Delay in Min

WEATHER_DELAY - Weather Delay in Min

CANCELLATION_CODE - Cancelled Code

NAS_DELAY - NationalAir Service Delay in Min

SECURITY_DELAY - Sec Delay in Min

LATE_AIRCRAFT_DELAY - Delay due to late Aircraft in Min

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7213446 entries, 0 to 7213445
Data columns (total 28 columns):
 #   Column               Dtype  
---  ------               -----  
 0   FL_DATE              object 
 1   OP_CARRIER           object 
 2   OP_CARRIER_FL_NUM    int64  
 3   ORIGIN               object 
 4   DEST                 object 
 5   CRS_DEP_TIME         int64  
 6   DEP_TIME             float64
 7   DEP_DELAY            float64
 8   TAXI_OUT             float64
 9   WHEELS_OFF           float64
 10  WHEELS_ON            float64
 11  TAXI_IN              float64
 12  CRS_ARR_TIME         int64  
 13  ARR_TIME             float64
 14  ARR_DELAY            float64
 15  CANCELLED            float64
 16  CANCELLATION_CODE    object 
 17  DIVERTED             float64
 18  CRS_ELAPSED_TIME     float64
 19  ACTUAL_ELAPSED_TIME  float64
 20  AIR_TIME             float64
 21  DISTANCE             float64
 22  CARRIER_DELAY        float64
 23  WEATHER_DELAY        float64
 24  NAS_DELAY            float64
 25  SECURITY_DELAY       float64
 26  LATE_AIRCRAFT_DELAY  float64
 27  Unnamed: 27          float64
dtypes: float64(20), int64(3), object(5)
memory usage: 1.5+ GB
```

7.2M records and 28 columns for major flights in 2018 Data on airlines, airport, flight number, etc Time is in minutes

In [4]:
```
data.describe().T
```

Out[4]:

	count	mean	std	min	25%	50%	75%	max
OP_CARRIER_FL_NUM	7213446.0	2607.531335	1860.122265	1.0	1029.0	2131.0	4074.0	7909.0
CRS_DEP_TIME	7213446.0	1329.687018	490.931982	1.0	915.0	1320.0	1735.0	2359.0
DEP_TIME	7101129.0	1333.853806	504.505548	1.0	916.0	1326.0	1744.0	2400.0
DEP_DELAY	7096212.0	9.969858	44.829641	-122.0	-5.0	-2.0	7.0	2710.0
TAXI_OUT	7097616.0	17.410614	9.920409	1.0	11.0	15.0	20.0	196.0
WHEELS_OFF	7097617.0	1357.798878	505.972136	1.0	932.0	1340.0	1759.0	2400.0
WHEELS_ON	7094200.0	1462.162009	533.467516	1.0	1044.0	1502.0	1911.0	2400.0
TAXI_IN	7094200.0	7.601246	6.064797	1.0	4.0	6.0	9.0	259.0
CRS_ARR_TIME	7213446.0	1486.341099	518.312428	1.0	1100.0	1515.0	1919.0	2400.0
ARR_TIME	7094201.0	1466.784165	537.708924	1.0	1049.0	1506.0	1916.0	2400.0
ARR_DELAY	7076406.0	5.048581	46.926637	-120.0	-14.0	-6.0	8.0	2692.0
CANCELLED	7213446.0	0.016162	0.126098	0.0	0.0	0.0	0.0	1.0
DIVERTED	7213446.0	0.002476	0.049696	0.0	0.0	0.0	0.0	1.0
CRS_ELAPSED_TIME	7213436.0	141.135648	73.344332	-99.0	88.0	122.0	171.0	704.0
ACTUAL_ELAPSED_TIME	7079004.0	136.499938	73.137578	14.0	83.0	118.0	167.0	757.0
AIR_TIME	7079004.0	111.502048	71.112927	7.0	60.0	92.0	141.0	696.0
DISTANCE	7213446.0	799.989490	598.178288	31.0	363.0	632.0	1034.0	4983.0
CARRIER_DELAY	1352710.0	19.455006	58.908119	0.0	0.0	0.0	17.0	2109.0
WEATHER_DELAY	1352710.0	3.636459	29.996006	0.0	0.0	0.0	0.0	2692.0
NAS_DELAY	1352710.0	15.885471	35.893497	0.0	0.0	3.0	20.0	1848.0
SECURITY_DELAY	1352710.0	0.093539	3.174306	0.0	0.0	0.0	0.0	987.0
LATE_AIRCRAFT_DELAY	1352710.0	25.644120	49.787761	0.0	0.0	3.0	31.0	2454.0
Unnamed: 27	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [5]:
```
data.columns
```

Out[5]:
```
Index(['FL_DATE', 'OP_CARRIER', 'OP_CARRIER_FL_NUM', 'ORIGIN', 'DEST',
       'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY', 'TAXI_OUT', 'WHEELS_OFF',
       'WHEELS_ON', 'TAXI_IN', 'CRS_ARR_TIME', 'ARR_TIME', 'ARR_DELAY',
       'CANCELLED', 'CANCELLATION_CODE', 'DIVERTED', 'CRS_ELAPSED_TIME',
       'ACTUAL_ELAPSED_TIME', 'AIR_TIME', 'DISTANCE', 'CARRIER_DELAY',
       'WEATHER_DELAY', 'NAS_DELAY', 'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY',
       'Unnamed: 27'],
      dtype='object')
```

In [6]:
```
data.head(10)
```

Out[6]:

	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	...	CRS_ELAP
0	2018-01-01	UA	2429	EWR	DEN	1517	1512.0	-5.0	15.0	1527.0	...	
1	2018-01-01	UA	2427	LAS	SFO	1115	1107.0	-8.0	11.0	1118.0	...	
2	2018-01-01	UA	2426	SNA	DEN	1335	1330.0	-5.0	15.0	1345.0	...	
3	2018-01-01	UA	2425	RSW	ORD	1546	1552.0	6.0	19.0	1611.0	...	
4	2018-01-01	UA	2424	ORD	ALB	630	650.0	20.0	13.0	703.0	...	
5	2018-01-01	UA	2422	ORD	OMA	2241	2244.0	3.0	15.0	2259.0	...	
6	2018-01-01	UA	2421	IAH	LAS	750	747.0	-3.0	14.0	801.0	...	
7	2018-01-01	UA	2420	DEN	CID	1324	1318.0	-6.0	11.0	1329.0	...	
8	2018-01-01	UA	2419	SMF	EWR	2224	2237.0	13.0	10.0	2247.0	...	
9	2018-01-01	UA	2418	RIC	DEN	1601	1559.0	-2.0	12.0	1611.0	...	

10 rows × 28 columns

In [7]:
```
# Check unique values in OP_CARRIER (airline) column
data.OP_CARRIER.unique()
```

Out[7]:
```
array(['UA', 'AS', '9E', 'B6', 'EV', 'F9', 'G4', 'HA', 'MQ', 'NK', 'OH',
       'OO', 'VX', 'WN', 'YV', 'YX', 'AA', 'DL'], dtype=object)
```

In [8]:
```
# Renaming airline codes to company names
# Source: https://en.wikipedia.org/wiki/List_of_airlines_of_the_United_States
data['OP_CARRIER'].replace({
    'UA':'United Airlines',
    'AS':'Alaska Airlines',
    '9E':'Endeavor Air',
    'B6':'JetBlue Airways',
    'EV':'ExpressJet',
    'F9':'Frontier Airlines',
    'G4':'Allegiant Air',
    'HA':'Hawaiian Airlines',
    'MQ':'Envoy Air',
    'NK':'Spirit Airlines',
    'OH':'PSA Airlines',
    'OO':'SkyWest Airlines',
    'VX':'Virgin America',
    'WN':'Southwest Airlines',
    'YV':'Mesa Airline',
    'YX':'Republic Airways',
    'AA':'American Airlines',
    'DL':'Delta Airlines'
},inplace=True)
```

In [9]:
```
data.OP_CARRIER.unique()
```

Out[9]:
```
array(['United Airlines', 'Alaska Airlines', 'Endeavor Air',
       'JetBlue Airways', 'ExpressJet', 'Frontier Airlines',
       'Allegiant Air', 'Hawaiian Airlines', 'Envoy Air',
       'Spirit Airlines', 'PSA Airlines', 'SkyWest Airlines',
       'Virgin America', 'Southwest Airlines', 'Mesa Airline',
       'Republic Airways', 'American Airlines', 'Delta Airlines'],
      dtype=object)
```

In [ ]:

Scrub

In [10]:
```
data.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
OP_CARRIER_FL_NUM	7213446.0	2607.531335	1860.122265	1.0	1029.0	2131.0	4074.0	7909.0
CRS_DEP_TIME	7213446.0	1329.687018	490.931982	1.0	915.0	1320.0	1735.0	2359.0
DEP_TIME	7101129.0	1333.853806	504.505548	1.0	916.0	1326.0	1744.0	2400.0
DEP_DELAY	7096212.0	9.969858	44.829641	-122.0	-5.0	-2.0	7.0	2710.0
TAXI_OUT	7097616.0	17.410614	9.920409	1.0	11.0	15.0	20.0	196.0
WHEELS_OFF	7097617.0	1357.798878	505.972136	1.0	932.0	1340.0	1759.0	2400.0
WHEELS_ON	7094200.0	1462.162009	533.467516	1.0	1044.0	1502.0	1911.0	2400.0
TAXI_IN	7094200.0	7.601246	6.064797	1.0	4.0	6.0	9.0	259.0
CRS_ARR_TIME	7213446.0	1486.341099	518.312428	1.0	1100.0	1515.0	1919.0	2400.0
ARR_TIME	7094201.0	1466.784165	537.708924	1.0	1049.0	1506.0	1916.0	2400.0
ARR_DELAY	7076406.0	5.048581	46.926637	-120.0	-14.0	-6.0	8.0	2692.0
CANCELLED	7213446.0	0.016162	0.126098	0.0	0.0	0.0	0.0	1.0
DIVERTED	7213446.0	0.002476	0.049696	0.0	0.0	0.0	0.0	1.0
CRS_ELAPSED_TIME	7213436.0	141.135648	73.344332	-99.0	88.0	122.0	171.0	704.0
ACTUAL_ELAPSED_TIME	7079004.0	136.499938	73.137578	14.0	83.0	118.0	167.0	757.0
AIR_TIME	7079004.0	111.502048	71.112927	7.0	60.0	92.0	141.0	696.0
DISTANCE	7213446.0	799.989490	598.178288	31.0	363.0	632.0	1034.0	4983.0
CARRIER_DELAY	1352710.0	19.455006	58.908119	0.0	0.0	0.0	17.0	2109.0
WEATHER_DELAY	1352710.0	3.636459	29.996006	0.0	0.0	0.0	0.0	2692.0
NAS_DELAY	1352710.0	15.885471	35.893497	0.0	0.0	3.0	20.0	1848.0
SECURITY_DELAY	1352710.0	0.093539	3.174306	0.0	0.0	0.0	0.0	987.0
LATE_AIRCRAFT_DELAY	1352710.0	25.644120	49.787761	0.0	0.0	3.0	31.0	2454.0
Unnamed: 27	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [11]:
```
data.isna().sum()
```

Out[11]:
```
FL_DATE                   0
OP_CARRIER                0
OP_CARRIER_FL_NUM         0
ORIGIN                    0
DEST                      0
CRS_DEP_TIME              0
DEP_TIME             112317
DEP_DELAY            117234
TAXI_OUT             115830
WHEELS_OFF           115829
WHEELS_ON            119246
TAXI_IN              119246
CRS_ARR_TIME              0
ARR_TIME             119245
ARR_DELAY            137040
CANCELLED                 0
CANCELLATION_CODE    7096862
DIVERTED                  0
CRS_ELAPSED_TIME         10
ACTUAL_ELAPSED_TIME  134442
AIR_TIME             134442
DISTANCE                  0
CARRIER_DELAY        5860736
WEATHER_DELAY        5860736
NAS_DELAY            5860736
SECURITY_DELAY       5860736
LATE_AIRCRAFT_DELAY  5860736
Unnamed: 27          7213446
dtype: int64
```

Cancelled Flights

In [12]:
```
# Total canceled flights
data.CANCELLED.sum()
```

Out[12]:
```
116584.0
```

In [13]:
```
# CANCELLED
data.CANCELLED.unique()
```

Out[13]:
```
array([0., 1.])
```

In [14]:
```
canceled = data[(data['CANCELLED'] > 0)]
```

In [15]:
```
# Set to see all columns
pd.set_option('display.max_columns', None)
```

In [16]:
```
canceled.head(10)
```

Out[16]:

	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEELS_(
178	2018-01-01	United Airlines	2034	IAH	MFE	1440	NaN	NaN	NaN	NaN	N
875	2018-01-01	United Airlines	864	LAS	SFO	1744	NaN	NaN	NaN	NaN	N
1244	2018-01-01	United Airlines	488	MFE	IAH	1726	NaN	NaN	NaN	NaN	N
1584	2018-01-01	Alaska Airlines	5	DCA	LAX	910	NaN	NaN	NaN	NaN	N
1639	2018-01-01	Alaska Airlines	64	WRG	KTN	1623	NaN	NaN	NaN	NaN	N
1642	2018-01-01	Alaska Airlines	65	WRG	PSG	1056	NaN	NaN	NaN	NaN	N
1644	2018-01-01	Alaska Airlines	65	PSG	JNU	1204	NaN	NaN	NaN	NaN	N
1655	2018-01-01	Alaska Airlines	73	SIT	JNU	600	NaN	NaN	NaN	NaN	N
1705	2018-01-01	Alaska Airlines	162	FAI	ANC	545	NaN	NaN	NaN	NaN	N
1731	2018-01-01	Alaska Airlines	313	SFO	SEA	500	NaN	NaN	NaN	NaN	N

