United States Airlines Analysis

Problem Statement:

According to air travel consumer reports, a large proportion of consumer complaints are about frequent flight delays. Out of all the complaints received from consumers about airline services, 32% were related to cancellations, delays, or other deviations from the airlines' schedules. There are unavoidable delays that can be caused by air traffic, no passengers at the airport, weather conditions, mechanical issues, passengers coming from delayed connecting flights, security clearance, and aircraft preparation.

Objectives:

The objective of this project is to identify the factors that contribute to avoidable flight delays. You are also required to build a model to predict if the flight will be delayed.

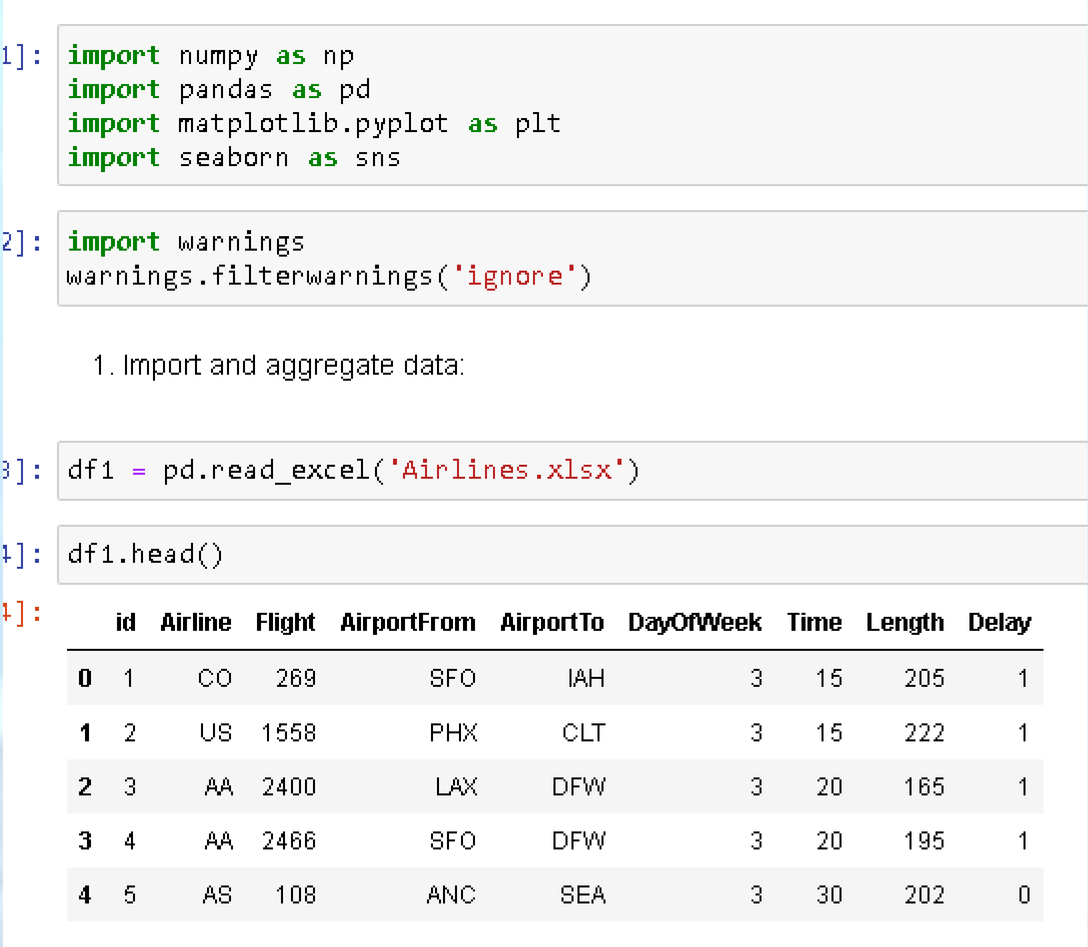
Dataset Description:

Airlines.xlsx
airports.xlsx
runways.xlsx

ANALYSIS:

Applied data science with Python:

1. Import and aggregate data:



1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2]:
```
import warnings
warnings.filterwarnings('ignore')
```

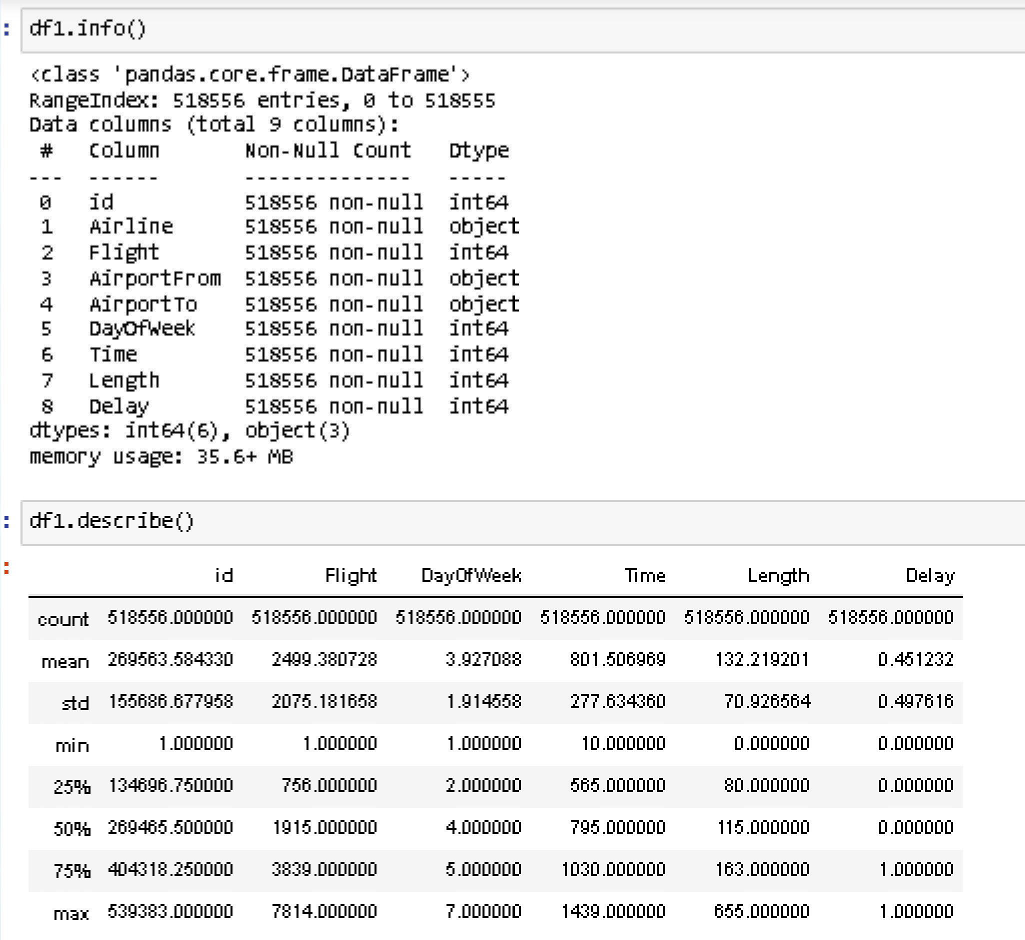
1. Import and aggregate data:

3]:
```
df1 = pd.read_excel('Airlines.xlsx')
```

4]:
```
df1.head()
```

4]:

	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay
0	1	CO	269	SFO	IAH	3	15	205	1
1	2	US	1558	PHX	CLT	3	15	222	1
2	3	AA	2400	LAX	DFW	3	20	165	1
3	4	AA	2466	SFO	DFW	3	20	195	1
4	5	AS	108	ANC	SEA	3	30	202	0

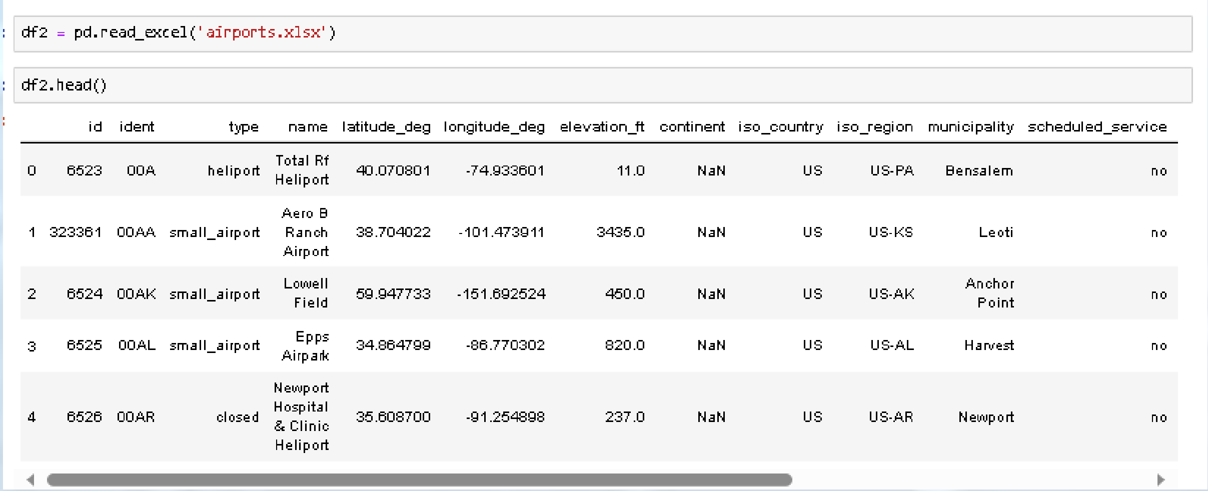


```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 518556 entries, 0 to 518555
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   id           518556 non-null  int64 
 1   Airline      518556 non-null  object
 2   Flight       518556 non-null  int64 
 3   AirportFrom  518556 non-null  object
 4   AirportTo    518556 non-null  object
 5   DayOfWeek    518556 non-null  int64 
 6   Time         518556 non-null  int64 
 7   Length       518556 non-null  int64 
 8   Delay        518556 non-null  int64 
dtypes: int64(6), object(3)
memory usage: 35.6+ MB
```

```
df1.describe()
```

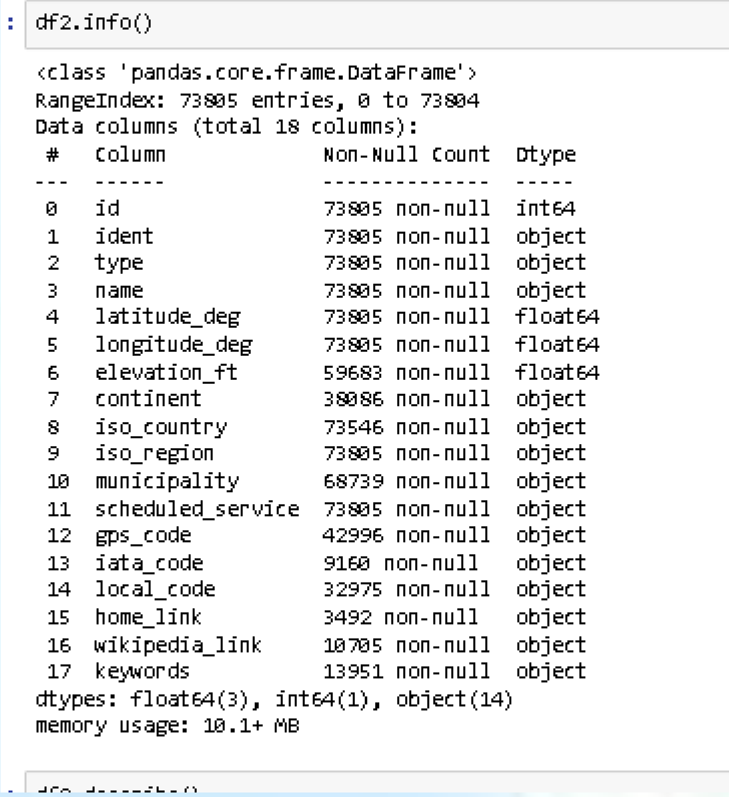
	id	Flight	DayOfWeek	Time	Length	Delay
count	518556.000000	518556.000000	518556.000000	518556.000000	518556.000000	518556.000000
mean	269563.584330	2499.380728	3.927088	801.506969	132.219201	0.451232
std	155686.677958	2075.181658	1.914558	277.634360	70.926564	0.497616
min	1.000000	1.000000	1.000000	10.000000	0.000000	0.000000
25%	134696.750000	756.000000	2.000000	565.000000	80.000000	0.000000
50%	269465.500000	1915.000000	4.000000	795.000000	115.000000	0.000000
75%	404318.250000	3839.000000	5.000000	1030.000000	163.000000	1.000000
max	539383.000000	7814.000000	7.000000	1439.000000	655.000000	1.000000



```
df2 = pd.read_excel('airports.xlsx')
```