In [17]:
```
# OPTIONAL: Leaving only non-canceled flights
data = data[(data['CANCELLED'] == 0)]
```

In [18]:
```
#Removed 116584 cancelled flights
```

In [19]:
```
data['LATE_AIRCRAFT_DELAY'].fillna(0,inplace=True)
```

In [20]:
```
delays = ['CARRIER_DELAY', 'WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY']

for delay in delays:
    data[delay].fillna(0,inplace=True)
```

In [21]:
```
data['DEP_DELAY'].fillna(0,inplace=True)
```

In [22]:
```
data.isna().sum()
```

Out[22]:
```
FL_DATE                   0
OP_CARRIER                0
OP_CARRIER_FL_NUM         0
ORIGIN                    0
DEST                      0
CRS_DEP_TIME              0
DEP_TIME                  0
DEP_DELAY                 0
TAXI_OUT                  0
WHEELS_OFF                0
WHEELS_ON              2662
TAXI_IN                2662
CRS_ARR_TIME              0
ARR_TIME               2661
ARR_DELAY             20456
CANCELLED                 0
CANCELLATION_CODE   7096862
DIVERTED                  0
CRS_ELAPSED_TIME          7
ACTUAL_ELAPSED_TIME   17858
AIR_TIME              17858
DISTANCE                  0
CARRIER_DELAY             0
WEATHER_DELAY             0
NAS_DELAY                 0
SECURITY_DELAY            0
LATE_AIRCRAFT_DELAY       0
Unnamed: 27         7096862
dtype: int64
```

In [23]:
```
#data['ARR_DELAY'] = data['ARR_TIME'] - data['CRS_ARR_TIME']
```

In [24]:
```
data.isna().sum()
```

Out[24]:
```
FL_DATE                   0
OP_CARRIER                0
OP_CARRIER_FL_NUM         0
```

```
ORIGIN                     0
DEST                       0
CRS_DEP_TIME               0
DEP_TIME                   0
DEP_DELAY                  0
TAXI_OUT                   0
WHEELS_OFF                 0
WHEELS_ON               2662
TAXI_IN                 2662
CRS_ARR_TIME               0
ARR_TIME                2661
ARR_DELAY              20456
CANCELLED                  0
CANCELLATION_CODE    7096862
DIVERTED                   0
CRS_ELAPSED_TIME           7
ACTUAL_ELAPSED_TIME    17858
AIR_TIME               17858
DISTANCE                   0
CARRIER_DELAY              0
WEATHER_DELAY              0
NAS_DELAY                  0
SECURITY_DELAY             0
LATE_AIRCRAFT_DELAY        0
Unnamed: 27          7096862
dtype: int64
```

In [25]:
```
data.dropna(subset=['WHEELS_ON', 'TAXI_IN','ARR_TIME','ARR_DELAY','CRS_ELAPSED_TIME','ACTUAL_ELAPSED_TIME', 'AIR_TIME'],i
```

In [26]:
```
data.isna().sum()
```

Out[26]:
```
FL_DATE                    0
OP_CARRIER                 0
OP_CARRIER_FL_NUM          0
ORIGIN                     0
DEST                       0
CRS_DEP_TIME               0
DEP_TIME                   0
DEP_DELAY                  0
TAXI_OUT                   0
WHEELS_OFF                 0
WHEELS_ON                  0
TAXI_IN                    0
CRS_ARR_TIME               0
ARR_TIME                   0
ARR_DELAY                  0
CANCELLED                  0
CANCELLATION_CODE    7076405
```

```
DIVERTED                   0
CRS_ELAPSED_TIME           0
ACTUAL_ELAPSED_TIME        0
AIR_TIME                   0
DISTANCE                   0
CARRIER_DELAY              0
WEATHER_DELAY              0
NAS_DELAY                  0
SECURITY_DELAY             0
LATE_AIRCRAFT_DELAY        0
Unnamed: 27          7076405
dtype: int64
```

In [27]:
```
data.drop(columns=['CANCELLED','CANCELLATION_CODE'],inplace=True)
```

In [28]:
```
data.reset_index(inplace=True)
```

In [29]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7076405 entries, 0 to 7076404
Data columns (total 27 columns):
 #   Column               Dtype  
---  ------               -----  
 0   index                int64  
 1   FL_DATE              object 
 2   OP_CARRIER           object 
 3   OP_CARRIER_FL_NUM    int64  
 4   ORIGIN               object 
 5   DEST                 object 
 6   CRS_DEP_TIME         int64  
 7   DEP_TIME             float64
 8   DEP_DELAY            float64
 9   TAXI_OUT             float64
 10  WHEELS_OFF           float64
 11  WHEELS_ON            float64
 12  TAXI_IN              float64
 13  CRS_ARR_TIME         int64  
 14  ARR_TIME             float64
 15  ARR_DELAY            float64
 16  DIVERTED             float64
 17  CRS_ELAPSED_TIME     float64
 18  ACTUAL_ELAPSED_TIME  float64
 19  AIR_TIME             float64
 20  DISTANCE             float64
 21  CARRIER_DELAY        float64
 22  WEATHER_DELAY        float64
 23  NAS_DELAY            float64
```

```
 24  SECURITY_DELAY       float64
 25  LATE_AIRCRAFT_DELAY  float64
 26  Unnamed: 27          float64
dtypes: float64(19), int64(4), object(4)
memory usage: 1.4+ GB
```

In [ ]:

In [ ]:

In [30]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7076405 entries, 0 to 7076404
Data columns (total 27 columns):
 #   Column               Dtype  
---  ------               -----  
 0   index                int64  
 1   FL_DATE              object 
 2   OP_CARRIER           object 
 3   OP_CARRIER_FL_NUM    int64  
 4   ORIGIN               object 
 5   DEST                 object 
 6   CRS_DEP_TIME         int64  
 7   DEP_TIME             float64
 8   DEP_DELAY            float64
 9   TAXI_OUT             float64
 10  WHEELS_OFF           float64
 11  WHEELS_ON            float64
 12  TAXI_IN              float64
 13  CRS_ARR_TIME         int64  
 14  ARR_TIME             float64
 15  ARR_DELAY            float64
 16  DIVERTED             float64
 17  CRS_ELAPSED_TIME     float64
 18  ACTUAL_ELAPSED_TIME  float64
 19  AIR_TIME             float64
 20  DISTANCE             float64
 21  CARRIER_DELAY        float64
 22  WEATHER_DELAY        float64
 23  NAS_DELAY            float64
 24  SECURITY_DELAY       float64
 25  LATE_AIRCRAFT_DELAY  float64
 26  Unnamed: 27          float64
dtypes: float64(19), int64(4), object(4)
memory usage: 1.4+ GB
```

In [31]:
```
data.head(10)
```

Out[31]:

	index	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEE
0	0	2018-01-01	United Airlines	2429	EWR	DEN	1517	1512.0	-5.0	15.0	1527.0	
1	1	2018-01-01	United Airlines	2427	LAS	SFO	1115	1107.0	-8.0	11.0	1118.0	
2	2	2018-01-01	United Airlines	2426	SNA	DEN	1335	1330.0	-5.0	15.0	1345.0	
3	3	2018-01-01	United Airlines	2425	RSW	ORD	1546	1552.0	6.0	19.0	1611.0	
4	4	2018-01-01	United Airlines	2424	ORD	ALB	630	650.0	20.0	13.0	703.0	
5	5	2018-01-01	United Airlines	2422	ORD	OMA	2241	2244.0	3.0	15.0	2259.0	
6	6	2018-01-01	United Airlines	2421	IAH	LAS	750	747.0	-3.0	14.0	801.0	
7	7	2018-01-01	United Airlines	2420	DEN	CID	1324	1318.0	-6.0	11.0	1329.0	
8	8	2018-01-01	United Airlines	2419	SMF	EWR	2224	2237.0	13.0	10.0	2247.0	
9	9	2018-01-01	United Airlines	2418	RIC	DEN	1601	1559.0	-2.0	12.0	1611.0	

In [ ]:

In [32]:
```
data[data.CARRIER_DELAY > 0]
```

Out[32]:

	index	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_O
11	11	2018-01-01	United Airlines	2416	ORD	CLE	2059	2300.0	121.0	24.0	2324

	index	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_O
15	15	2018-01-01	United Airlines	2412	MCO	LAX	653	747.0	54.0	14.0	801
16	16	2018-01-01	United Airlines	2411	EWR	SMF	1810	1922.0	72.0	16.0	193
17	17	2018-01-01	United Airlines	2410	RSW	EWR	1250	1337.0	47.0	12.0	134
27	27	2018-01-01	United Airlines	2398	MSY	EWR	2043	2131.0	48.0	10.0	2141
...	...	...	...	...	...	...	...	...	...	...	
7076351	7213392	2018-12-31	American Airlines	1771	DFW	LAS	1950	2025.0	35.0	22.0	2047
7076363	7213404	2018-12-31	American Airlines	1784	LAX	IAH	720	809.0	49.0	21.0	83
7076371	7213412	2018-12-31	American Airlines	1792	CMH	LAX	1622	1659.0	37.0	16.0	171
7076388	7213429	2018-12-31	American Airlines	1805	CLT	DTW	1632	1720.0	48.0	15.0	173
7076392	7213433	2018-12-31	American Airlines	1809	PHL	CLT	500	514.0	14.0	39.0	553

662152 rows × 27 columns

In [ ]:

In [33]:
```
#Cancelled
print("Percent of CANCELLED is " + str(116584/7213446))
```

```
Percent of CANCELLED is 0.01616203961324449
```

In [34]:
```
delays = ['CARRIER_DELAY', 'WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY']
total = 7213446
numberindel = 0
percent = 0
```

```
for delay in delays:
    numberindel = len(data[data[delay] > 0])
    percent = numberindel/total
    print("Percent of {0} is {1}".format(delay,percent))
```

```
Percent of CARRIER_DELAY is 0.09179413001774742
Percent of WEATHER_DELAY is 0.01179117442620351
Percent of NAS_DELAY is 0.10266396948143786
Percent of SECURITY_DELAY is 0.000602763228559554
Percent of LATE_AIRCRAFT_DELAY is 0.09693993689008
```

In [35]:
```
data[data['CARRIER_DELAY']==0]
```

Out[35]:

	index	FL_DATE	OP_CARRIER	OP_CARRIER_FL_NUM	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_O
0	0	2018-01-01	United Airlines	2429	EWR	DEN	1517	1512.0	-5.0	15.0	1527
1	1	2018-01-01	United Airlines	2427	LAS	SFO	1115	1107.0	-8.0	11.0	1118
2	2	2018-01-01	United Airlines	2426	SNA	DEN	1335	1330.0	-5.0	15.0	1345
3	3	2018-01-01	United Airlines	2425	RSW	ORD	1546	1552.0	6.0	19.0	1611
4	4	2018-01-01	United Airlines	2424	ORD	ALB	630	650.0	20.0	13.0	703
...	...	...	...	...	...	...	...	...	...	...	
7076400	7213441	2018-12-31	American Airlines	1815	DCA	CLT	1534	1530.0	-4.0	20.0	1550
7076401	7213442	2018-12-31	American Airlines	1816	CLT	DFW	1751	1757.0	6.0	18.0	1815
7076402	7213443	2018-12-31	American Airlines	1817	CLT	MEM	2015	2010.0	-5.0	36.0	2046
7076403	7213444	2018-12-31	American Airlines	1818	CLT	RDU	1300	1323.0	23.0	11.0	1334
7076404	7213445	2018-12-31	American Airlines	1818	RDU	CLT	1435	1443.0	8.0	8.0	1451

6414253 rows × 27 columns

In [36]:
```
# Month variable
data['month'] = pd.to_datetime(data['FL_DATE']).dt.month_name()
# Weekday variable
data['day'] = pd.to_datetime(data['FL_DATE']).dt.dayofweek
data['weekday'] = pd.to_datetime(data['FL_DATE']).dt.day_name()
```

In [37]:
```
data.to_csv('./data/2018_flights_tidy.csv',encoding='utf-8')
```

In [ ]:

In [ ]:

In [ ]:

EDA of Data

Load Tidy Data

In [38]:
```
#Load Tidy Data
data = pd.read_csv('./data/2018_flights_tidy.csv')
```

In [39]:
```
data.drop(columns=['Unnamed: 0'],inplace=True)
```

In [40]:
```
data.drop(columns=['index'],inplace=True)
```

In [41]:
```
#Create Feature DELAYED if gt= 15 mins
data['DELAYED'] = (data['ARR_DELAY']>=15).astype(int)
```

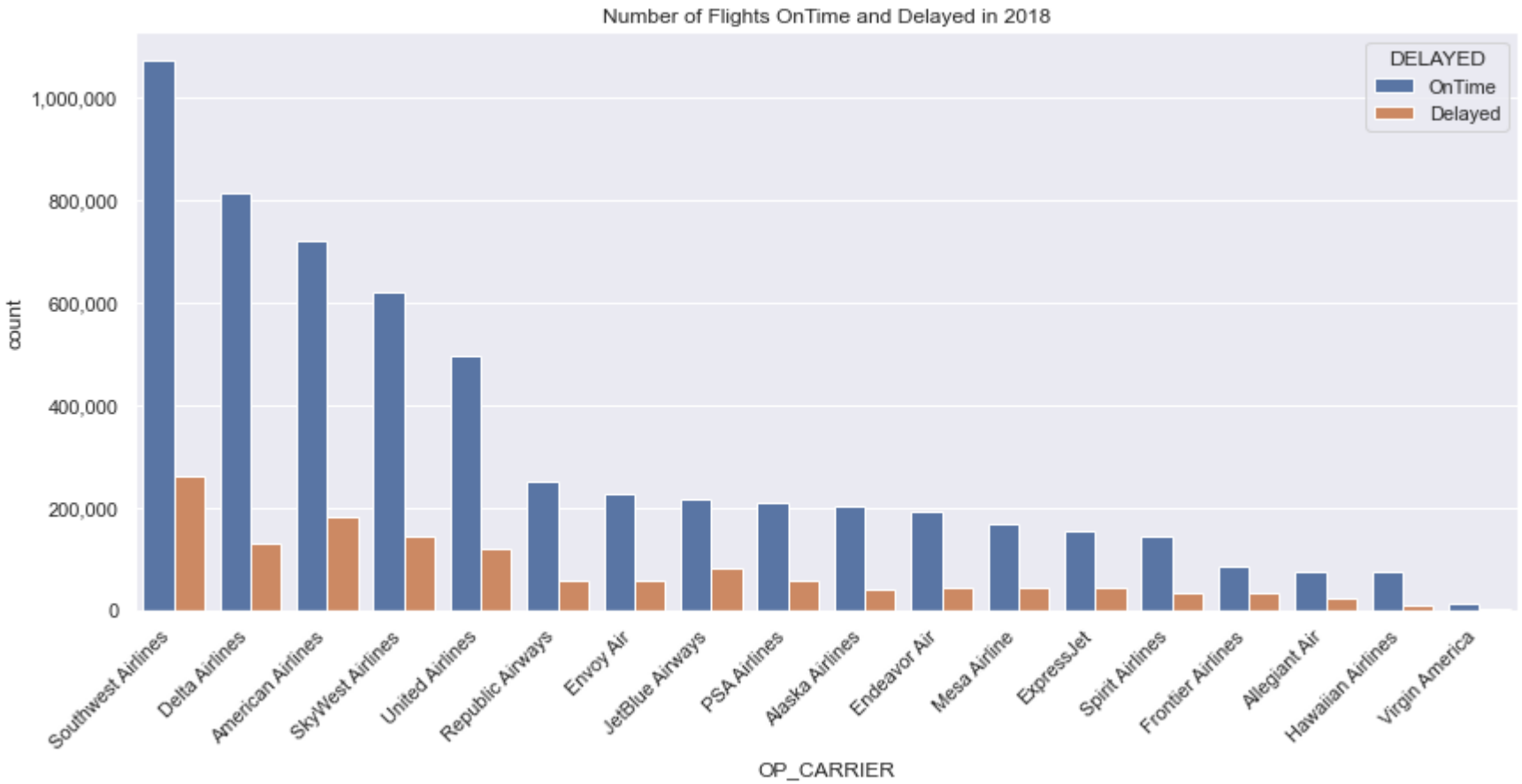
Import Viz Libs

In [42]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
%matplotlib inline
```

OnTime Delayed Numbers

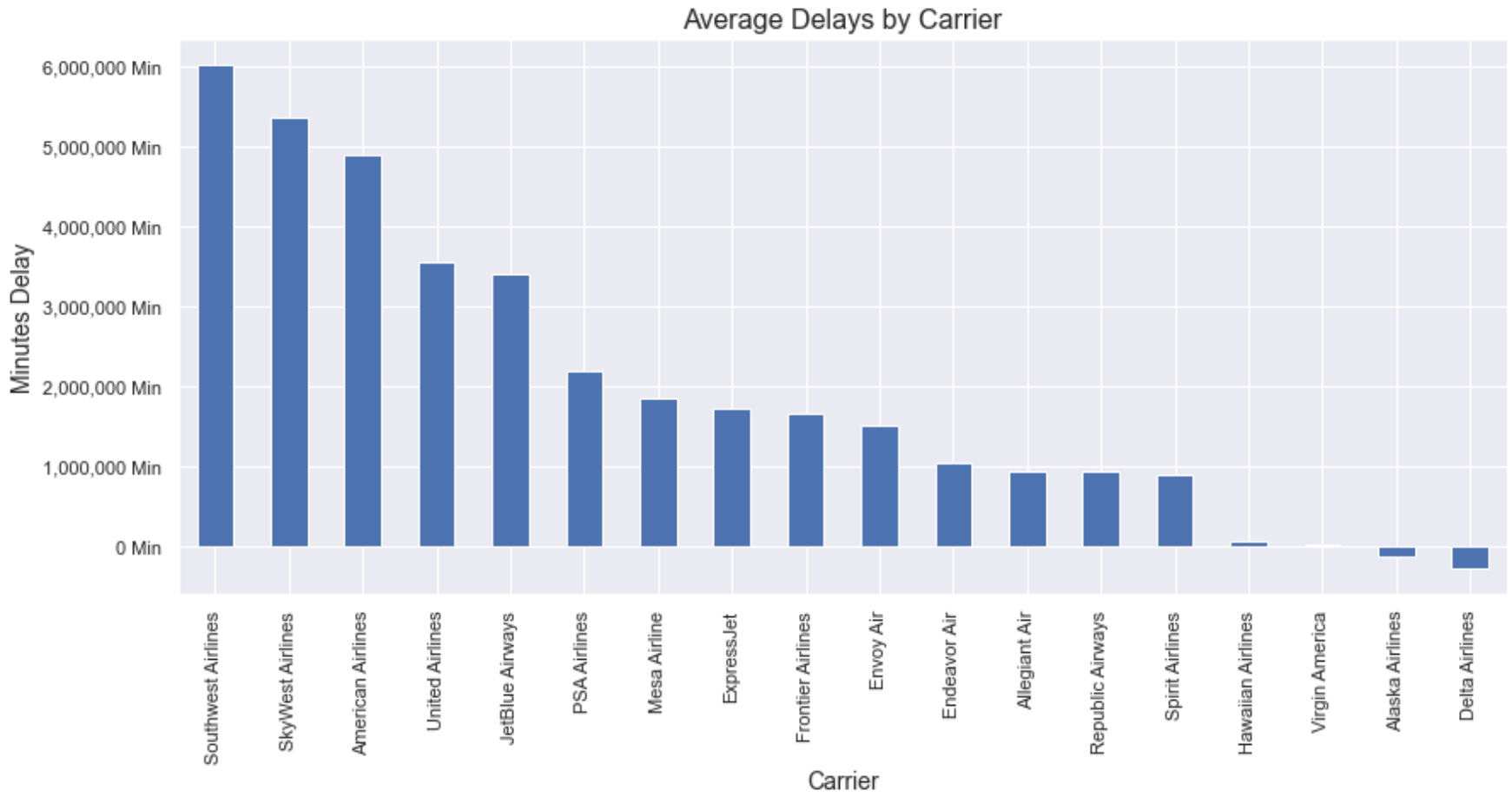
In [43]:
```
order = ['Southwest Airlines', 'Delta Airlines', 'American Airlines',
                        'SkyWest Airlines', 'United Airlines','Republic Airways',
                           'Envoy Air', 'JetBlue Airways', 'PSA Airlines',
                           'Alaska Airlines','Endeavor Air',
                           'Mesa Airline','ExpressJet','Spirit Airlines',
                           'Frontier Airlines','Allegiant Air', 'Hawaiian Airlines','Virgin America']
```

In [44]:
```
sns.set(style="darkgrid")
fig8, ax8 = plt.subplots(figsize=(14,6),nrows=1,ncols=1, )
ax8.legend(['OnTime',"Delayed"])
plt.xticks(rotation = 45, ha = 'right')
ax8.set_title("Number of Flights OnTime and Delayed in 2018")
g = sns.countplot(x='OP_CARRIER', hue='DELAYED', data=data, ax=ax8,
order = ['Southwest Airlines', 'Delta Airlines', 'American Airlines',
                        'SkyWest Airlines', 'United Airlines','Republic Airways',
                           'Envoy Air', 'JetBlue Airways', 'PSA Airlines',
                           'Alaska Airlines','Endeavor Air',
                           'Mesa Airline','ExpressJet','Spirit Airlines',
                           'Frontier Airlines','Allegiant Air', 'Hawaiian Airlines','Virgin America']
)

# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax8.get_yticks().tolist()
ax8.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax8.set_yticklabels(['{:,.0f}'.format(y)  for y in ticks_loc])

# replace labels
new_labels = ['OnTime', 'Delayed']
for t, l in zip(g.legend_.texts, new_labels): t.set_text(l);
```



Average Delay by Carrier

In [45]:
```
sns.set(style="darkgrid")
fig11, ax11 = plt.subplots(figsize=(14, 6))
data.groupby('OP_CARRIER').ARR_DELAY.sum().sort_values(ascending=False).plot.bar()