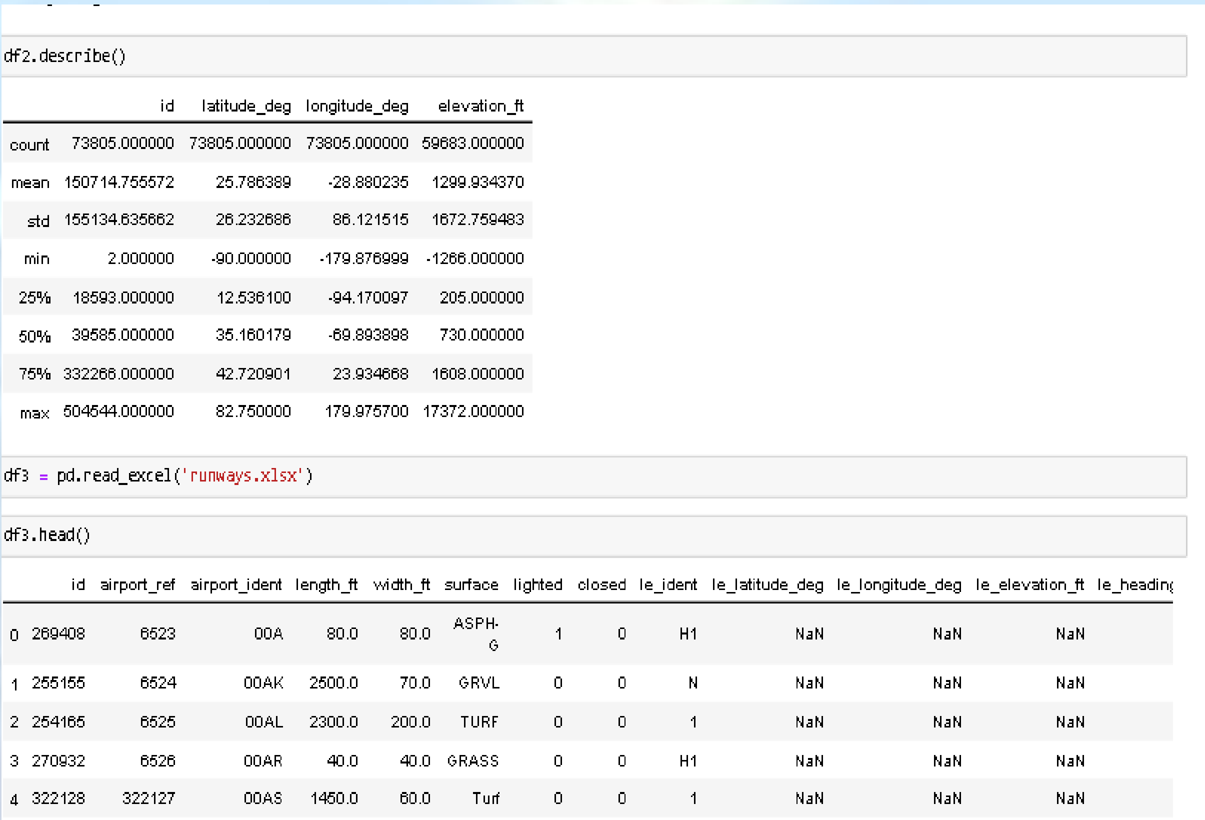
```
df2.head()
```

	id	ident	type	name	latitude_deg	longitude_deg	elevation_ft	continent	iso_country	iso_region	municipality	scheduled_service
0	6523	00A	heliport	Total Rf Heliport	40.070801	-74.933601	11.0	NaN	US	US-PA	Bensalem	no
1	323361	00AA	small_airport	Aero B Ranch Airport	38.704022	-101.473911	3435.0	NaN	US	US-KS	Leoti	no
2	6524	00AK	small_airport	Lowell Field	59.947733	-151.692524	450.0	NaN	US	US-AK	Anchor Point	no
3	6525	00AL	small_airport	Epps Airpark	34.864799	-86.770302	820.0	NaN	US	US-AL	Harvest	no
4	6526	00AR	closed	Newport Hospital & Clinic Heliport	35.608700	-91.254898	237.0	NaN	US	US-AR	Newport	no



```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73805 entries, 0 to 73804
Data columns (total 18 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 73805 non-null  int64  
 1   ident              73805 non-null  object 
 2   type               73805 non-null  object 
 3   name               73805 non-null  object 
 4   latitude_deg       73805 non-null  float64
 5   longitude_deg      73805 non-null  float64
 6   elevation_ft       59683 non-null  float64
 7   continent          38086 non-null  object 
 8   iso_country        73546 non-null  object 
 9   iso_region         73805 non-null  object 
 10  municipality       68739 non-null  object 
 11  scheduled_service  73805 non-null  object 
 12  gps_code           42996 non-null  object 
 13  iata_code          9160 non-null   object 
 14  local_code         32975 non-null  object 
 15  home_link          3492 non-null   object 
 16  wikipedia_link     10705 non-null  object 
 17  keywords           13951 non-null  object 
dtypes: float64(3), int64(1), object(14)
memory usage: 10.1+ MB
```



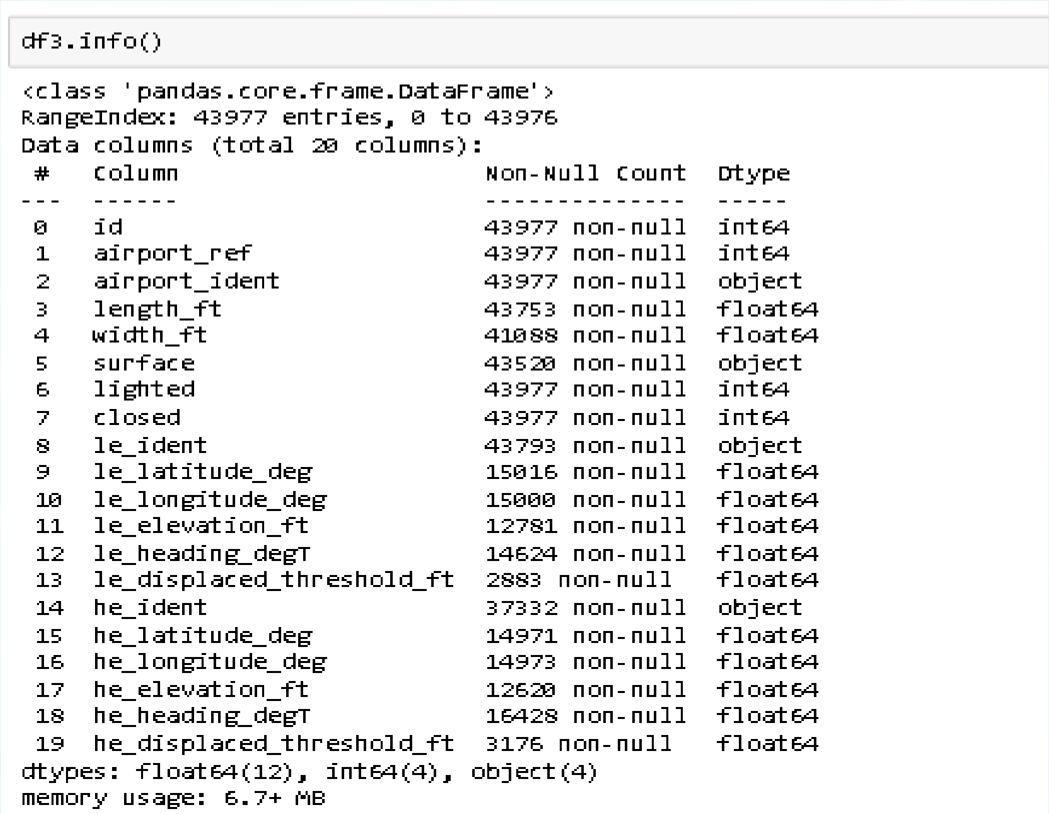
```
df2.describe()
```

	id	latitude_deg	longitude_deg	elevation_ft
count	73805.000000	73805.000000	73805.000000	59683.000000
mean	150714.755572	25.786389	-28.880235	1299.934370
std	155134.635662	26.232686	86.121515	1672.759483
min	2.000000	-90.000000	-179.876999	-1266.000000
25%	18593.000000	12.536100	-94.170097	205.000000
50%	39585.000000	35.160179	-69.893898	730.000000
75%	332266.000000	42.720901	23.934668	1608.000000
max	504544.000000	82.750000	179.975700	17372.000000

```
df3 = pd.read_excel('runways.xlsx')
```

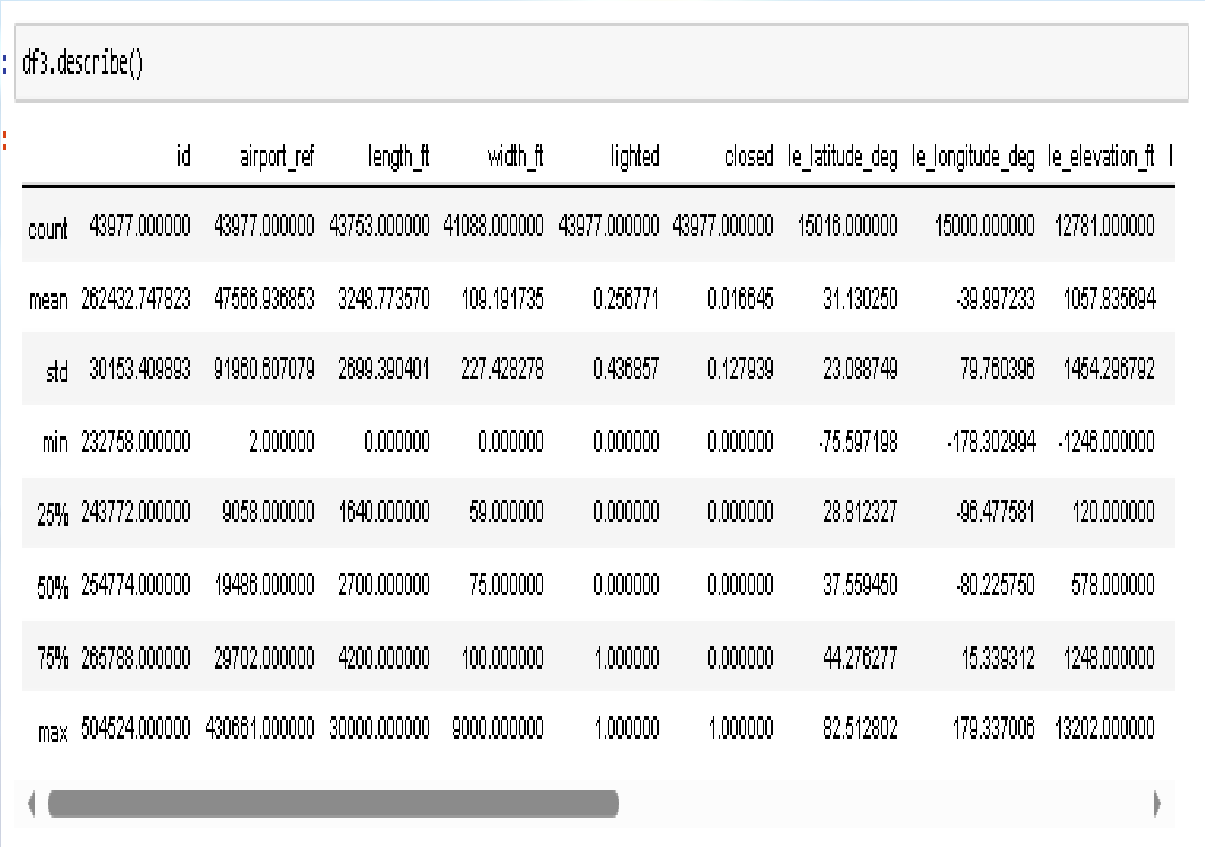
```
df3.head()
```

	id	airport_ref	airport_ident	length_ft	width_ft	surface	lighted	closed	le_ident	le_latitude_deg	le_longitude_deg	le_elevation_ft	le_heading
0	269408	6523	00A	80.0	80.0	ASPH-G	1	0	H1	NaN	NaN	NaN	
1	255155	6524	00AK	2500.0	70.0	GRVL	0	0	N	NaN	NaN	NaN	
2	254165	6525	00AL	2300.0	200.0	TURF	0	0	1	NaN	NaN	NaN	
3	270932	6526	00AR	40.0	40.0	GRASS	0	0	H1	NaN	NaN	NaN	
4	322128	322127	00AS	1450.0	60.0	Turf	0	0	1	NaN	NaN	NaN	



```
df3.info()
```

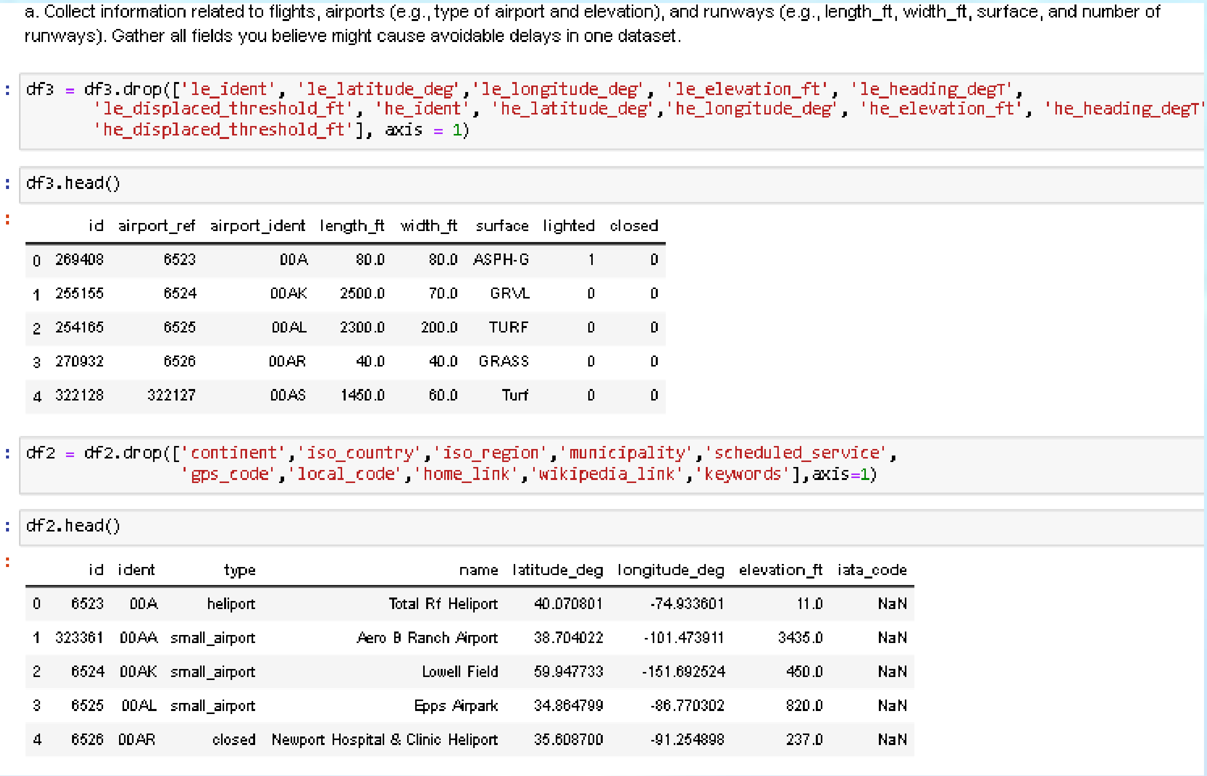
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43977 entries, 0 to 43976
Data columns (total 20 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   id                         43977 non-null  int64
 1   airport_ref                43977 non-null  int64
 2   airport_ident              43977 non-null  object
 3   length_ft                  43753 non-null  float64
 4   width_ft                   41088 non-null  float64
 5   surface                    43520 non-null  object
 6   lighted                    43977 non-null  int64
 7   closed                     43977 non-null  int64
 8   le_ident                   43793 non-null  object
 9   le_latitude_deg            15016 non-null  float64
 10  le_longitude_deg           15000 non-null  float64
 11  le_elevation_ft            12781 non-null  float64
 12  le_heading_degT            14624 non-null  float64
 13  le_displaced_threshold_ft  2883 non-null   float64
 14  he_ident                   37332 non-null  object
 15  he_latitude_deg            14971 non-null  float64
 16  he_longitude_deg           14973 non-null  float64
 17  he_elevation_ft            12620 non-null  float64
 18  he_heading_degT            16428 non-null  float64
 19  he_displaced_threshold_ft  3176 non-null   float64
dtypes: float64(12), int64(4), object(4)
memory usage: 6.7+ MB
```



```
df3.describe()
```

	id	airport_ref	length_ft	width_ft	lighted	closed	le_latitude_deg	le_longitude_deg	le_elevation_ft	l
count	43977.000000	43977.000000	43753.000000	41088.000000	43977.000000	43977.000000	15016.000000	15000.000000	12781.000000	
mean	262432.747823	47566.936853	3248.773570	109.191735	0.256771	0.016645	31.130250	-39.997233	1057.835694	
std	30153.409893	91960.607079	2699.390401	227.428278	0.436857	0.127939	23.088749	79.760396	1454.296792	
min	232758.000000	2.000000	0.000000	0.000000	0.000000	0.000000	-75.597198	-178.302994	-1246.000000	
25%	243772.000000	9058.000000	1640.000000	59.000000	0.000000	0.000000	28.812327	-96.477581	120.000000	
50%	254774.000000	19486.000000	2700.000000	75.000000	0.000000	0.000000	37.559450	-80.225750	578.000000	
75%	265788.000000	29702.000000	4200.000000	100.000000	1.000000	0.000000	44.276277	15.339312	1248.000000	
max	504524.000000	430661.000000	30000.000000	9000.000000	1.000000	1.000000	82.512802	179.337006	13202.000000	

a. Collect information related to flights, airports (e.g., type of airport and elevation), and runways (e.g., length_ft, width_ft, surface, and number of runways). Gather all fields you believe might cause avoidable delays in one dataset.



a. Collect information related to flights, airports (e.g., type of airport and elevation), and runways (e.g., length_ft, width_ft, surface, and number of runways). Gather all fields you believe might cause avoidable delays in one dataset.

```
df3 = df3.drop(['le_ident', 'le_latitude_deg','le_longitude_deg', 'le_elevation_ft', 'le_heading_degT',
       'le_displaced_threshold_ft', 'he_ident', 'he_latitude_deg','he_longitude_deg', 'he_elevation_ft', 'he_heading_degT'
       'he_displaced_threshold_ft'], axis = 1)
```

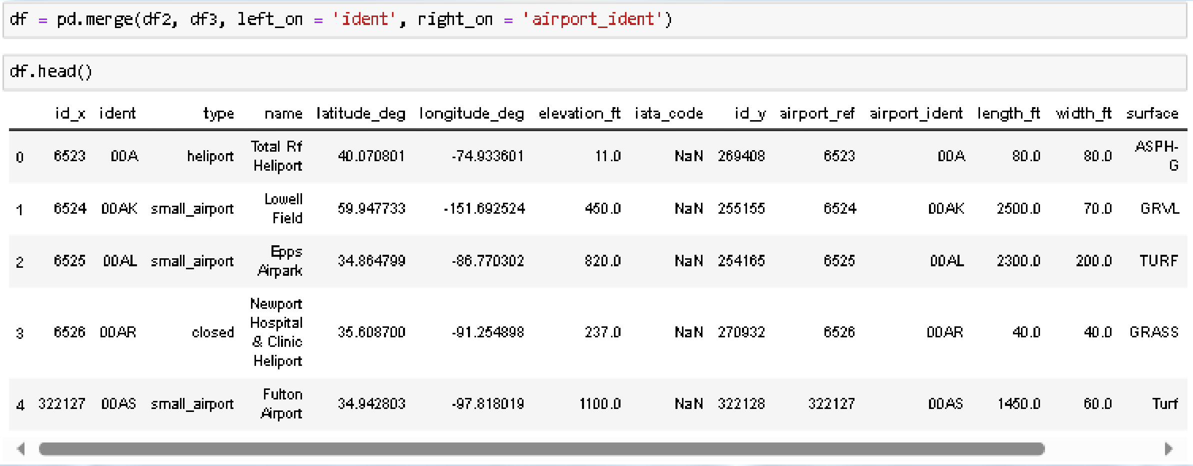
```
df3.head()
```

	id	airport_ref	airport_ident	length_ft	width_ft	surface	lighted	closed
0	269408	6523	00A	80.0	80.0	ASPH-G	1	0
1	255155	6524	00AK	2500.0	70.0	GRVL	0	0
2	254165	6525	00AL	2300.0	200.0	TURF	0	0
3	270932	6526	00AR	40.0	40.0	GRASS	0	0
4	322128	322127	00AS	1450.0	60.0	Turf	0	0

```
df2 = df2.drop(['continent','iso_country','iso_region','municipality','scheduled_service',
               'gps_code','local_code','home_link','wikipedia_link','keywords'],axis=1)
```

```
df2.head()
```

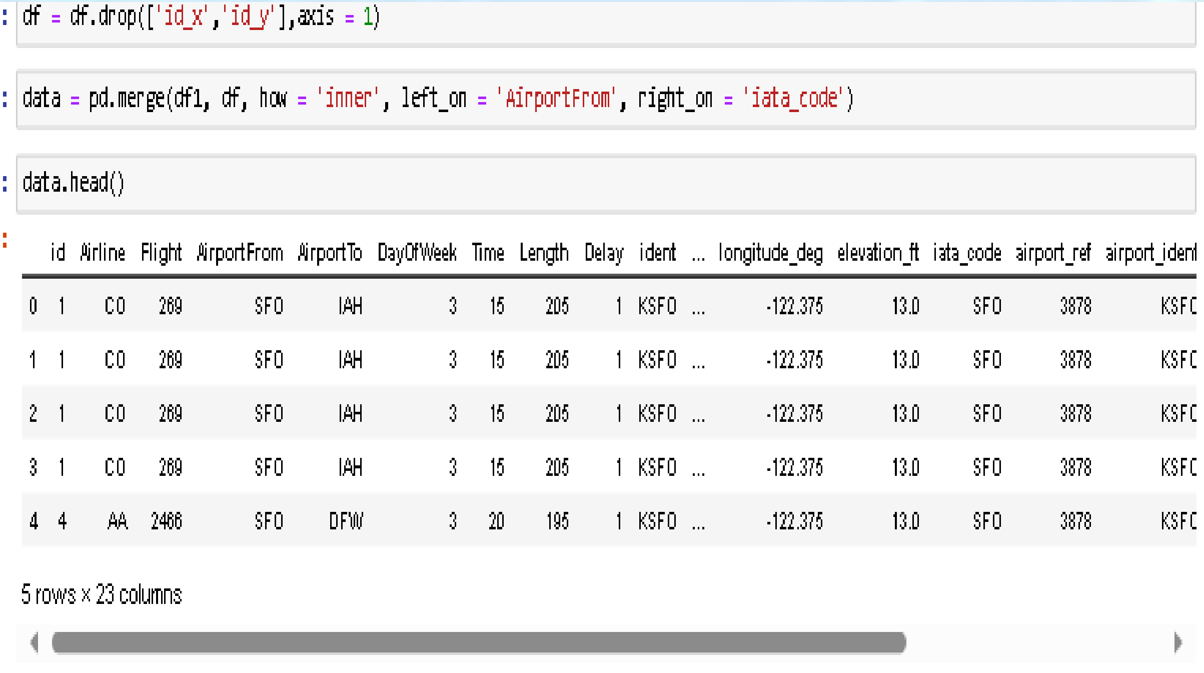
	id	ident	type	name	latitude_deg	longitude_deg	elevation_ft	iata_code
0	6523	00A	heliport	Total Rf Heliport	40.070801	-74.933601	11.0	NaN
1	323361	00AA	small_airport	Aero B Ranch Airport	38.704022	-101.473911	3435.0	NaN
2	6524	00AK	small_airport	Lowell Field	59.947733	-151.692524	450.0	NaN
3	6525	00AL	small_airport	Epps Airpark	34.864799	-86.770302	820.0	NaN
4	6526	00AR	closed	Newport Hospital & Clinic Heliport	35.608700	-91.254898	237.0	NaN