plt.title('Arrival Delays by Carrier', fontsize=16)
plt.xlabel('Airline', fontsize=14)
plt.ylabel('Minutes Delay', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax11.get_yticks().tolist()
ax11.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax11.set_yticklabels(['{:,.0f}'.format(y) + ' Min' for y in ticks_loc])
plt.show()
```



In [177…
```
data2 = data.loc[data['OP_CARRIER'].isin(['Southwest Airlines','Delta Airlines'])]
```

In [178…
```
data2['OP_CARRIER'].unique()
```

Out[178…
```
array(['Southwest Airlines', 'Delta Airlines'], dtype=object)
```

In [197…
```
data2.groupby('OP_CARRIER').agg({'ARR_DELAY': ['mean', 'min', 'max', 'count','sum','median']})
```

Out[197…

	ARR_DELAY					
	mean	min	max	count	sum	median
OP_CARRIER						
Delta Airlines	-0.289825	-83.0	1255.0	943837	-273548.0	-9.0

	ARR_DELAY					
	mean	min	max	count	sum	median
OP_CARRIER						
Southwest Airlines	4.519644	-117.0	698.0	1331609	6018399.0	-4.0

In [ ]:

In [194…
```
data[data['OP_CARRIER'] == 'Southwest Airlines']['ARR_DELAY'].min()
```

Out[194…
```
4.519644279965065
```

In [165…
```
data[data['OP_CARRIER'] == 'Delta Airlines']['ARR_DELAY'].min()
```

Out[165…
```
-83.0
```

In [159…
```
data[data['OP_CARRIER'] == 'Southwest Airlines']['ARR_DELAY'].max()
```

Out[159…
```
698.0
```

In [160…
```
data[data['OP_CARRIER'] == 'Delta Airlines']['ARR_DELAY'].max()
```

Out[160…
```
1255.0
```

In [157…
```
data[data['OP_CARRIER'] == 'Southwest Airlines']['ARR_DELAY'].mean()
```

Out[157…
```
4.519644279965065
```

In [158…
```
data[data['OP_CARRIER'] == 'Delta Airlines']['ARR_DELAY'].mean()
```

Out[158…
```
-0.2898254677449602
```

In [155…
```
SWestTotalDelayedMins=data[data['OP_CARRIER'] == 'Southwest Airlines']['ARR_DELAY'].sum()
```

In [144…
```
DeltaTotalDelayedMins=data[data['OP_CARRIER'] == 'Delta Airlines']['ARR_DELAY'].sum()
```

In [148…
```
numSWDelayed = data[data['OP_CARRIER'] == 'Southwest Airlines']['DELAYED'].sum()
```

In [149…
```
numDeltaDelayed = data[data['OP_CARRIER'] == 'Delta Airlines']['DELAYED'].sum()
```

In [156…
```
avgSWDelay = SWestTotalDelayedMins / numSWDelayed
avgSWDelay
```

Out[156… 23.116570001920493

In [154…
```
avgDeltaDelay = DeltaTotalDelayedMins / numDeltaDelayed
avgDeltaDelay
```

Out[154… -2.1011606203289066

In [ ]:
```
#Average SWest Airlines Delays 23 mins  Average Delta -2.1 mins
```

In [140…
```
data.groupby('OP_CARRIER').sum()
```

Out[140…

OP_CARRIER	OP_CARRIER_FL_NUM	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEELS_ON	TAXI_IN	CRS_ARR_TI
Alaska Airlines	166776237	323542094	3.222664e+08	589980.0	4525223.0	3.269179e+08	3.562264e+08	1921577.0	3592653
Allegiant Air	97656911	124211944	1.253571e+08	1228190.0	1237818.0	1.272853e+08	1.440005e+08	647982.0	1462933
American Airlines	1363797364	1181771087	1.187688e+09	8990432.0	16901973.0	1.209921e+09	1.319092e+09	8636479.0	13386087
Delta Airlines	1604828513	1256132708	1.257254e+09	6990842.0	16280649.0	1.279877e+09	1.396927e+09	6938634.0	14220837
Endeavor Air	1015886192	311363207	3.131327e+08	2492404.0	5214460.0	3.215118e+08	3.466479e+08	1731522.0	3535152
Envoy Air	1051516180	376353070	3.791613e+08	2175875.0	5326938.0	3.878432e+08	4.148439e+08	2489242.0	4188613
ExpressJet	833419950	259565832	2.613833e+08	2408534.0	3808556.0	2.675635e+08	2.880108e+08	1556681.0	2928446
Frontier Airlines	125448225	154038966	1.548962e+08	2306163.0	1904769.0	1.567258e+08	1.678521e+08	1192206.0	1725807
Hawaiian Airlines	19519306	107387481	1.071411e+08	72863.0	1002513.0	1.088539e+08	1.178509e+08	596678.0	1194244

OP_CARRIER	OP_CARRIER_FL_NUM	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	TAXI_OUT	WHEELS_OFF	WHEELS_ON	TAXI_IN	CRS_ARR_TIM
JetBlue Airways	297361257	406298118	4.033509e+08	4693897.0	5381765.0	4.060897e+08	4.174684e+08	2149815.0	4266301
Mesa Airline	1256693290	278396316	2.801887e+08	2332852.0	3647709.0	2.858318e+08	3.095620e+08	1567838.0	3145382
PSA Airlines	1422249983	356588383	3.598986e+08	3176751.0	4492096.0	3.669103e+08	3.904345e+08	2269744.0	3964402
Republic Airways	1362690700	400645588	4.018236e+08	2221205.0	6385703.0	4.127692e+08	4.501318e+08	2629248.0	4580944
SkyWest Airlines	3546048364	1003880869	1.009763e+09	7600834.0	15567628.0	1.034898e+09	1.112011e+09	5806788.0	11234972
Southwest Airlines	3330379038	1784481810	1.799154e+09	14788796.0	16490612.0	1.824137e+09	1.941650e+09	6980082.0	19712921
Spirit Airlines	102137343	240131574	2.382950e+08	1713309.0	2643205.0	2.401045e+08	2.527750e+08	1720296.0	2595377
United Airlines	757180769	812996278	8.135393e+08	6134104.0	11974875.0	8.265714e+08	8.948394e+08	4760636.0	9101325
Virgin America	25826748	23521459	2.352155e+07	108451.0	345602.0	2.395395e+07	2.609600e+07	148754.0	264569

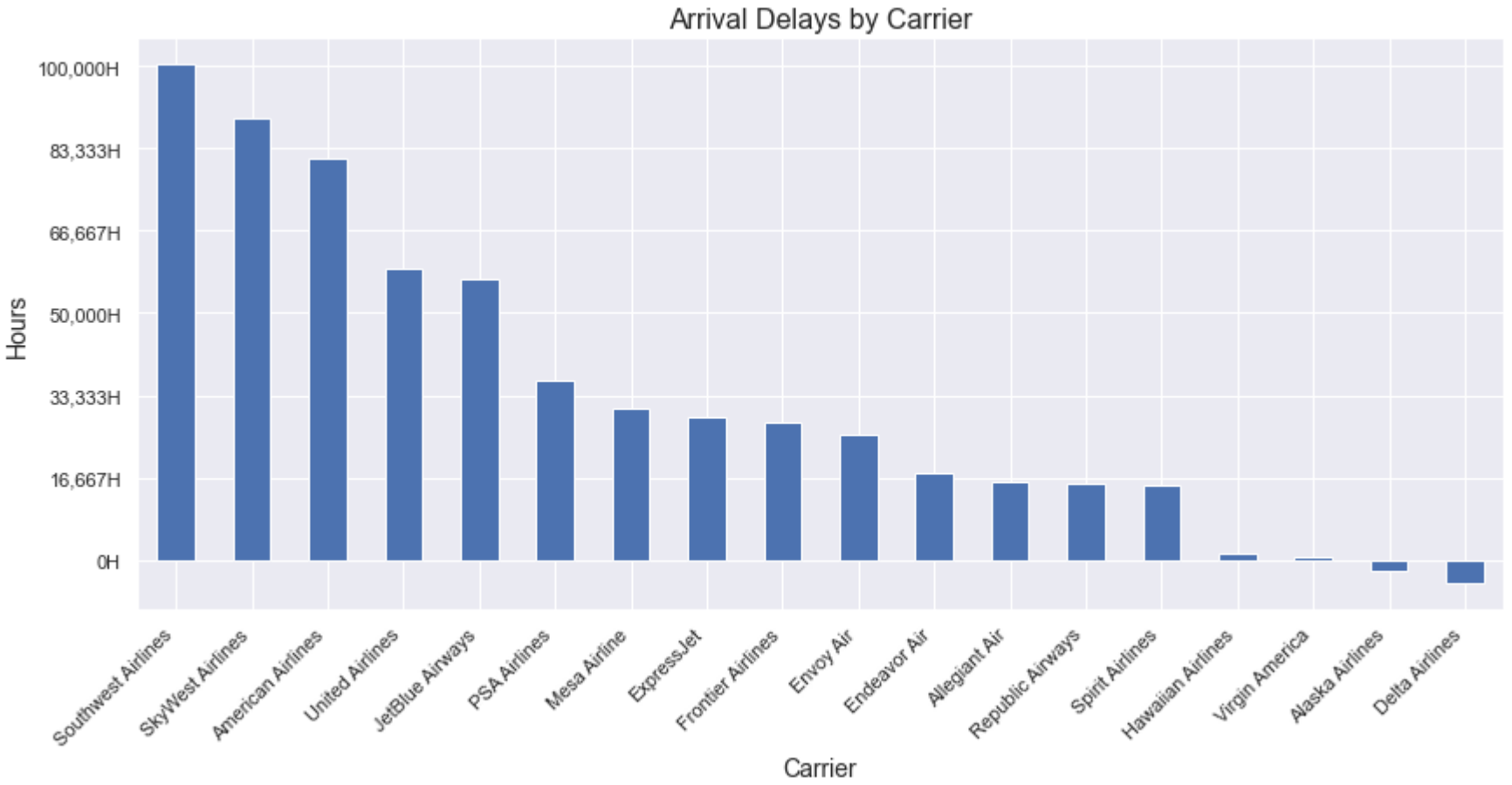
In [ ]:

Viz Arrival Delays by Carrier

In [138…

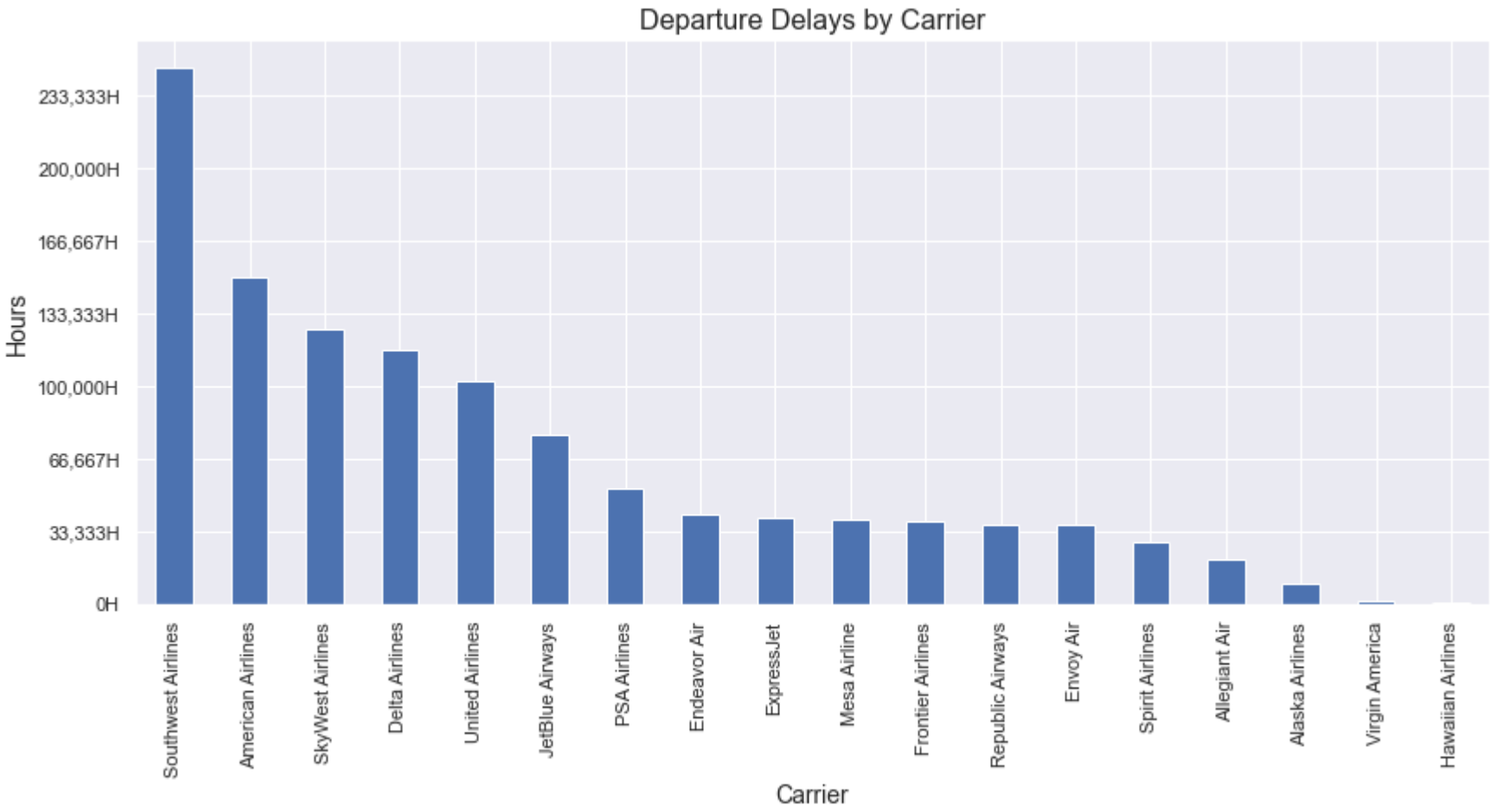
```
sns.set(style="darkgrid")
fig1, ax1 = plt.subplots(figsize=(14, 6))
data.groupby('OP_CARRIER').ARR_DELAY.sum().sort_values(ascending=False).plot.bar()
plt.xticks(rotation = 45, ha = 'right')
plt.title('Arrival Delays by Carrier', fontsize=16)
plt.xlabel('Carrier', fontsize=14)
plt.ylabel('Hours', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax1.get_yticks().tolist()
ax1.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
```

```
ax1.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])
plt.show()
```



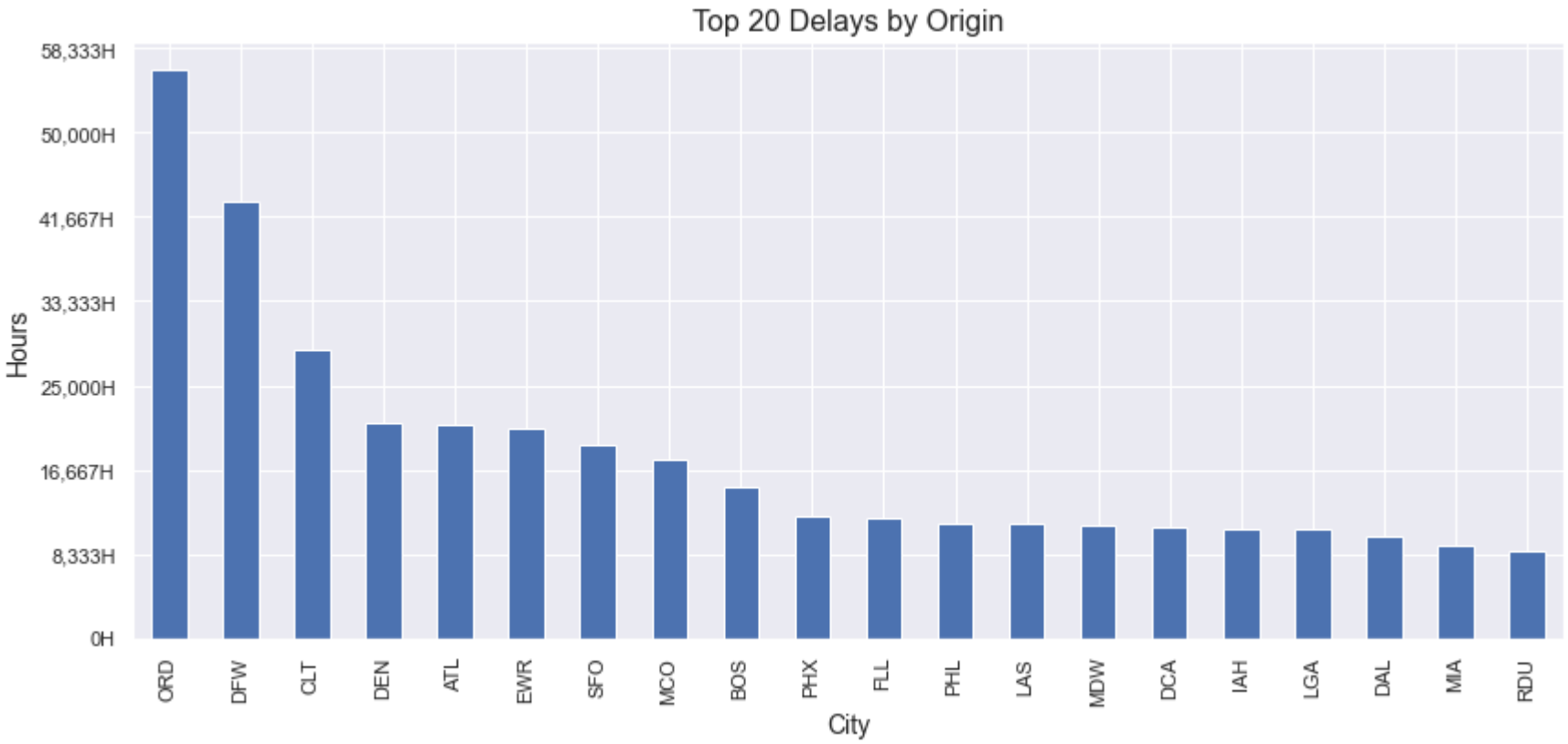
Viz Departure Delays by Carrier

In [49]:
```
fig2, ax2 = plt.subplots(figsize=(14, 6))
data.groupby('OP_CARRIER').DEP_DELAY.sum().sort_values(ascending=False).plot.bar()