```
df = pd.merge(df2, df3, left_on = 'ident', right_on = 'airport_ident')
```

```
df.head()
```

	id_x	ident	type	name	latitude_deg	longitude_deg	elevation_ft	iata_code	id_y	airport_ref	airport_ident	length_ft	width_ft	surface
0	6523	00A	heliport	Total Rf Heliport	40.070801	-74.933601	11.0	NaN	269408	6523	00A	80.0	80.0	ASPH-G
1	6524	00AK	small_airport	Lowell Field	59.947733	-151.692524	450.0	NaN	255155	6524	00AK	2500.0	70.0	GRVL
2	6525	00AL	small_airport	Epps Airpark	34.864799	-86.770302	820.0	NaN	254165	6525	00AL	2300.0	200.0	TURF
3	6526	00AR	closed	Newport Hospital & Clinic Heliport	35.608700	-91.254898	237.0	NaN	270932	6526	00AR	40.0	40.0	GRASS
4	322127	00AS	small_airport	Fulton Airport	34.942803	-97.818019	1100.0	NaN	322128	322127	00AS	1450.0	60.0	Turf



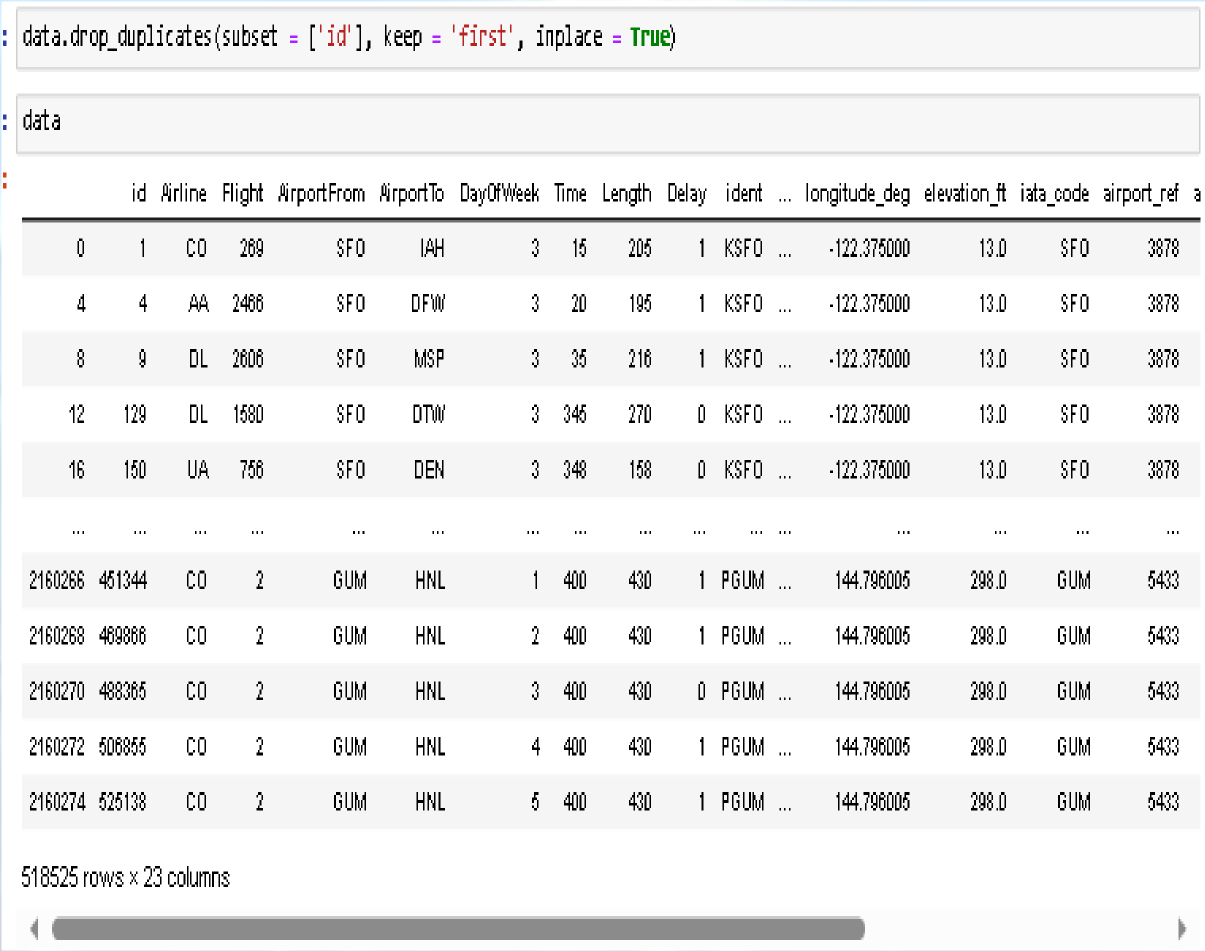
```
df = df.drop(['id_x','id_y'],axis = 1)
```

```
data = pd.merge(df1, df, how = 'inner', left_on = 'AirportFrom', right_on = 'iata_code')
```

```
data.head()
```

	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay	ident	...	longitude_deg	elevation_ft	iata_code	airport_ref	airport_ident
0	1	CO	269	SFO	IAH	3	15	205	1	KSFO	...	-122.375	13.0	SFO	3878	KSFC
1	1	CO	269	SFO	IAH	3	15	205	1	KSFO	...	-122.375	13.0	SFO	3878	KSFC
2	1	CO	269	SFO	IAH	3	15	205	1	KSFO	...	-122.375	13.0	SFO	3878	KSFC
3	1	CO	269	SFO	IAH	3	15	205	1	KSFO	...	-122.375	13.0	SFO	3878	KSFC
4	4	AA	2466	SFO	DFW	3	20	195	1	KSFO	...	-122.375	13.0	SFO	3878	KSFC

5 rows × 23 columns



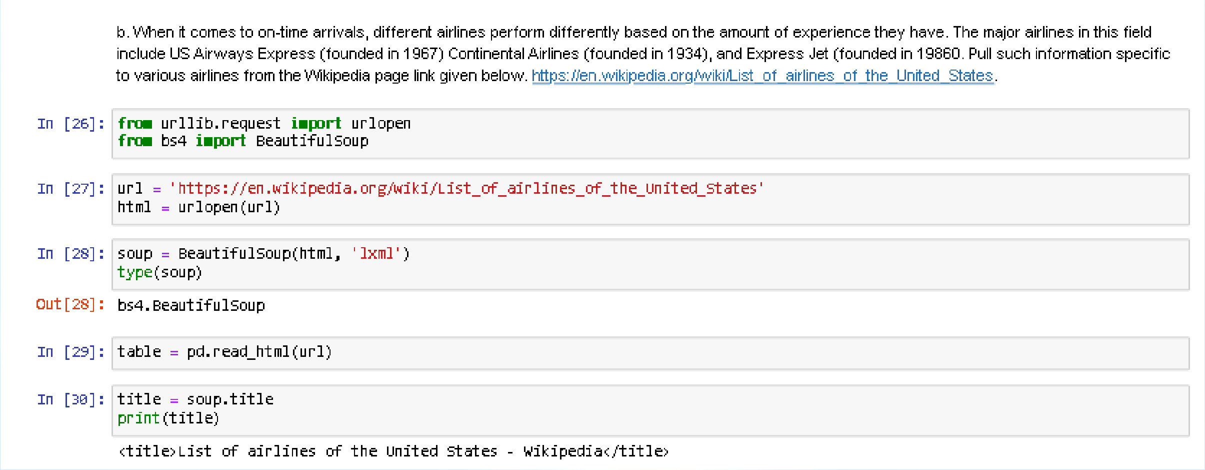
```
data.drop_duplicates(subset = ['id'], keep = 'first', inplace = True)
```

```
data
```

	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay	ident	...	longitude_deg	elevation_ft	iata_code	airport_ref	a
0	1	CO	269	SFO	IAH	3	15	205	1	KSFO	...	-122.375000	13.0	SFO	3878	
4	4	AA	2466	SFO	DFW	3	20	195	1	KSFO	...	-122.375000	13.0	SFO	3878	
8	9	DL	2606	SFO	MSP	3	35	216	1	KSFO	...	-122.375000	13.0	SFO	3878	
12	129	DL	1580	SFO	DTW	3	345	270	0	KSFO	...	-122.375000	13.0	SFO	3878	
16	150	UA	756	SFO	DEN	3	348	158	0	KSFO	...	-122.375000	13.0	SFO	3878	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2160266	451344	CO	2	GUM	HNL	1	400	430	1	PGUM	...	144.796005	298.0	GUM	5433	
2160268	469866	CO	2	GUM	HNL	2	400	430	1	PGUM	...	144.796005	298.0	GUM	5433	
2160270	488365	CO	2	GUM	HNL	3	400	430	0	PGUM	...	144.796005	298.0	GUM	5433	
2160272	506855	CO	2	GUM	HNL	4	400	430	1	PGUM	...	144.796005	298.0	GUM	5433	
2160274	525138	CO	2	GUM	HNL	5	400	430	1	PGUM	...	144.796005	298.0	GUM	5433	

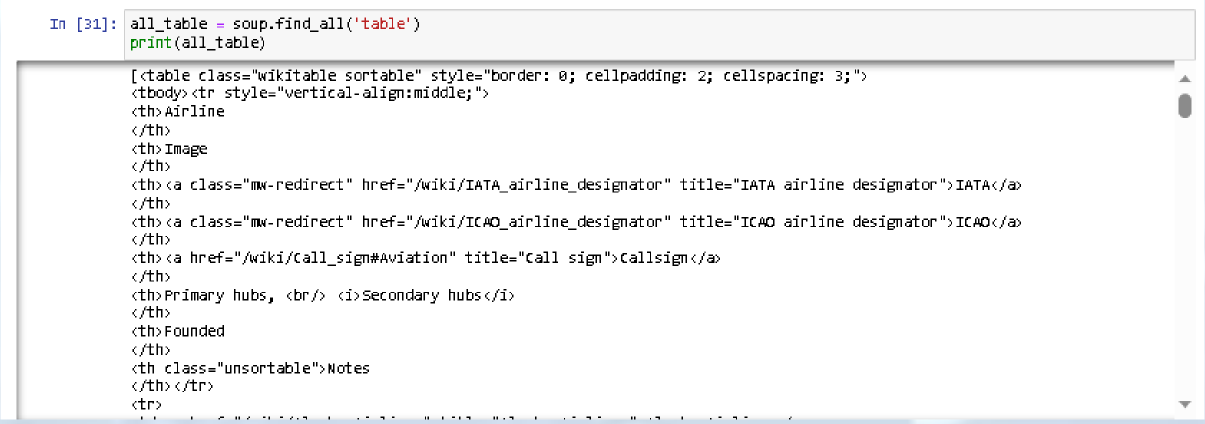
518525 rows × 23 columns

b. When it comes to on-time arrivals, different airlines perform differently based on the amount of experience they have. The major airlines in this field include US Airways Express (founded in 1967) Continental Airlines (founded in 1934), and Express Jet (founded in 19860. Pull such information specific to various airlines from the Wikipedia page link given below. https://en.wikipedia.org/wiki/List_of_airlines_of_the_United_States.

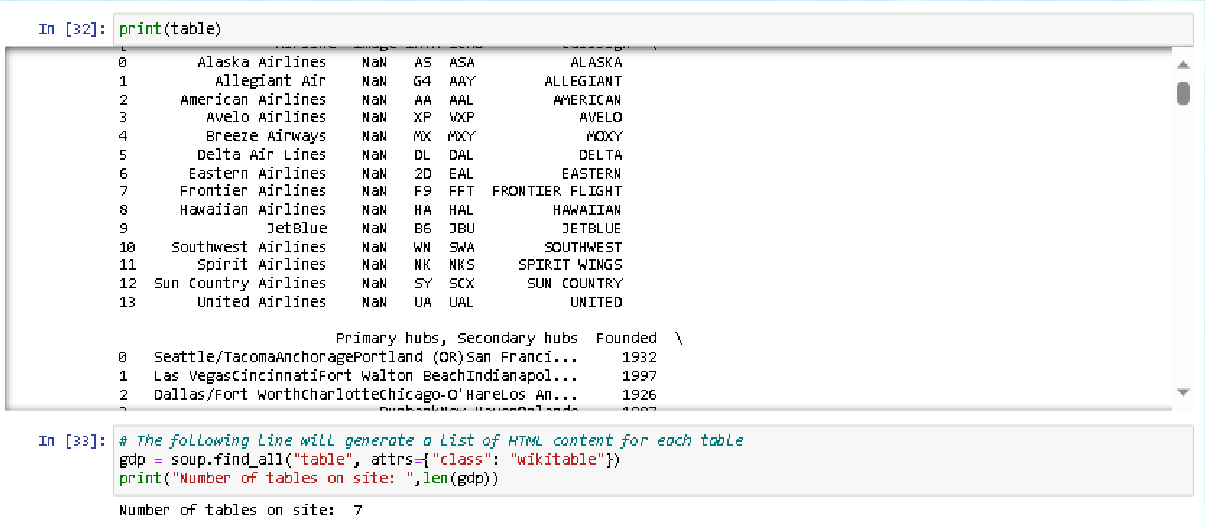


b. When it comes to on-time arrivals, different airlines perform differently based on the amount of experience they have. The major airlines in this field include US Airways Express (founded in 1967) Continental Airlines (founded in 1934), and Express Jet (founded in 19860. Pull such information specific to various airlines from the Wikipedia page link given below. https://en.wikipedia.org/wiki/List_of_airlines_of_the_United_States.

```
In [26]: from urllib.request import urlopen
         from bs4 import BeautifulSoup

In [27]: url = 'https://en.wikipedia.org/wiki/List_of_airlines_of_the_United_States'
         html = urlopen(url)

In [28]: soup = BeautifulSoup(html, 'lxml')
         type(soup)

Out[28]: bs4.BeautifulSoup

In [29]: table = pd.read_html(url)

In [30]: title = soup.title
         print(title)

         <title>List of airlines of the United States - Wikipedia</title>
```



```
In [31]: all_table = soup.find_all('table')
         print(all_table)

         [<table class="wikitable sortable" style="border: 0; cellpadding: 2; cellspacing: 3;">
         <tbody><tr style="vertical-align:middle;">
         <th>Airline
         </th>
         <th>Image
         </th>
         <th><a class="mw-redirect" href="/wiki/IATA_airline_designator" title="IATA airline designator">IATA</a>
         </th>
         <th><a class="mw-redirect" href="/wiki/ICAO_airline_designator" title="ICAO airline designator">ICAO</a>
         </th>
         <th><a href="/wiki/Call_sign#Aviation" title="Call sign">Callsign</a>
         </th>
         <th>Primary hubs, <br/> <i>Secondary hubs</i>
         </th>
         <th>Founded
         </th>
         <th class="unsortable">Notes
         </th></tr>
         <tr>
```

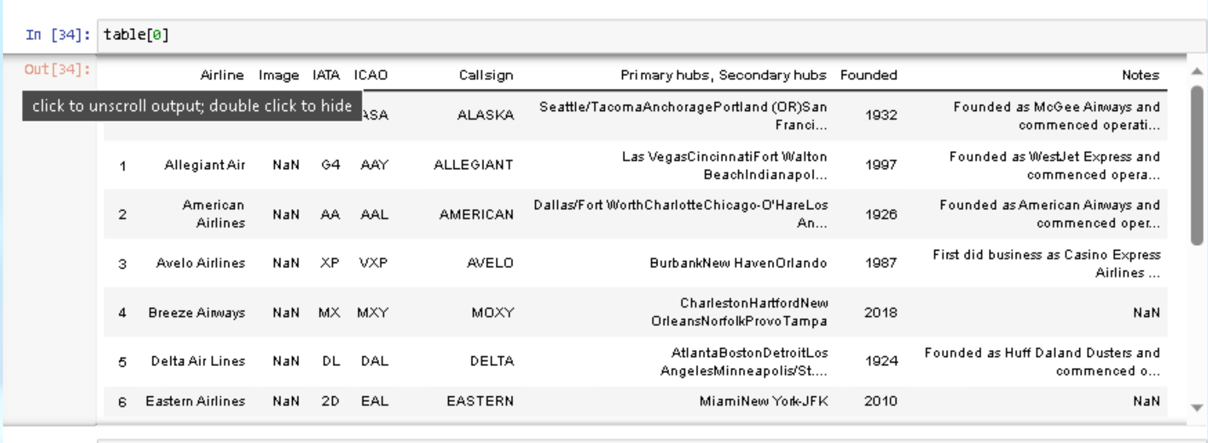


In [32]:
```
print(table)
```

```
[               Airline  Image IATA ICAO          Callsign  \
0        Alaska Airlines    NaN   AS  ASA            ALASKA
1          Allegiant Air    NaN   G4  AAY         ALLEGIANT
2      American Airlines    NaN   AA  AAL          AMERICAN
3         Avelo Airlines    NaN   XP  VXP             AVELO
4         Breeze Airways    NaN   MX  MXY              MOXY
5        Delta Air Lines    NaN   DL  DAL             DELTA
6       Eastern Airlines    NaN   2D  EAL           EASTERN
7      Frontier Airlines    NaN   F9  FFT   FRONTIER FLIGHT
8      Hawaiian Airlines    NaN   HA  HAL          HAWAIIAN
9                JetBlue    NaN   B6  JBU           JETBLUE
10    Southwest Airlines    NaN   WN  SWA         SOUTHWEST
11       Spirit Airlines    NaN   NK  NKS      SPIRIT WINGS
12  Sun Country Airlines    NaN   SY  SCX       SUN COUNTRY
13       United Airlines    NaN   UA  UAL            UNITED

                        Primary hubs, Secondary hubs  Founded  \
0   Seattle/TacomaAnchoragePortland (OR)San Franci...     1932
1   Las VegasCincinnatiFort Walton BeachIndianapol...     1997
2   Dallas/Fort WorthCharlotteChicago-O'HareLos An...     1926
3                       BurbankNew HavenOrlando     1987
```

In [33]:
```
# The following line will generate a list of HTML content for each table
gdp = soup.find_all("table", attrs={"class": "wikitable"})
print("Number of tables on site: ",len(gdp))
```