plt.title('Departure Delays by Carrier', fontsize=16)
plt.xlabel('Carrier', fontsize=14)
plt.ylabel('Hours', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax2.get_yticks().tolist()
ax2.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax2.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])
plt.show()
```



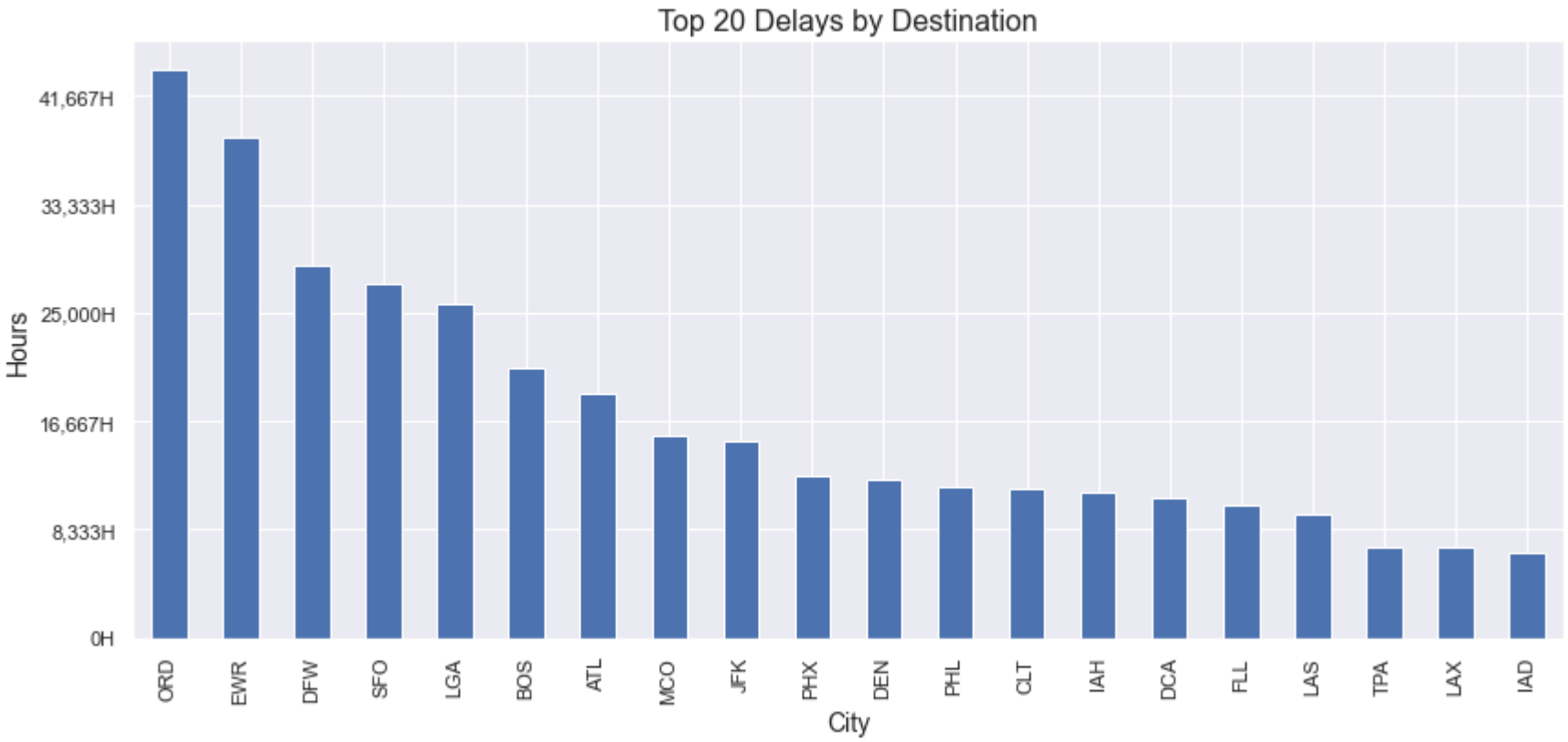
Top 20 Delays by Origin

In [50]:
```
# Delays by City
fig3, ax3 = plt.subplots(figsize=(14, 6))
origins = data.groupby('ORIGIN').ARR_DELAY.sum().sort_values(ascending=False)

origins[:20].plot.bar()
plt.title('Top 20 Delays by Origin', fontsize=16)
plt.xlabel('City', fontsize=14)
plt.ylabel('Hours', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax3.get_yticks().tolist()
ax3.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax3.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])
plt.show()
```



Top 20 Delays by Destination

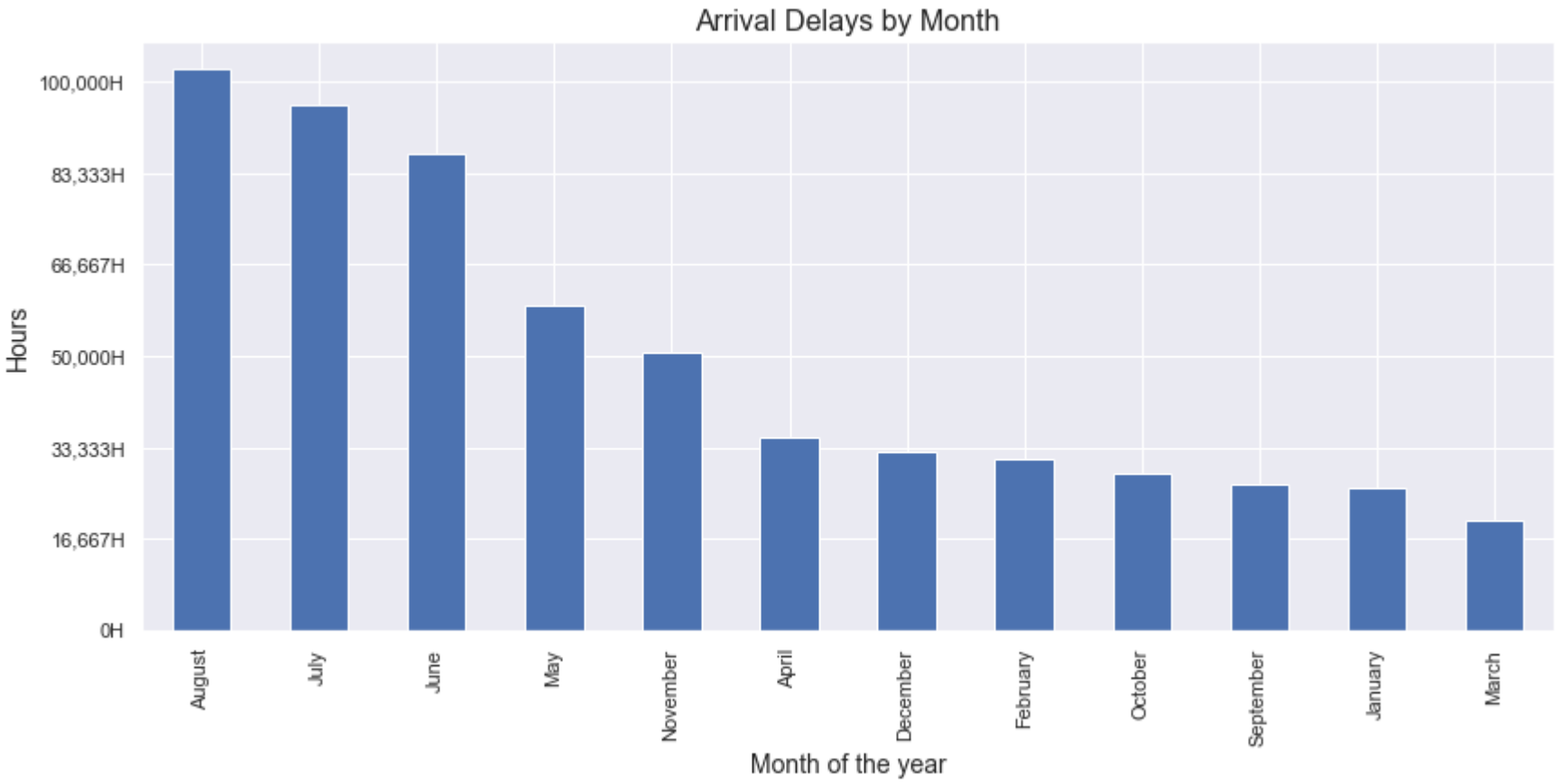
In [51]:
```
# Delays by City
fig4, ax4 = plt.subplots(figsize=(14, 6))
origins = data.groupby('DEST').ARR_DELAY.sum().sort_values(ascending=False)


origins[:20].plot.bar()
plt.title('Top 20 Delays by Destination', fontsize=16)
plt.xlabel('City', fontsize=14)
plt.ylabel('Hours', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax4.get_yticks().tolist()
ax4.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax4.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])
plt.show()
```



Arrival Delays by Month

In [ ]:

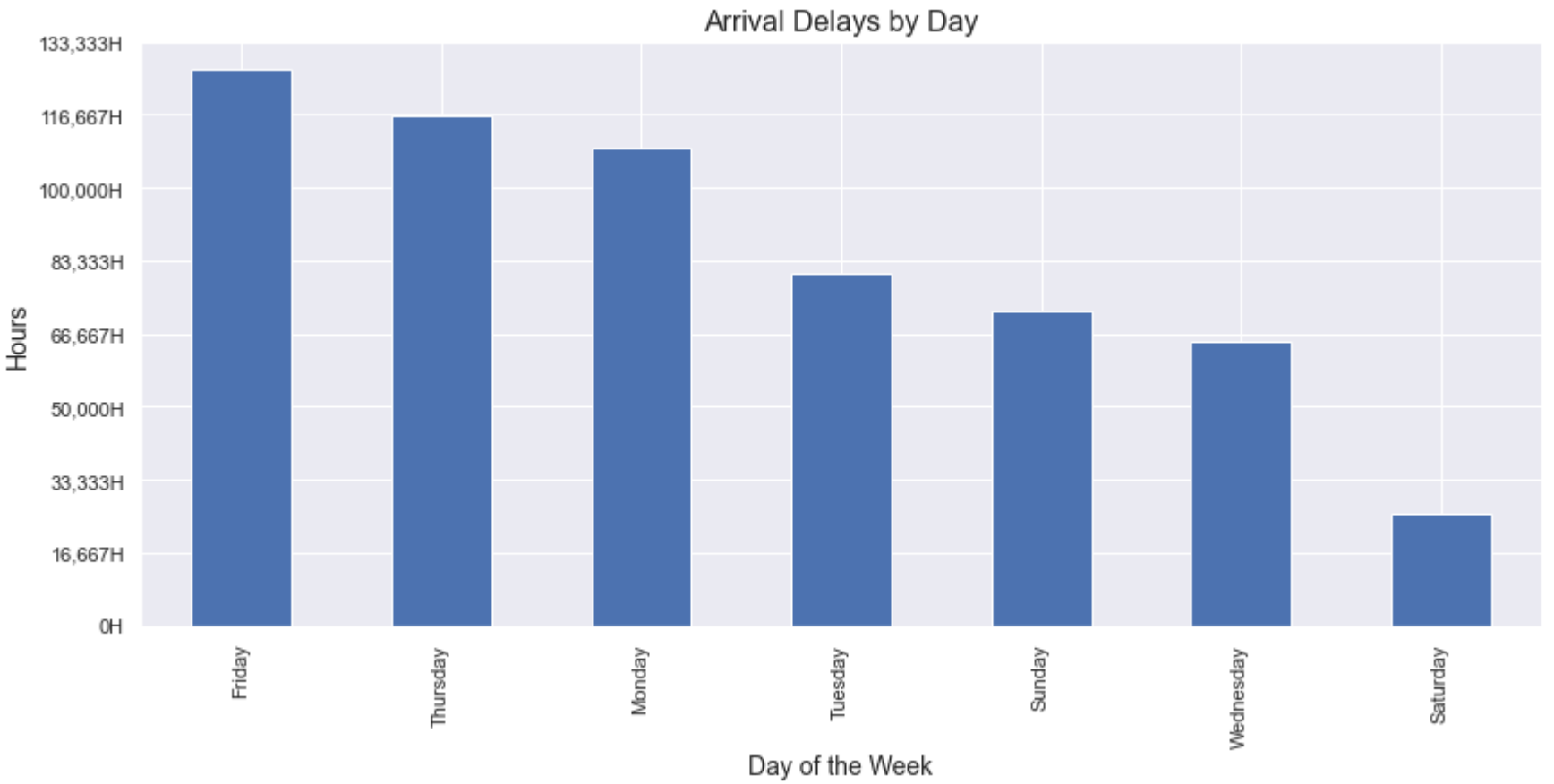
In [52]:
```
#Delays by month
fig5, ax5 = plt.subplots(figsize=(14, 6))

data.groupby('month').ARR_DELAY.sum().sort_values(ascending=False).plot.bar()
plt.title('Arrival Delays by Month', fontsize=16)
plt.ylabel('Hours', fontsize=14)
plt.xlabel('Month of the year', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax5.get_yticks().tolist()
ax5.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax5.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])

plt.show()
```



Arrival Delays by Day

In [ ]:

In [53]:
```
#Delays by month
fig6, ax6 = plt.subplots(figsize=(14, 6))