```
Number of tables on site:  7
```

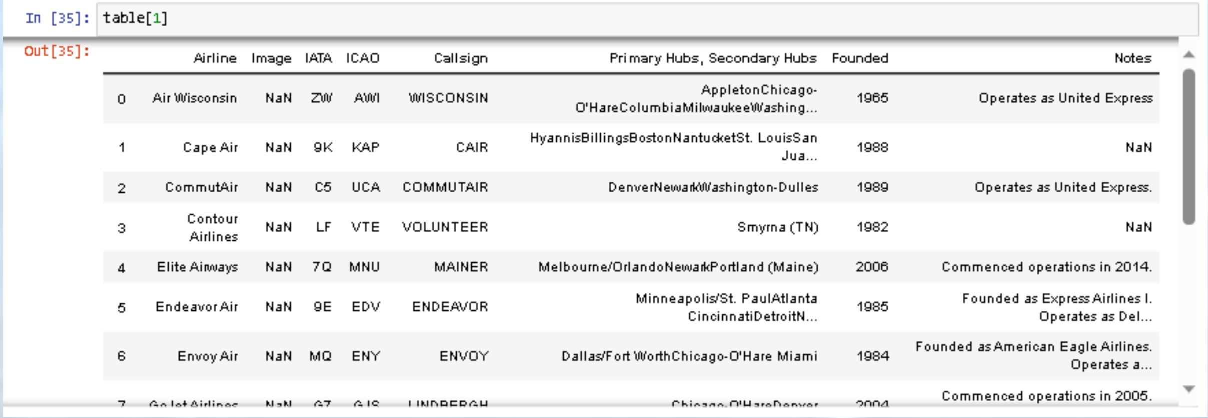


In [34]:
```
table[0]
```

Out[34]:

click to unscroll output; double click to hide

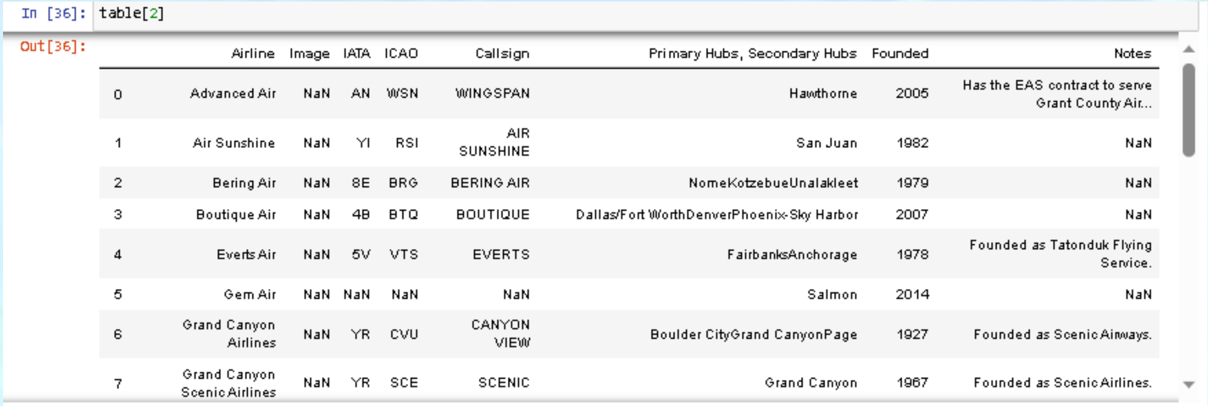
	Airline	Image	IATA	ICAO	Callsign	Primary hubs, Secondary hubs	Founded	Notes
				ASA	ALASKA	Seattle/TacomaAnchoragePortland (OR)San Franci...	1932	Founded as McGee Airways and commenced operati...
1	Allegiant Air	NaN	G4	AAY	ALLEGIANT	Las VegasCincinnatiFort Walton BeachIndianapol...	1997	Founded as WestJet Express and commenced opera...
2	American Airlines	NaN	AA	AAL	AMERICAN	Dallas/Fort WorthCharlotteChicago-O'HareLos An...	1926	Founded as American Airways and commenced oper...
3	Avelo Airlines	NaN	XP	VXP	AVELO	BurbankNew HavenOrlando	1987	First did business as Casino Express Airlines ...
4	Breeze Airways	NaN	MX	MXY	MOXY	CharlestonHartfordNew OrleansNorfolkProvoTampa	2018	NaN
5	Delta Air Lines	NaN	DL	DAL	DELTA	AtlantaBostonDetroitLos AngelesMinneapolis/St....	1924	Founded as Huff Daland Dusters and commenced o...
6	Eastern Airlines	NaN	2D	EAL	EASTERN	MiamiNew York-JFK	2010	NaN



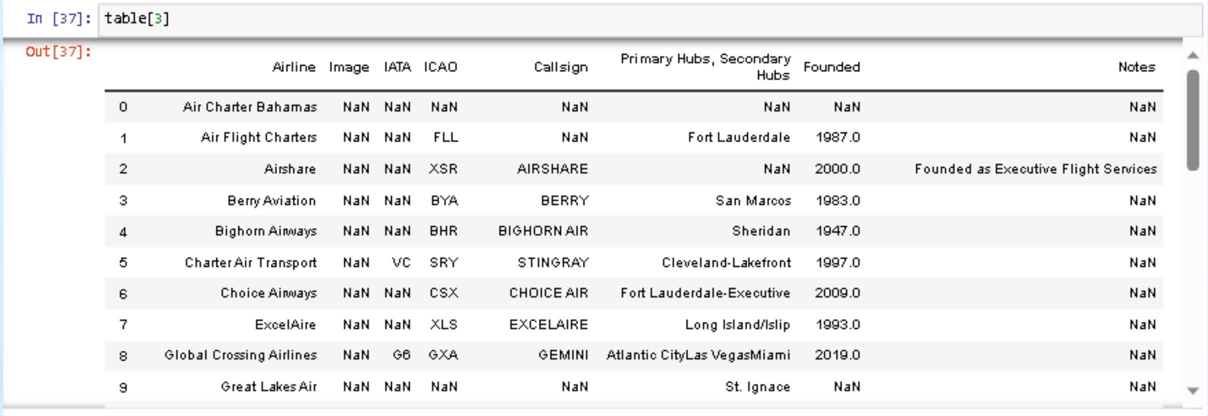
In [35]:
```
table[1]
```

Out[35]:

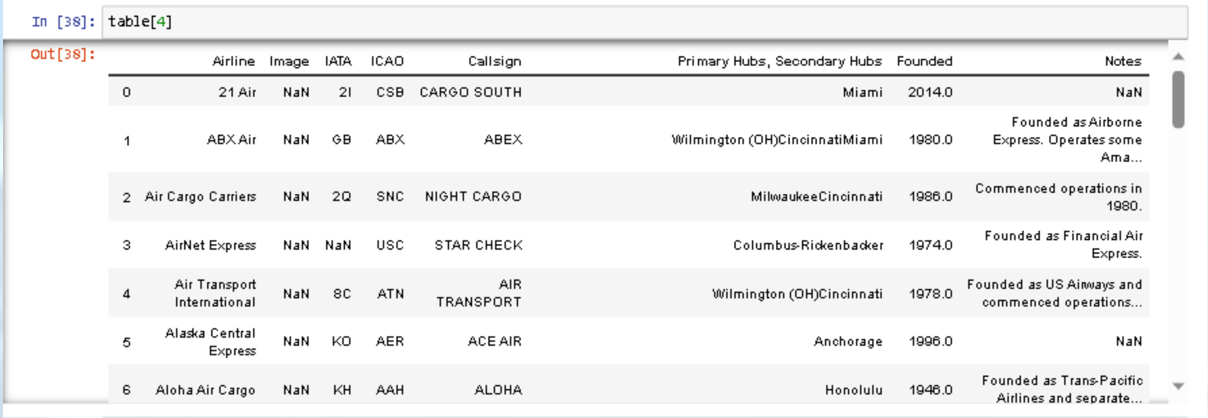
	Airline	Image	IATA	ICAO	Callsign	Primary Hubs, Secondary Hubs	Founded	Notes
0	Air Wisconsin	NaN	ZW	AWI	WISCONSIN	AppletonChicago-O'HareColumbiaMilwaukeeWashing...	1965	Operates as United Express
1	Cape Air	NaN	9K	KAP	CAIR	HyannisBillingsBostonNantucketSt. LouisSan Jua...	1988	NaN
2	CommutAir	NaN	C5	UCA	COMMUTAIR	DenverNewarkWashington-Dulles	1989	Operates as United Express.
3	Contour Airlines	NaN	LF	VTE	VOLUNTEER	Smyrna (TN)	1982	NaN
4	Elite Airways	NaN	7Q	MNU	MAINER	Melbourne/OrlandoNewarkPortland (Maine)	2006	Commenced operations in 2014.
5	Endeavor Air	NaN	9E	EDV	ENDEAVOR	Minneapolis/St. PaulAtlanta CincinnatiDetroitN...	1985	Founded as Express Airlines I. Operates as Del...
6	Envoy Air	NaN	MQ	ENY	ENVOY	Dallas/Fort WorthChicago-O'Hare Miami	1984	Founded as American Eagle Airlines. Operates a...
7	GoJet Airlines	NaN	G7	GJS	LINDBERGH	Chicago-O'HareDenver	2004	Commenced operations in 2005.



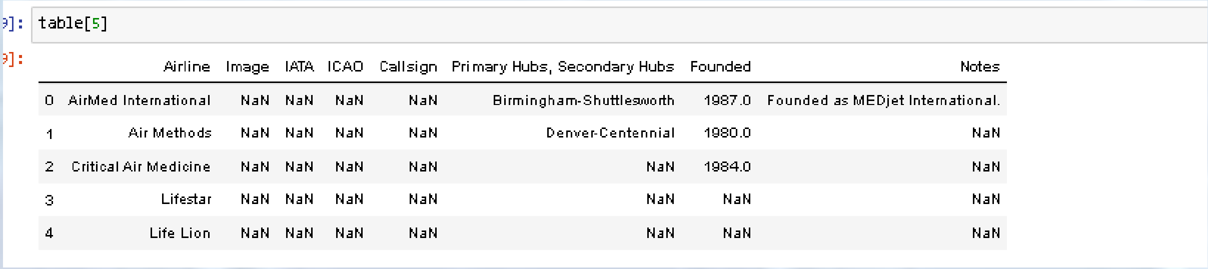
```
In [36]: table[2]

Out[36]:
             Airline  Image  IATA  ICAO     Callsign          Primary Hubs, Secondary Hubs  Founded                         Notes
 0      Advanced Air    NaN    AN   WSN    WINGSPAN                              Hawthorne     2005   Has the EAS contract to serve
                                                                                                           Grant County Air...
 1      Air Sunshine    NaN    YI   RSI         AIR                               San Juan     1982                           NaN
                                           SUNSHINE
 2        Bering Air    NaN    8E   BRG  BERING AIR              NomeKotzebueUnalakleet     1979                           NaN
 3      Boutique Air    NaN    4B   BTQ    BOUTIQUE  Dallas/Fort WorthDenverPhoenix-Sky Harbor  2007                      NaN
 4         Everts Air    NaN    5V   VTS      EVERTS                     FairbanksAnchorage     1978    Founded as Tatonduk Flying
                                                                                                                      Service.
 5           Gem Air    NaN   NaN   NaN         NaN                                 Salmon     2014                           NaN
 6       Grand Canyon   NaN    YR   CVU      CANYON           Boulder CityGrand CanyonPage     1927     Founded as Scenic Airways.
             Airlines                           VIEW
 7       Grand Canyon   NaN    YR   SCE      SCENIC                           Grand Canyon     1967    Founded as Scenic Airlines.
      Scenic Airlines
```



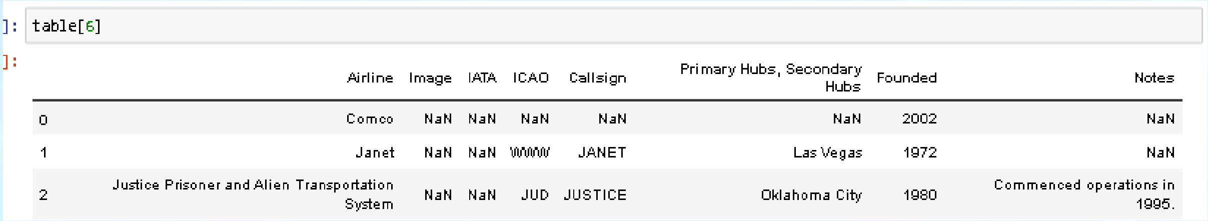
```
In [37]: table[3]

Out[37]:
                Airline  Image  IATA  ICAO    Callsign  Primary Hubs, Secondary  Founded                              Notes
                                                                          Hubs
 0   Air Charter Bahamas    NaN   NaN   NaN         NaN                     NaN      NaN                                NaN
 1    Air Flight Charters   NaN   NaN   FLL         NaN          Fort Lauderdale   1987.0                                NaN
 2               Airshare   NaN   NaN   XSR   AIRSHARE                      NaN   2000.0  Founded as Executive Flight Services
 3         Berry Aviation   NaN   NaN   BYA      BERRY               San Marcos   1983.0                                NaN
 4        Bighorn Airways   NaN   NaN   BHR  BIGHORN AIR                Sheridan   1947.0                                NaN
 5   Charter Air Transport  NaN    VC   SRY    STINGRAY       Cleveland-Lakefront   1997.0                                NaN
 6        Choice Airways    NaN   NaN   CSX  CHOICE AIR  Fort Lauderdale-Executive   2009.0                                NaN
 7             ExcelAire    NaN   NaN   XLS   EXCELAIRE         Long Island/Islip   1993.0                                NaN
 8  Global Crossing Airlines NaN    G6   GXA      GEMINI  Atlantic CityLas VegasMiami 2019.0                                NaN
 9        Great Lakes Air   NaN   NaN   NaN         NaN                St. Ignace      NaN                                NaN
```



```
In [38]: table[4]

Out[38]:
             Airline  Image  IATA  ICAO      Callsign     Primary Hubs, Secondary Hubs  Founded                     Notes
 0            21 Air    NaN    2I   CSB   CARGO SOUTH                            Miami   2014.0                       NaN
 1           ABX Air    NaN    GB   ABX          ABEX   Wilmington (OH)CincinnatiMiami   1980.0       Founded as Airborne
                                                                                                  Express. Operates some
                                                                                                                  Ama...
 2  Air Cargo Carriers  NaN    2Q   SNC   NIGHT CARGO              MilwaukeeCincinnati   1986.0   Commenced operations in
                                                                                                                   1980.
 3     AirNet Express   NaN   NaN   USC    STAR CHECK              Columbus-Rickenbacker  1974.0     Founded as Financial Air
                                                                                                                Express.
 4       Air Transport  NaN    8C   ATN           AIR        Wilmington (OH)Cincinnati   1978.0   Founded as US Airways and
         International                      TRANSPORT                                             commenced operations...
 5      Alaska Central  NaN    KO   AER       ACE AIR                        Anchorage   1996.0                       NaN
             Express
 6     Aloha Air Cargo  NaN    KH   AAH         ALOHA                         Honolulu   1946.0     Founded as Trans-Pacific
                                                                                                  Airlines and separate...
```



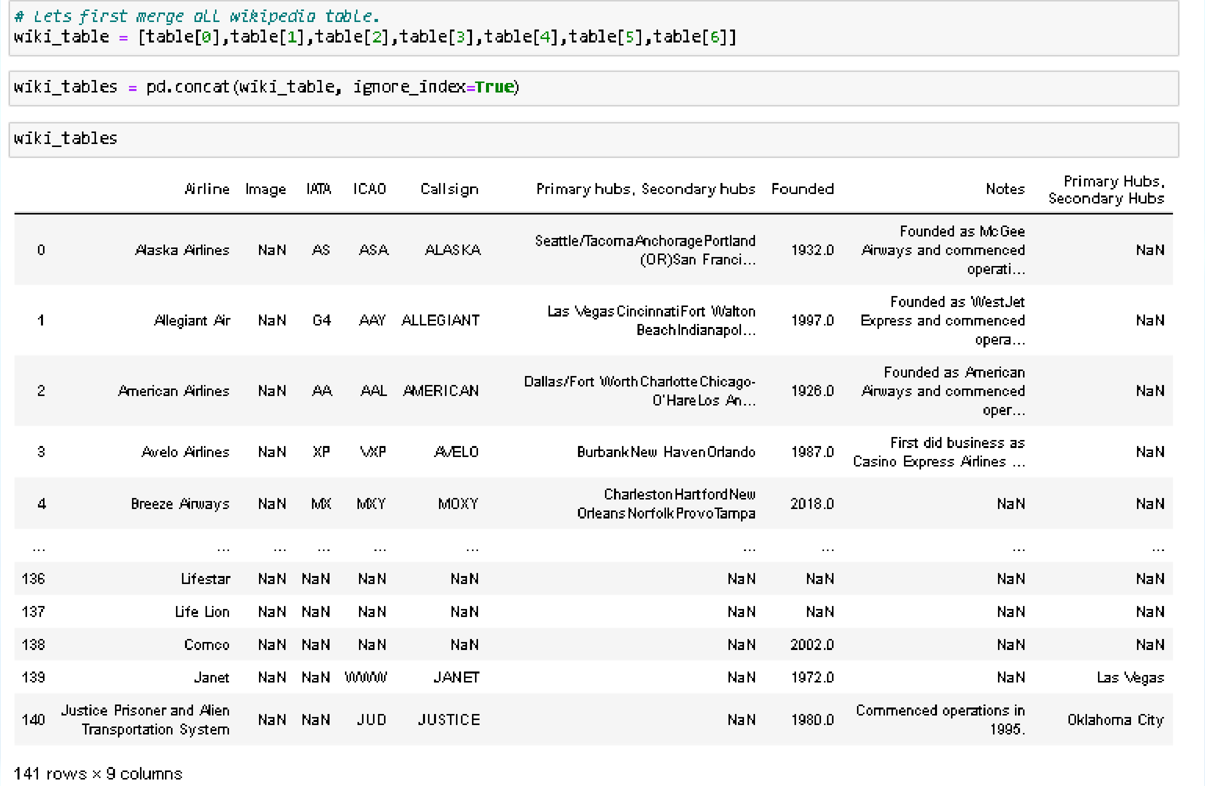
```
]: table[5]

]:
               Airline  Image  IATA  ICAO  Callsign  Primary Hubs, Secondary Hubs  Founded                           Notes
 0  AirMed International    NaN   NaN   NaN       NaN     Birmingham-Shuttlesworth   1987.0  Founded as MEDjet International.
 1           Air Methods    NaN   NaN   NaN       NaN            Denver-Centennial   1980.0                              NaN
 2  Critical Air Medicine   NaN   NaN   NaN       NaN                          NaN   1984.0                              NaN
 3              Lifestar    NaN   NaN   NaN       NaN                          NaN      NaN                              NaN
 4             Life Lion    NaN   NaN   NaN       NaN                          NaN      NaN                              NaN
```



```
]: table[6]
```

]:

	Airline	Image	IATA	ICAO	Callsign	Primary Hubs, Secondary Hubs	Founded	Notes
0	Comco	NaN	NaN	NaN	NaN	NaN	2002	NaN
1	Janet	NaN	NaN	WWW	JANET	Las Vegas	1972	NaN
2	Justice Prisoner and Alien Transportation System	NaN	NaN	JUD	JUSTICE	Oklahoma City	1980	Commenced operations in 1995.



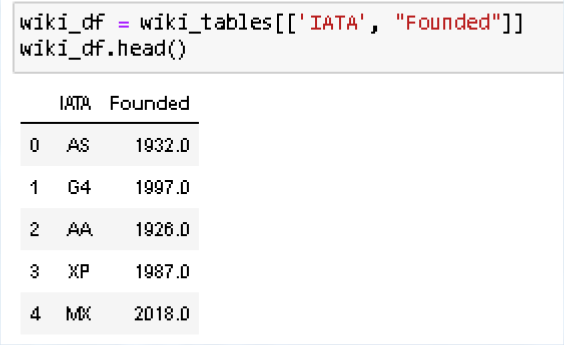
```
# Lets first merge all wikipedia table.
wiki_table = [table[0],table[1],table[2],table[3],table[4],table[5],table[6]]
```

```
wiki_tables = pd.concat(wiki_table, ignore_index=True)
```

```
wiki_tables
```

	Airline	Image	IATA	ICAO	Callsign	Primary hubs, Secondary hubs	Founded	Notes	Primary Hubs, Secondary Hubs
0	Alaska Airlines	NaN	AS	ASA	ALASKA	Seattle/TacomaAnchoragePortland (OR)San Franci...	1932.0	Founded as McGee Airways and commenced operati...	NaN
1	Allegiant Air	NaN	G4	AAY	ALLEGIANT	Las VegasCincinnatiFort Walton BeachIndianapol...	1997.0	Founded as WestJet Express and commenced opera...	NaN
2	American Airlines	NaN	AA	AAL	AMERICAN	Dallas/Fort WorthCharlotteChicago-O'HareLos An...	1926.0	Founded as American Airways and commenced oper...	NaN
3	Avelo Airlines	NaN	XP	VXP	AVELO	BurbankNew HavenOrlando	1987.0	First did business as Casino Express Airlines ...	NaN
4	Breeze Airways	NaN	MX	MXY	MOXY	CharlestonHartfordNew OrleansNorfolkProvoTampa	2018.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...
136	Lifestar	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
137	Life Lion	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
138	Comco	NaN	NaN	NaN	NaN	NaN	2002.0	NaN	NaN
139	Janet	NaN	NaN	WWW	JANET	NaN	1972.0	NaN	Las Vegas
140	Justice Prisoner and Alien Transportation System	NaN	NaN	JUD	JUSTICE	NaN	1980.0	Commenced operations in 1995.	Oklahoma City

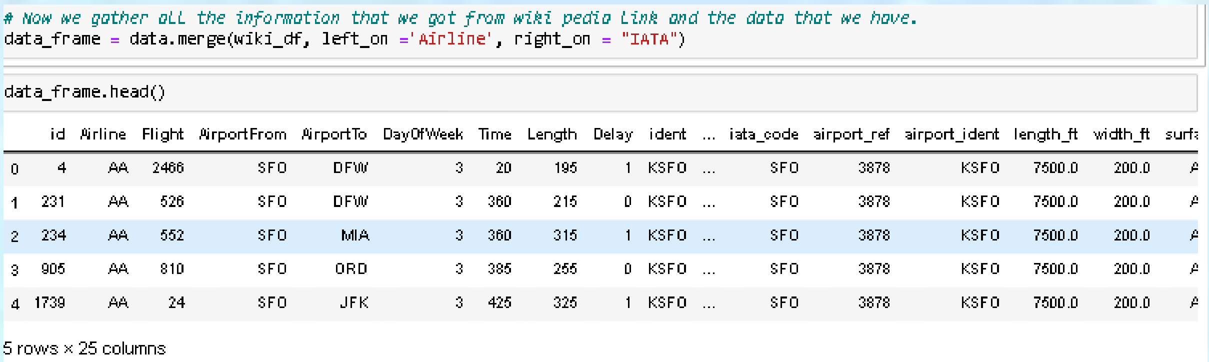
141 rows × 9 columns



```
wiki_df = wiki_tables[['IATA', "Founded"]]
wiki_df.head()
```

	IATA	Founded
0	AS	1932.0
1	G4	1997.0
2	AA	1926.0
3	XP	1987.0
4	MX	2018.0

c. You should then get all the information gathered so far in one place



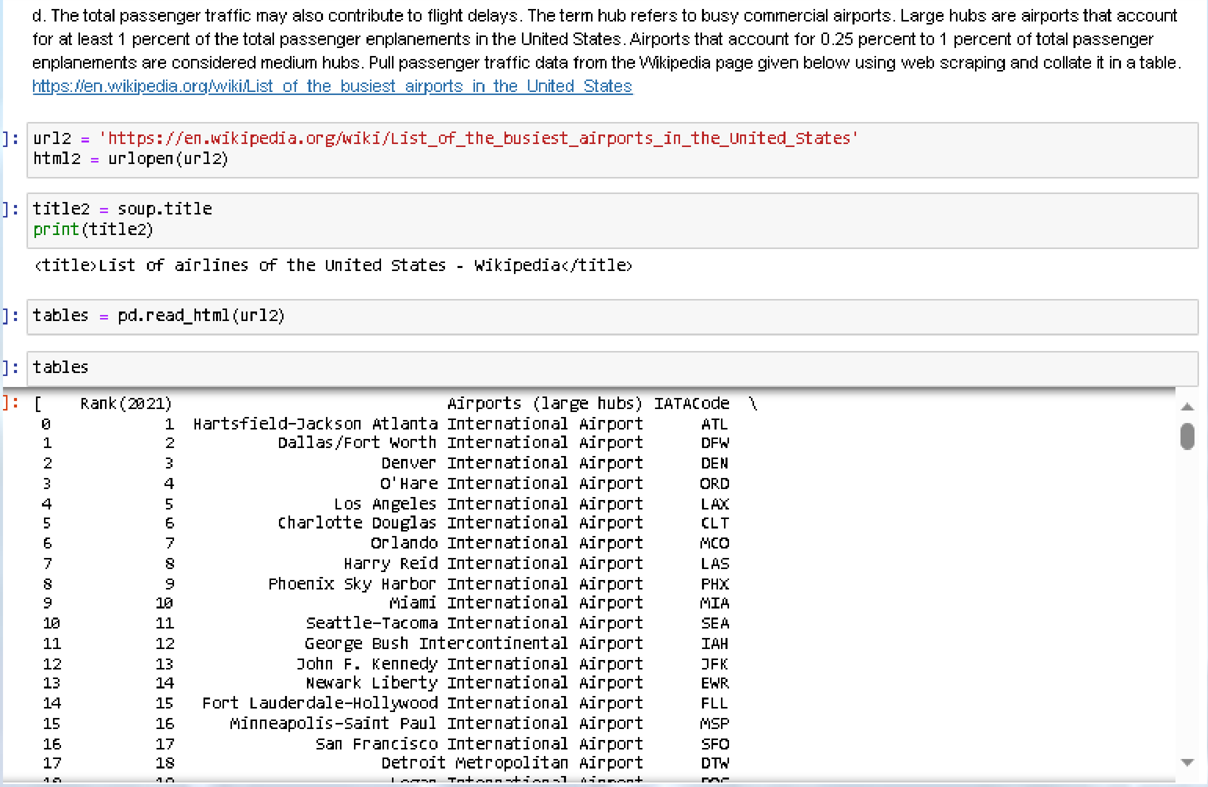
```python
# Now we gather all the information that we got from wiki pedia Link and the data that we have.
data_frame = data.merge(wiki_df, left_on ='Airline', right_on = "IATA")
```

```python
data_frame.head()
```

	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay	ident	...	iata_code	airport_ref	airport_ident	length_ft	width_ft	surfa
0	4	AA	2466	SFO	DFW	3	20	195	1	KSFO	...	SFO	3878	KSFO	7500.0	200.0	A
1	231	AA	526	SFO	DFW	3	360	215	0	KSFO	...	SFO	3878	KSFO	7500.0	200.0	A
2	234	AA	552	SFO	MIA	3	360	315	1	KSFO	...	SFO	3878	KSFO	7500.0	200.0	A
3	905	AA	810	SFO	ORD	3	385	255	0	KSFO	...	SFO	3878	KSFO	7500.0	200.0	A
4	1739	AA	24	SFO	JFK	3	425	325	1	KSFO	...	SFO	3878	KSFO	7500.0	200.0	A

5 rows × 25 columns

d. The total passenger traffic may also contribute to flight delays. The term hub refers to busy commercial airports. Large hubs are airports that account for at least 1 percent of the total passenger enplanements in the United States. Airports that account for 0.25 percent to 1 percent of total passenger enplanements are considered medium hubs. Pull passenger traffic data from the Wikipedia page given below using web scraping and collate it in a table. https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_the_United_States



d. The total passenger traffic may also contribute to flight delays. The term hub refers to busy commercial airports. Large hubs are airports that account for at least 1 percent of the total passenger enplanements in the United States. Airports that account for 0.25 percent to 1 percent of total passenger enplanements are considered medium hubs. Pull passenger traffic data from the Wikipedia page given below using web scraping and collate it in a table. https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_the_United_States