data.groupby('weekday').ARR_DELAY.sum().sort_values(ascending=False).plot.bar()
plt.title('Arrival Delays by Day', fontsize=16)
plt.ylabel('Hours', fontsize=14)
plt.xlabel('Day of the Week', fontsize=14)
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax6.get_yticks().tolist()
ax6.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax6.set_yticklabels(['{:,.0f}'.format(y/60) + 'H' for y in ticks_loc])

plt.show()
```



In [ ]:

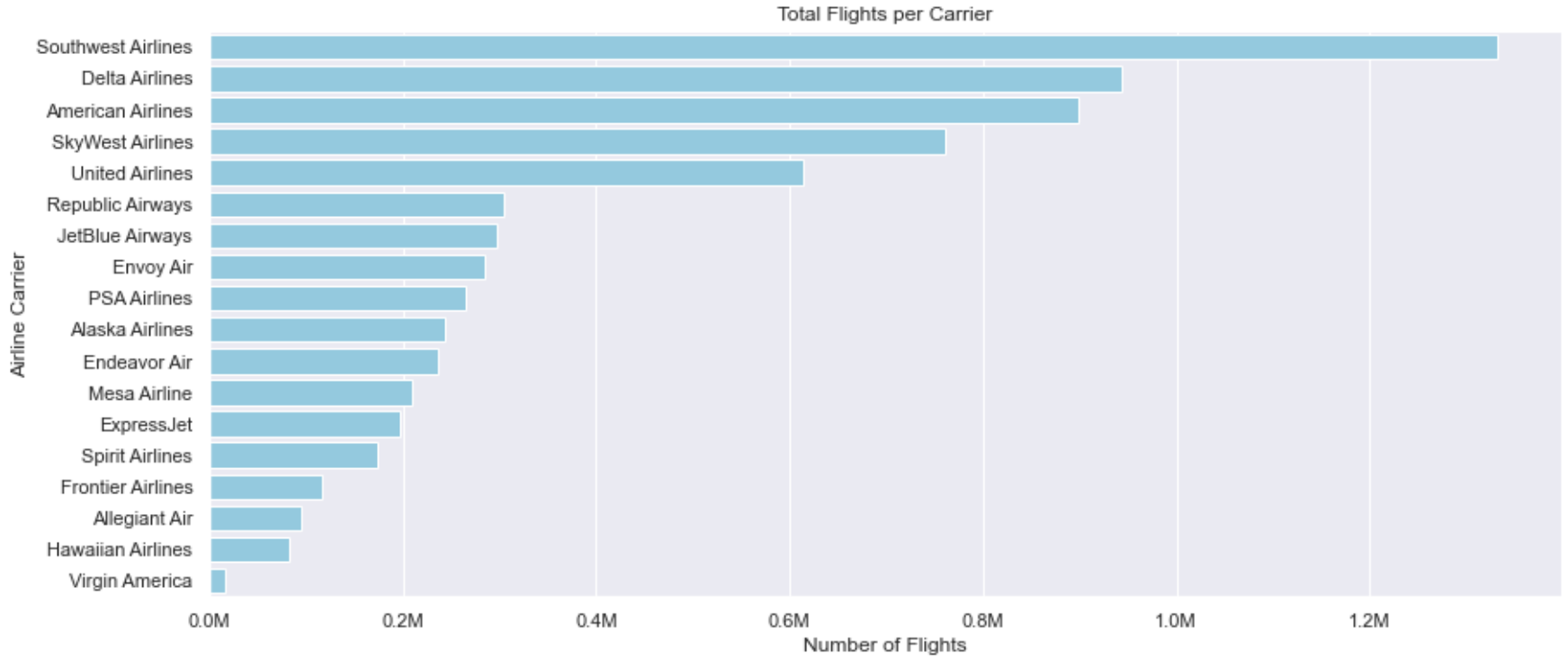
In [ ]:

In [54]:
```
#Most Frequent Origins and Dests
```

Largest Airlines by Flights

In [55]:
```
#Largest Airlines by Flights
#Delays by month

fig7, ax7 = plt.subplots(figsize=(14,6),nrows=1,ncols=1, )
flightsbyCar =pd.DataFrame(data['OP_CARRIER'].value_counts().items())
```

```
ax7 = sns.barplot(data=flightsbyCar, x=1, y=0, ax=ax7, color = 'skyblue')
ax7.set(ylabel = 'Airline Carrier', xlabel='Number of Flights', title='Total Flights per Carrier')
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax7.get_xticks().tolist()
ax7.xaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax7.set_xticklabels(['{:,.1f}'.format(x/1000000) + 'M' for x in ticks_loc]);
```



In [56]:
```
data['OP_CARRIER'].value_counts()
```

Out[56]:
```
Southwest Airlines    1331609
Delta Airlines         943837
American Airlines      899527
SkyWest Airlines       760796
United Airlines        614915
Republic Airways       305251
JetBlue Airways        297758
Envoy Air              284497
PSA Airlines           265784
Alaska Airlines        242979
Endeavor Air           236403
Mesa Airline           209094
```

```
ExpressJet             196583
Spirit Airlines        174129
Frontier Airlines      117523
Allegiant Air           95192
Hawaiian Airlines       83375
Virgin America          17153
Name: OP_CARRIER, dtype: int64
```

In [57]:
```
data['OP_CARRIER'].value_counts(normalize=True)
```

Out[57]:
```
Southwest Airlines    0.188176
Delta Airlines        0.133378
American Airlines     0.127116
SkyWest Airlines      0.107512
United Airlines       0.086897
Republic Airways      0.043136
JetBlue Airways       0.042078
Envoy Air             0.040204
PSA Airlines          0.037559
Alaska Airlines       0.034337
Endeavor Air          0.033407
Mesa Airline          0.029548
ExpressJet            0.027780
Spirit Airlines       0.024607
Frontier Airlines     0.016608
Allegiant Air         0.013452
Hawaiian Airlines     0.011782
Virgin America        0.002424
Name: OP_CARRIER, dtype: float64
```

In [58]:
```
.188176 + 0.133378 + 0.127116 + 0.107512 +0.086897
```

Out[58]:
```
0.643079
```

In [ ]:

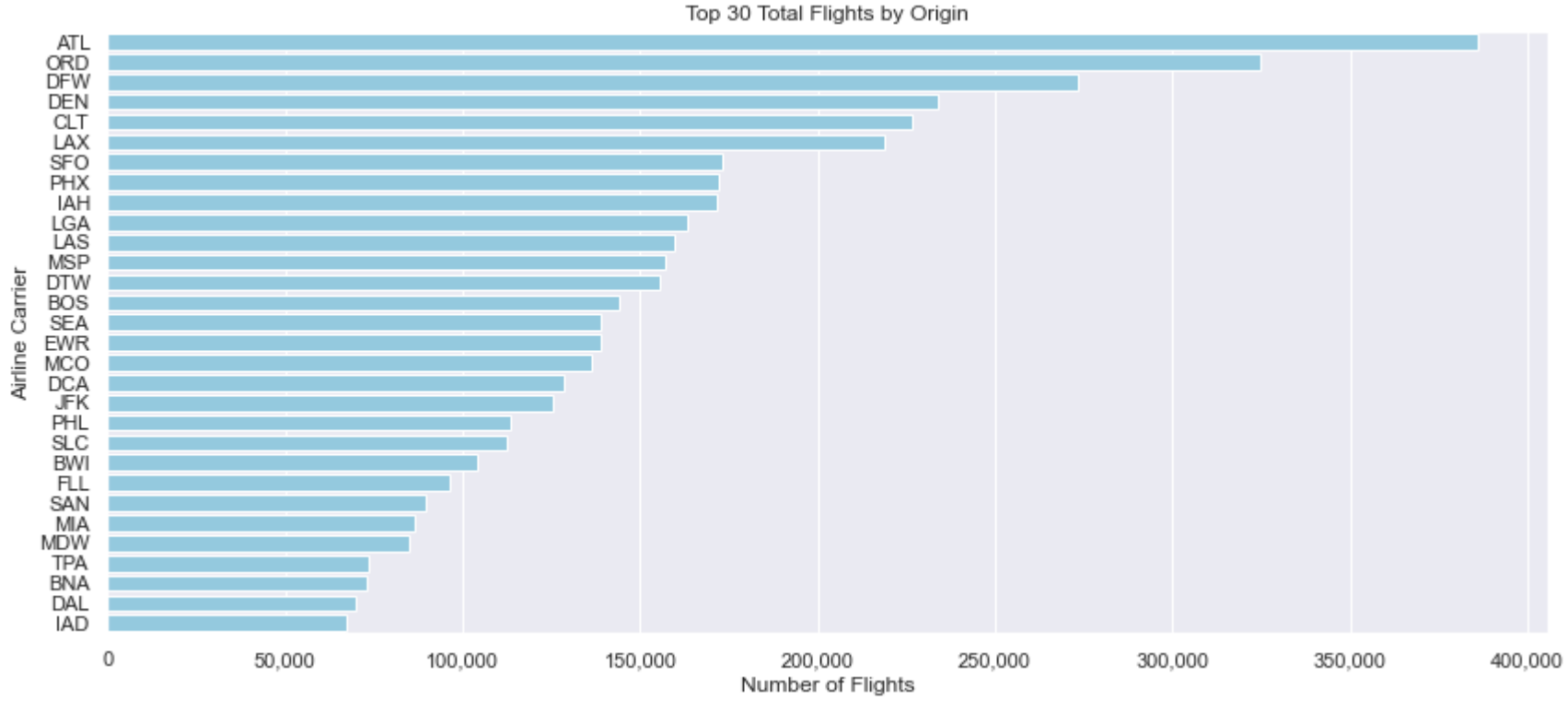
In [ ]:

Busiest Cities by Flights

In [59]:
```
flightsbyOrigin =pd.DataFrame(data['ORIGIN'].value_counts().items())
```

In [60]:
```
fig9, ax9 = plt.subplots(figsize=(14,6),nrows=1,ncols=1, )
ax9 = sns.barplot(data=flightsbyOrigin[:30], x=1, y=0, ax=ax9, color = 'skyblue')
ax9.set(ylabel = 'Airline Carrier', xlabel='Number of Flights', title='Top 30 Total Flights by Origin')
```

```
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax9.get_xticks().tolist()
ax9.xaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax9.set_xticklabels(['{:,.0f}'.format(x)  for x in ticks_loc]);
```



In [61]:
```
flightsbyOrigin
```

Out[61]:

	0	1
0	ATL	386179
1	ORD	324893
2	DFW	273214
3	DEN	233741
4	CLT	226773
...	...	...
353	AKN	63

	0	1
354	CYS	57
355	IFP	45
356	ART	24
357	YNG	2

358 rows × 2 columns

In [62]:
```
#create top 30 cities by Origin by Flights
originCities = flightsbyOrigin[:30][0]
```

In [63]:
```
flightsbyOrigin[:30][1].sum()
```

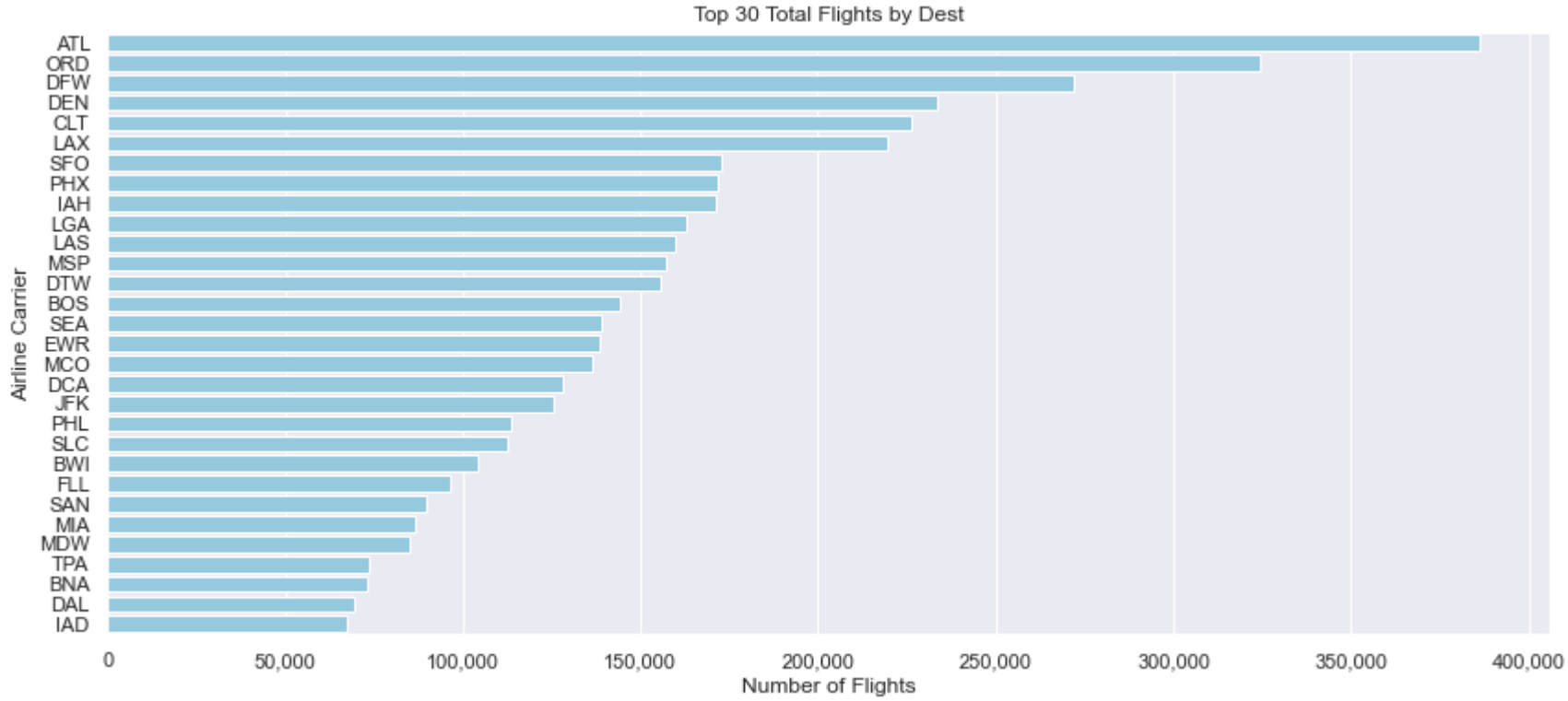
Out[63]: 4599833

In [64]:
```
4599833/7076405
```

Out[64]: 0.6500239881691339

In [65]:
```
flightsbyDest =pd.DataFrame(data['DEST'].value_counts().items())
```

In [66]:
```
fig10, ax10 = plt.subplots(figsize=(14,6),nrows=1,ncols=1, )
ax10 = sns.barplot(data=flightsbyDest[:30], x=1, y=0, ax=ax10, color = 'skyblue')
ax10.set(ylabel = 'Airline Carrier', xlabel='Number of Flights', title='Top 30 Total Flights by Dest')
# fixing xticks warning with matplotlib.ticker "FixedLocator"
ticks_loc = ax10.get_xticks().tolist()
ax10.xaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax10.set_xticklabels(['{:,.0f}'.format(x)  for x in ticks_loc]);
```



In [ ]:

In [ ]:

In [67]:
```
flightsbyDest
```

Out[67]:

	0	1
0	ATL	386390
1	ORD	324260
2	DFW	272147
3	DEN	233411
4	CLT	226110
...	...	...
353	AKN	63

	0	1
354	CYS	57
355	IFP	45
356	ART	25
357	YNG	2

358 rows × 2 columns

In [68]:
```
#create top 30 cities by Dest by Flights
destCities = flightsbyDest[:30][0]
```

In [69]:
```
flightsbyDest[:30][1].sum()
```

Out[69]: 4596482

In [70]:
```
4596482/7076405
```

Out[70]: 0.6495504426329471

In [ ]:
```

```

Model Building

Reducing Features

Only Focus on top 5 Airlines which consist of 65% of Flights

Southwest Airlines 0.188176 Delta Airlines 0.133378 American Airlines 0.127116 SkyWest Airlines 0.107512 United Airlines 0.086897

In [71]:
```
options = ['Southwest Airlines', 'Delta Airlines', 'American Airlines', 'SkyWest Airlines' , 'United Airlines']