```python
url2 = 'https://en.wikipedia.org/wiki/List_of_the_busiest_airports_in_the_United_States'
html2 = urlopen(url2)
```

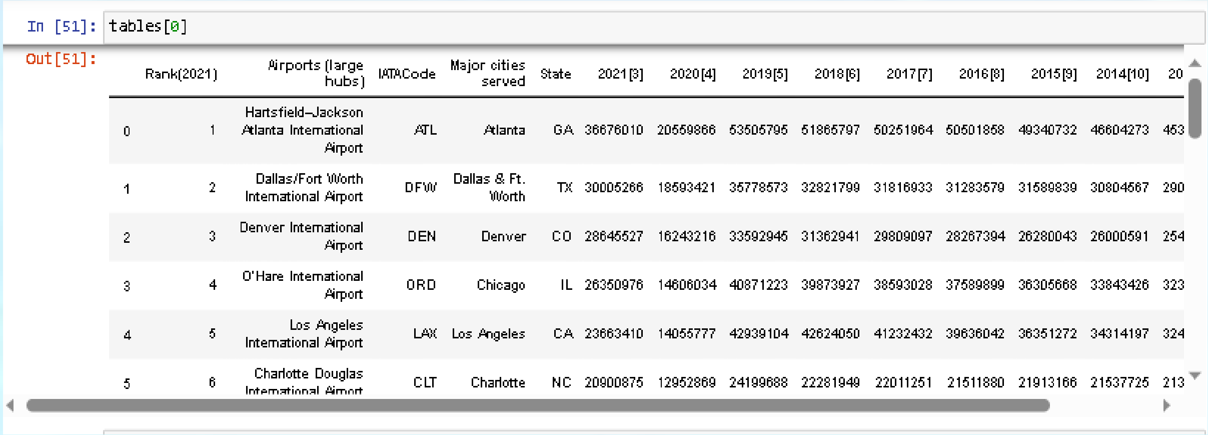
```python
title2 = soup.title
print(title2)
```

```
<title>List of airlines of the United States - Wikipedia</title>
```

```python
tables = pd.read_html(url2)
```

```python
tables
```

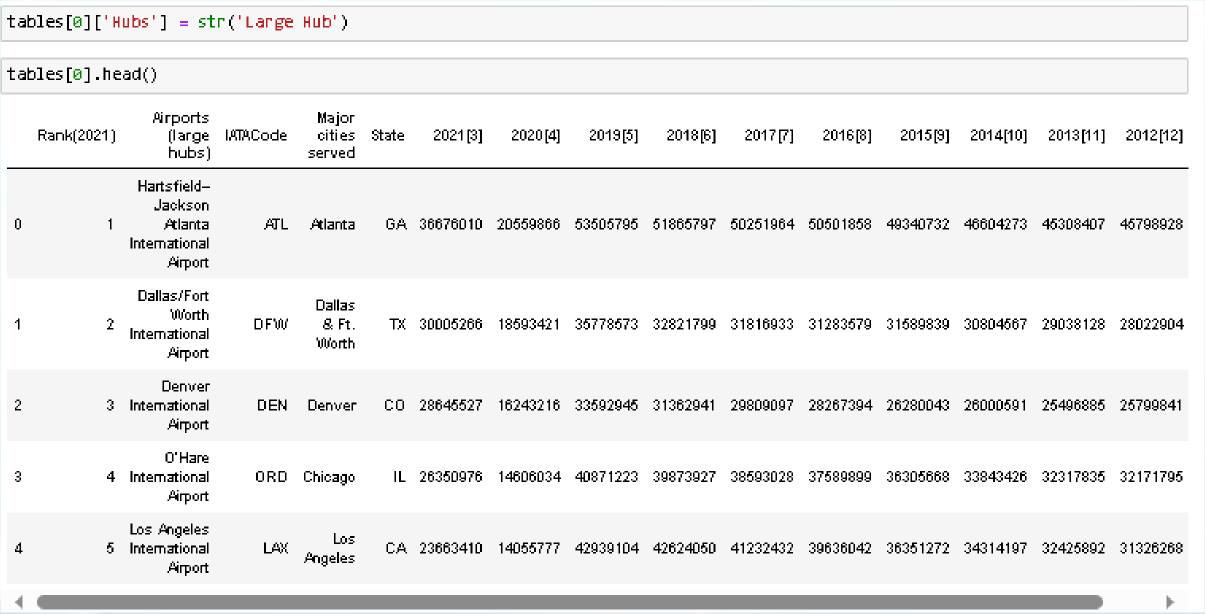
```
[    Rank(2021)                       Airports (large hubs) IATACode  \
 0            1  Hartsfield-Jackson Atlanta International Airport      ATL
 1            2        Dallas/Fort Worth International Airport      DFW
 2            3                  Denver International Airport      DEN
 3            4                  O'Hare International Airport      ORD
 4            5             Los Angeles International Airport      LAX
 5            6       Charlotte Douglas International Airport      CLT
 6            7                 Orlando International Airport      MCO
 7            8              Harry Reid International Airport      LAS
 8            9      Phoenix Sky Harbor International Airport      PHX
 9           10                   Miami International Airport      MIA
 10          11         Seattle-Tacoma International Airport      SEA
 11          12          George Bush Intercontinental Airport      IAH
 12          13           John F. Kennedy International Airport      JFK
 13          14         Newark Liberty International Airport      EWR
 14          15 Fort Lauderdale-Hollywood International Airport      FLL
 15          16    Minneapolis-Saint Paul International Airport      MSP
 16          17         San Francisco International Airport      SFO
 17          18                  Detroit Metropolitan Airport      DTW
 18          19                   Logan International Airport      BOS
```



```
In [51]: tables[0]
```

Out[51]:

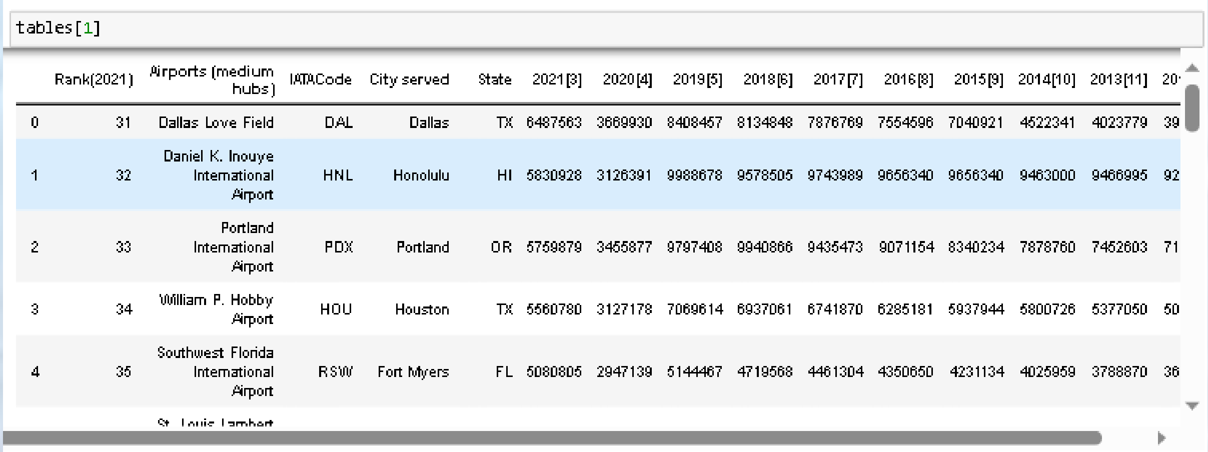
	Rank(2021)	Airports (large hubs)	IATACode	Major cities served	State	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	20
0	1	Hartsfield–Jackson Atlanta International Airport	ATL	Atlanta	GA	36676010	20559866	53505795	51865797	50251964	50501858	49340732	46604273	453
1	2	Dallas/Fort Worth International Airport	DFW	Dallas & Ft. Worth	TX	30005266	18593421	35778573	32821799	31816933	31283579	31589839	30804567	290
2	3	Denver International Airport	DEN	Denver	CO	28645527	16243216	33592945	31362941	29809097	28267394	26280043	26000591	254
3	4	O'Hare International Airport	ORD	Chicago	IL	26350976	14606034	40871223	39873927	38593028	37589899	36305668	33843426	323
4	5	Los Angeles International Airport	LAX	Los Angeles	CA	23663410	14055777	42939104	42624050	41232432	39636042	36351272	34314197	324
5	6	Charlotte Douglas International Airport	CLT	Charlotte	NC	20900875	12952869	24199688	22281949	22011251	21511880	21913166	21537725	213



```
tables[0]['Hubs'] = str('Large Hub')
```

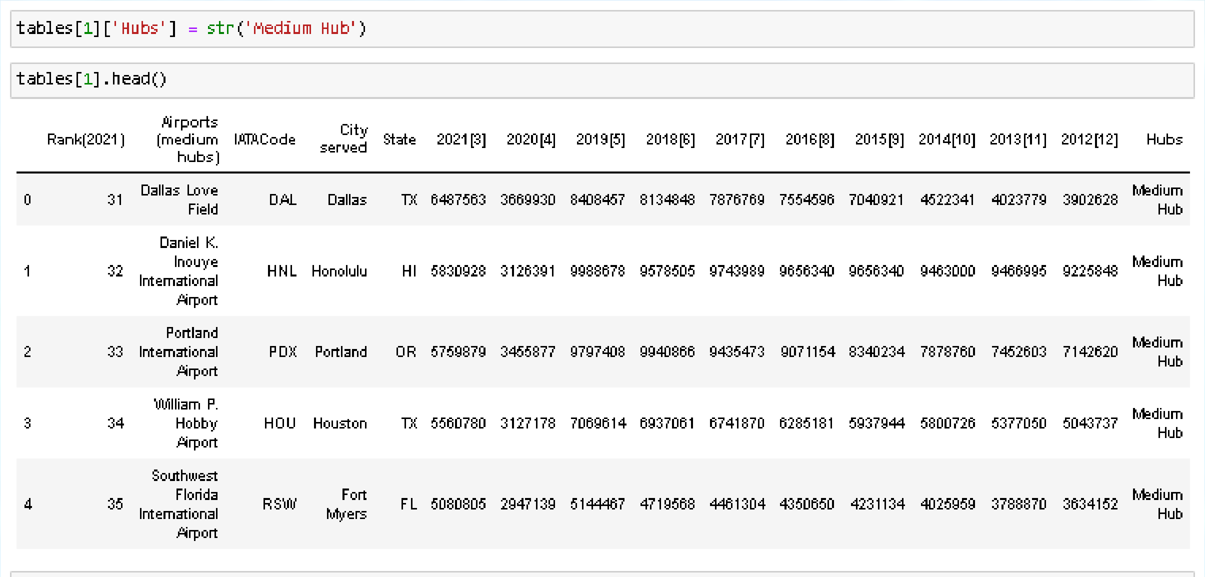
```
tables[0].head()
```

	Rank(2021)	Airports (large hubs)	IATACode	Major cities served	State	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]	2012[12]
0	1	Hartsfield–Jackson Atlanta International Airport	ATL	Atlanta	GA	36676010	20559866	53505795	51865797	50251964	50501858	49340732	46604273	45308407	45798928
1	2	Dallas/Fort Worth International Airport	DFW	Dallas & Ft. Worth	TX	30005266	18593421	35778573	32821799	31816933	31283579	31589839	30804567	29038128	28022904
2	3	Denver International Airport	DEN	Denver	CO	28645527	16243216	33592945	31362941	29809097	28267394	26280043	26000591	25496885	25799841
3	4	O'Hare International Airport	ORD	Chicago	IL	26350976	14606034	40871223	39873927	38593028	37589899	36305668	33843426	32317835	32171795
4	5	Los Angeles International Airport	LAX	Los Angeles	CA	23663410	14055777	42939104	42624050	41232432	39636042	36351272	34314197	32425892	31326268



```
tables[1]
```