# selecting rows based on condition
air5_df = data[data['OP_CARRIER'].isin(options)]
```

In [72]:
```
#resulting 4550684 entries
```

```
air5_df.reset_index(drop=True, inplace=True)
```

In [73]:
```
#Select only records with dest or origin city in top 30
#destCities selecting rows based on condition
air5_df = air5_df[air5_df['DEST'].isin(destCities)]
#resulting on 3145902 flights
```

In [74]:
```
#Select only records with dest or origin city in top 30
#destCities# selecting rows based on condition
air5_df = air5_df[air5_df['ORIGIN'].isin(originCities)]
#resulting on 1943580 flights
```

In [75]:
```
air5_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1943580 entries, 0 to 4550680
Data columns (total 30 columns):
 #   Column               Dtype  
---  ------               -----  
 0   FL_DATE              object 
 1   OP_CARRIER           object 
 2   OP_CARRIER_FL_NUM    int64  
 3   ORIGIN               object 
 4   DEST                 object 
 5   CRS_DEP_TIME         int64  
 6   DEP_TIME             float64
 7   DEP_DELAY            float64
 8   TAXI_OUT             float64
 9   WHEELS_OFF           float64
 10  WHEELS_ON            float64
 11  TAXI_IN              float64
 12  CRS_ARR_TIME         int64  
 13  ARR_TIME             float64
 14  ARR_DELAY            float64
 15  DIVERTED             float64
 16  CRS_ELAPSED_TIME     float64
 17  ACTUAL_ELAPSED_TIME  float64
 18  AIR_TIME             float64
 19  DISTANCE             float64
 20  CARRIER_DELAY        float64
 21  WEATHER_DELAY        float64
 22  NAS_DELAY            float64
 23  SECURITY_DELAY       float64
 24  LATE_AIRCRAFT_DELAY  float64
 25  Unnamed: 27          float64
 26  month                object 
 27  day                  int64  
```

```
 28  weekday              object
 29  DELAYED              int32
dtypes: float64(19), int32(1), int64(4), object(6)
memory usage: 452.3+ MB
```

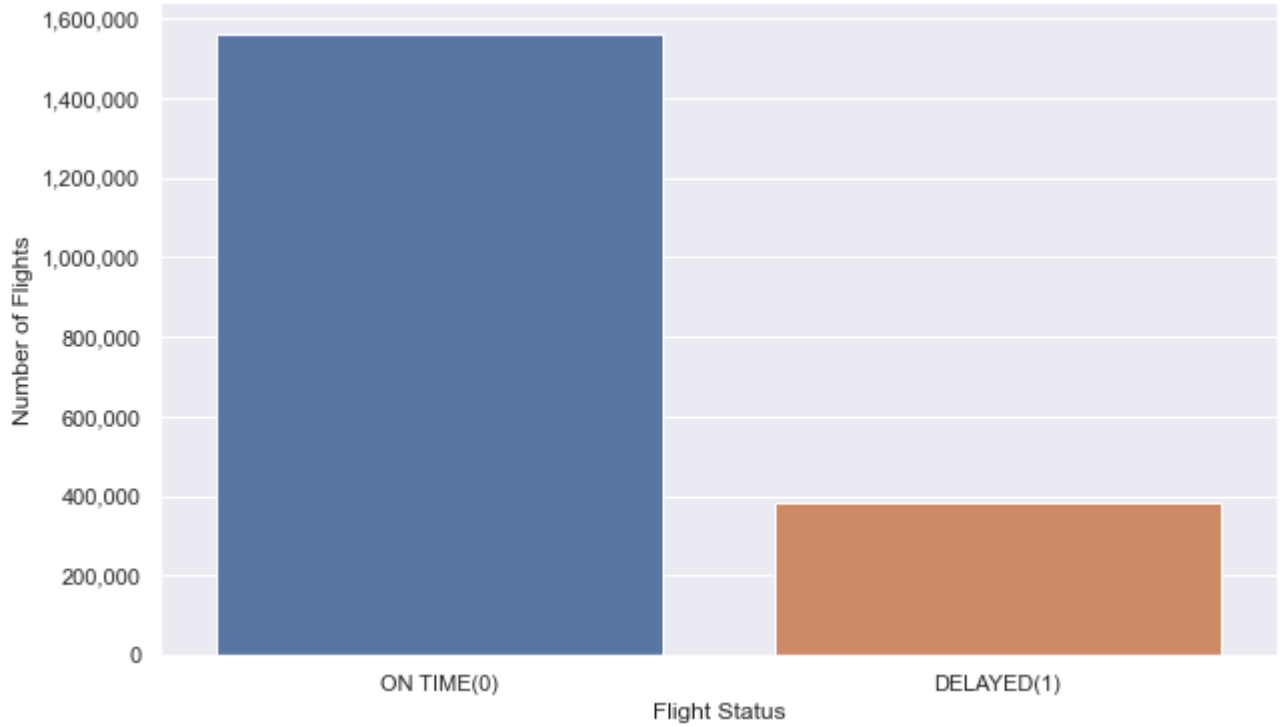
In [136…

```
def balance(data):

    counts = air5_df.DELAYED.value_counts() # 'data' is our input which will be any of the 3 dataframes created
    print('Legend:')
    print(counts)

    fig20, ax20 = plt.subplots(figsize=(10,6),nrows=1,ncols=1,)

    ax20 = sns.barplot(x=counts.index, y=counts.values, ax=ax20)

    plt.rcParams["figure.facecolor"] = "white"
    ax20.set(xlabel='Flight Status',ylabel='Number of Flights')

    plt.xticks(range(len(counts.index)), ['ON TIME(0)', 'DELAYED(1)'])
    # fixing xticks warning with matplotlib.ticker "FixedLocator"
    ticks_loc = ax20.get_yticks().tolist()
    ax20.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
    ax20.set_yticklabels(['{:,.0f}'.format(x)  for x in ticks_loc]);
    ax20.
    plt.show()
```

In [137…

```
balance(air5_df)
```

```
Legend:
0    1562024
1     381556
Name: DELAYED, dtype: int64
```



In [78]:
```
air5_df.DELAYED.value_counts()
```

Out[78]:
```
0    1562024
1     381556
Name: DELAYED, dtype: int64
```

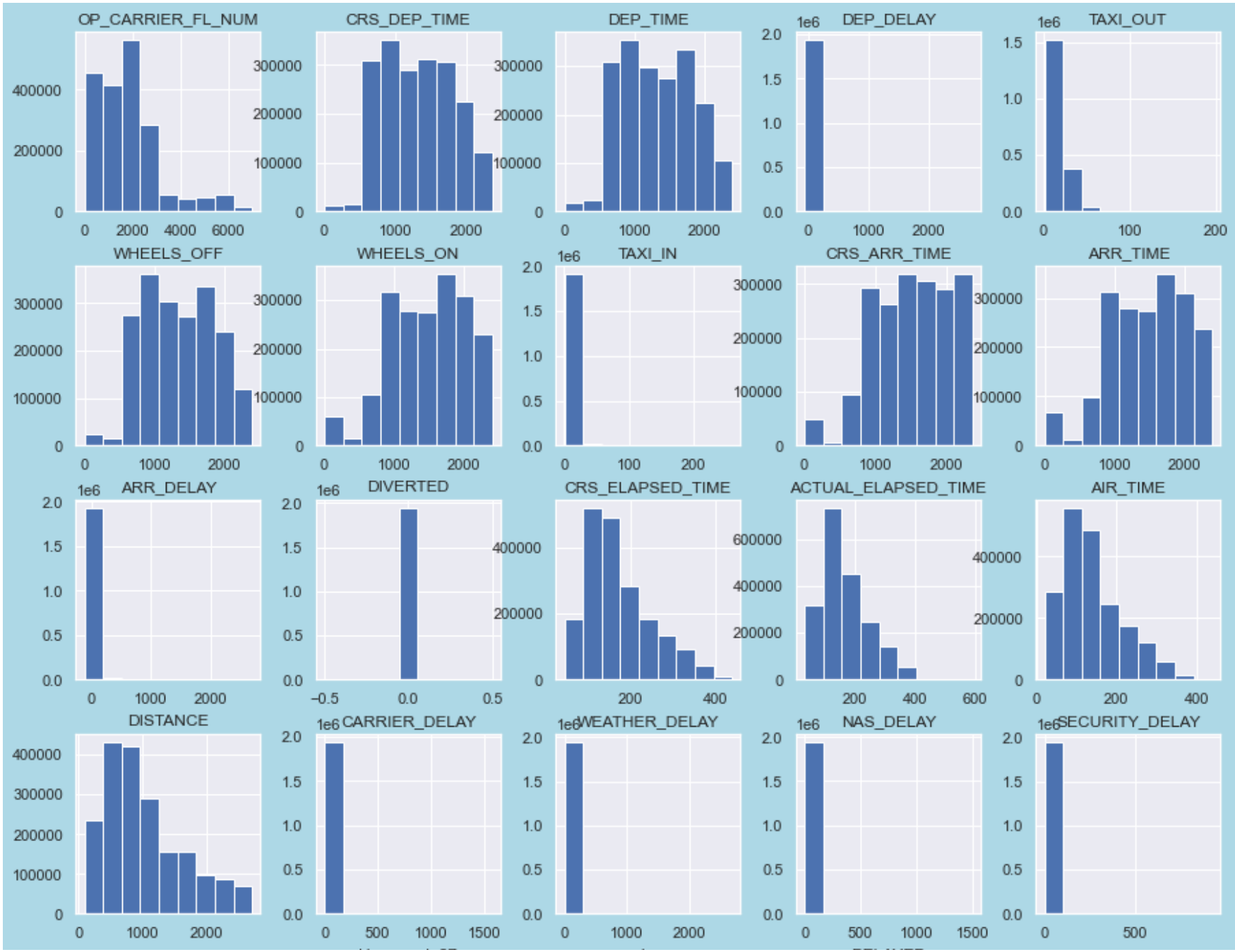
In [79]:
```
air5_df.DELAYED.value_counts(normalize=True)
```

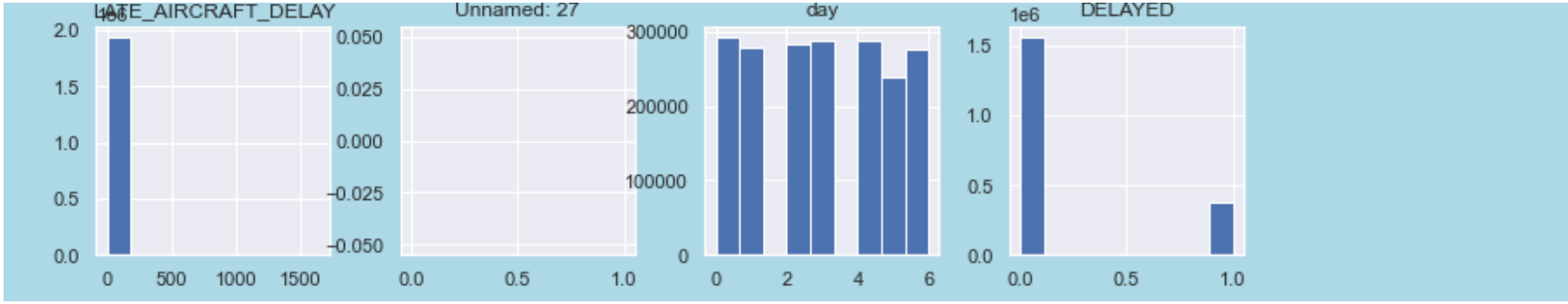
Out[79]:
```
0    0.803684
1    0.196316
Name: DELAYED, dtype: float64
```

In [80]:
```
air5_df.columns
```

Out[80]:
```
Index(['FL_DATE', 'OP_CARRIER', 'OP_CARRIER_FL_NUM', 'ORIGIN', 'DEST',
       'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY', 'TAXI_OUT', 'WHEELS_OFF',
       'WHEELS_ON', 'TAXI_IN', 'CRS_ARR_TIME', 'ARR_TIME', 'ARR_DELAY',
       'DIVERTED', 'CRS_ELAPSED_TIME', 'ACTUAL_ELAPSED_TIME', 'AIR_TIME',
       'DISTANCE', 'CARRIER_DELAY', 'WEATHER_DELAY', 'NAS_DELAY',
       'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY', 'Unnamed: 27', 'month', 'day',
       'weekday', 'DELAYED'],
      dtype='object')
```

In [81]:
```
#Histogram of all Columns
air5_df.hist(figsize  = [15, 15],bins=9)
plt.show()
```





In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [82]:
```
import statsmodels.api as sm
import sklearn.preprocessing as preprocessing
from scipy import stats
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier

from sklearn.metrics import accuracy_score, roc_curve, auc, roc_auc_score
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.svm import SVC
```

In [83]:
```
x_feats = ['OP_CARRIER', 'ORIGIN', 'DEST','month','weekday']
X_dummies = pd.get_dummies(air5_df[x_feats], drop_first=True, dtype=float)
```

In [84]:
```
X_dummies
```

Out[84]:

	OP_CARRIER_Delta Airlines	OP_CARRIER_SkyWest Airlines	OP_CARRIER_Southwest Airlines	OP_CARRIER_United Airlines	ORIGIN_BNA	ORIGIN_BOS	ORIGIN_BWI	ORI(
0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
6	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
13	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
15	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	
4550676	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4550677	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4550678	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4550679	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4550680	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

1943580 rows × 79 columns

In [85]:
```
wdum_df = pd.concat([air5_df, X_dummies], axis=1)
```

In [86]:
```
wdum_df.columns
```

Out[86]:
```
Index(['FL_DATE', 'OP_CARRIER', 'OP_CARRIER_FL_NUM', 'ORIGIN', 'DEST',
       'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY', 'TAXI_OUT', 'WHEELS_OFF',
       ...
       'month_May', 'month_November', 'month_October', 'month_September',
       'weekday_Monday', 'weekday_Saturday', 'weekday_Sunday',
       'weekday_Thursday', 'weekday_Tuesday', 'weekday_Wednesday'],
      dtype='object', length=109)
```

In [ ]:

In [ ]:

In [87]:
```
model_df = wdum_df.drop(columns=['OP_CARRIER','FL_DATE','ORIGIN','OP_CARRIER_FL_NUM','DEST','CRS_DEP_TIME', 'DEP_TIME',
                      'DEP_DELAY', 'TAXI_OUT', 'WHEELS_OFF','WHEELS_ON','TAXI_IN',
```

```
                       'CRS_ARR_TIME', 'ARR_TIME','ARR_DELAY','DIVERTED', 'CRS_ELAPSED_TIME',
                      'ACTUAL_ELAPSED_TIME','CARRIER_DELAY','WEATHER_DELAY','NAS_DELAY',
                      'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY','Unnamed: 27','day', 'month' ,'weekday'], axis = 1)
```

In [88]:
```
model_df.reset_index(drop=True, inplace=True)
```

In [89]:
```
model_df
```

Out[89]:

	AIR_TIME	DISTANCE	DELAYED	OP_CARRIER_Delta Airlines	OP_CARRIER_SkyWest Airlines	OP_CARRIER_Southwest Airlines	OP_CARRIER_United Airlines	ORIGIN_BNA
0	225.0	1605.0	0	0.0	0.0	0.0	1.0	0.0
1	65.0	414.0	0	0.0	0.0	0.0	1.0	0.0
2	173.0	1222.0	0	0.0	0.0	0.0	1.0	0.0
3	124.0	746.0	0	0.0	0.0	0.0	1.0	0.0
4	302.0	2218.0	1	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...
1943575	45.0	226.0	0	0.0	0.0	0.0	0.0	0.0
1943576	142.0	1303.0	0	0.0	0.0	0.0	0.0	0.0
1943577	46.0	331.0	0	0.0	0.0	0.0	0.0	0.0
1943578	72.0	331.0	0	0.0	0.0	0.0	0.0	0.0
1943579	148.0	936.0	0	0.0	0.0	0.0	0.0	0.0

1943580 rows × 82 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Write model data

In [90]:
```
model_df.to_csv('./data/2018_model_data.csv',encoding='utf-8')
```

Build Baseline Model

In [87]:
```
model_df.columns
```

Out[87]:
```
Index(['AIR_TIME', 'DISTANCE', 'DELAYED', 'OP_CARRIER_Delta Airlines',
       'OP_CARRIER_SkyWest Airlines', 'OP_CARRIER_Southwest Airlines',
       'OP_CARRIER_United Airlines', 'ORIGIN_BNA', 'ORIGIN_BOS', 'ORIGIN_BWI',
       'ORIGIN_CLT', 'ORIGIN_DAL', 'ORIGIN_DCA', 'ORIGIN_DEN', 'ORIGIN_DFW',
       'ORIGIN_DTW', 'ORIGIN_EWR', 'ORIGIN_FLL', 'ORIGIN_IAD', 'ORIGIN_IAH',
       'ORIGIN_JFK', 'ORIGIN_LAS', 'ORIGIN_LAX', 'ORIGIN_LGA', 'ORIGIN_MCO',
       'ORIGIN_MDW', 'ORIGIN_MIA', 'ORIGIN_MSP', 'ORIGIN_ORD', 'ORIGIN_PHL',
       'ORIGIN_PHX', 'ORIGIN_SAN', 'ORIGIN_SEA', 'ORIGIN_SFO', 'ORIGIN_SLC',
       'ORIGIN_TPA', 'DEST_BNA', 'DEST_BOS', 'DEST_BWI', 'DEST_CLT',
       'DEST_DAL', 'DEST_DCA', 'DEST_DEN', 'DEST_DFW', 'DEST_DTW', 'DEST_EWR',
       'DEST_FLL', 'DEST_IAD', 'DEST_IAH', 'DEST_JFK', 'DEST_LAS', 'DEST_LAX',
       'DEST_LGA', 'DEST_MCO', 'DEST_MDW', 'DEST_MIA', 'DEST_MSP', 'DEST_ORD',
       'DEST_PHL', 'DEST_PHX', 'DEST_SAN', 'DEST_SEA', 'DEST_SFO', 'DEST_SLC',
       'DEST_TPA', 'month_August', 'month_December', 'month_February',
       'month_January', 'month_July', 'month_June', 'month_March', 'month_May',
       'month_November', 'month_October', 'month_September', 'weekday_Monday',
       'weekday_Saturday', 'weekday_Sunday', 'weekday_Thursday',
       'weekday_Tuesday', 'weekday_Wednesday'],
      dtype='object')
```

In [ ]:

In [ ]:

In [ ]:

In [199…
```
# Split the dataset in the ratio train 80% and test 20%
X = model_df.drop("DELAYED", axis = 1)
y = model_df['DELAYED']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)
```

In [200…
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.transform(X_test)
```

In [89]:
```
X_train
```

Out[89]:

	AIR_TIME	DISTANCE	OP_CARRIER_Delta Airlines	OP_CARRIER_SkyWest Airlines	OP_CARRIER_Southwest Airlines	OP_CARRIER_United Airlines	ORIGIN_BNA	ORIGIN_B
1200163	98.0	731.0	1.0	0.0	0.0	0.0	0.0	
1332715	97.0	599.0	0.0	1.0	0.0	0.0	0.0	
834130	216.0	1744.0	0.0	0.0	0.0	1.0	0.0	
1570432	71.0	349.0	0.0	0.0	1.0	0.0	0.0	
1434726	76.0	547.0	1.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	
259178	87.0	591.0	1.0	0.0	0.0	0.0	0.0	
1414414	83.0	594.0	1.0	0.0	0.0	0.0	0.0	
131932	181.0	1379.0	0.0	0.0	0.0	1.0	0.0	
671155	262.0	2182.0	1.0	0.0	0.0	0.0	0.0	
121958	139.0	1065.0	0.0	0.0	0.0	1.0	0.0	

1554864 rows × 81 columns

In [ ]:
```
```

In [202…
```
# Build a regular tree as baseline
tree_clf1 = DecisionTreeClassifier(class_weight='balanced')
tree_clf1.fit(X_train, y_train)
```

Out[202…
```
DecisionTreeClassifier(class_weight='balanced')
```

In [203…
```
# Make predictions for test data
y_pred1 = tree_clf1.predict(X_test)
```

In [204…
```
print(classification_report(y_test,y_pred1))
```

```
                precision    recall  f1-score   support

           0       0.83      0.75      0.79    312133
           1       0.27      0.39      0.32     76583

    accuracy                           0.68    388716
   macro avg       0.55      0.57      0.55    388716
weighted avg       0.72      0.68      0.69    388716
```

In [205…
```
# Checking results on train data
y_pred1_train = tree_clf1.predict(X_train_scaled)
```

In [ ]:
```
print(classification_report(y_train,y_pred1_train))
```

In [ ]:
```

```

In [ ]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.transform(X_test)
```

In [ ]:
```
# Build a regular tree as baseline
tree_clf2 = DecisionTreeClassifier(class_weight='balanced')
tree_clf2.fit(X_train_scaled, y_train)
```

In [ ]:
```

```

In [ ]:
```
tree_clf2.feature_importances_
```

In [ ]:
```
sns.barplot(y=X.columns, x=tree_clf2.feature_importances_)
```

In [ ]:
```
# Make predictions for test data
y_pred2 = tree_clf2.predict(X_test)
```

In [ ]:
```
X_test
```

In [ ]:
```
print(classification_report(y_test,y_pred2))
```

In [ ]:
```
# Make predictions for test data
y_pred2_train = tree_clf2.predict(X_train_scaled)
```

In [ ]:
```
print(classification_report(y_train,y_pred2_train))
```

In [ ]:
```

```

In [ ]:
```
# Calculate accuracy
acc = accuracy_score(y_test,y_pred1) * 100
print('Accuracy is :{0}'.format(acc))

# Check the AUC for predictions
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred1)
roc_auc = auc(false_positive_rate, true_positive_rate)
print('\nAUC is :{0}'.format(round(roc_auc, 2)))

# Create and print a confusion matrix
print('\nConfusion Matrix')
print('----------------')
pd.crosstab(y_test, y_pred1, rownames=['True'], colnames=['Predicted'], margins=True)
```

In [ ]:
```
# Alternative confusion matrix
from sklearn.metrics import plot_confusion_matrix

plot_confusion_matrix(tree_clf1, X, y, values_format='.3g')
plt.show()
```

In [ ]:
```
tree_clf2.feature_importances_
```

In [ ]:
```
# Make predictions for test data
y_pred2 = tree_clf2.predict(X_test)
```

In [ ]:
```
# Calculate accuracy
acc = accuracy_score(y_test,y_pred2) * 100
print('Accuracy is :{0}'.format(acc))

# Check the AUC for predictions
false_positive_rate, true_positive_rate, thresholds = roc_curve(y_test, y_pred2)
roc_auc = auc(false_positive_rate, true_positive_rate)
print('\nAUC is :{0}'.format(round(roc_auc, 2)))

# Create and print a confusion matrix
print('\nConfusion Matrix')
print('----------------')
pd.crosstab(y_test, y_pred2, rownames=['True'], colnames=['Predicted'], margins=True)
```

In [ ]:

In [ ]:

In [ ]:
```
classification_models = []
classification_models.append(('Kernel SVM', SVC(kernel = 'rbf',gamma='scale')))
classification_models.append(('Decision Tree', DecisionTreeClassifier(criterion = "entropy")))
classification_models.append(('Random Forest', RandomForestClassifier(n_estimators=100, criterion="entropy")))

for name, model in classification_models:
  kfold = KFold(n_splits=10, random_state=7)
  result = cross_val_score(model, X, y, cv=kfold, scoring='accuracy')
  print("%s: Mean Accuracy = %.2f%% - SD Accuracy = %.2f%%" % (name, result.mean()*100, result.std()*100))
```

In [ ]:

In [ ]:

In [ ]:

Interpret

This was a complex data set for classification.
Modeling with continuous features Distance, Flight Time, Categoricals Weekdays, Months, Top 5 Airlines, Top 30 Origins and Destinations
To Classify if Delayed or not. Delays are on Arrival Delays and gt=15 mins

Best Predictive Results were found with the XGBoost algorithm With a Recall of 59%, Accuracy 66% , F1 value of .59

Additional examation with PCA could glean some more insight on usable features.

Model	Recall	Accuracy	F1
XGBoost	59%	66%	59%
Random Forest	59%	65%	57%
Decision Tree	39%	68%	55%

In [ ]: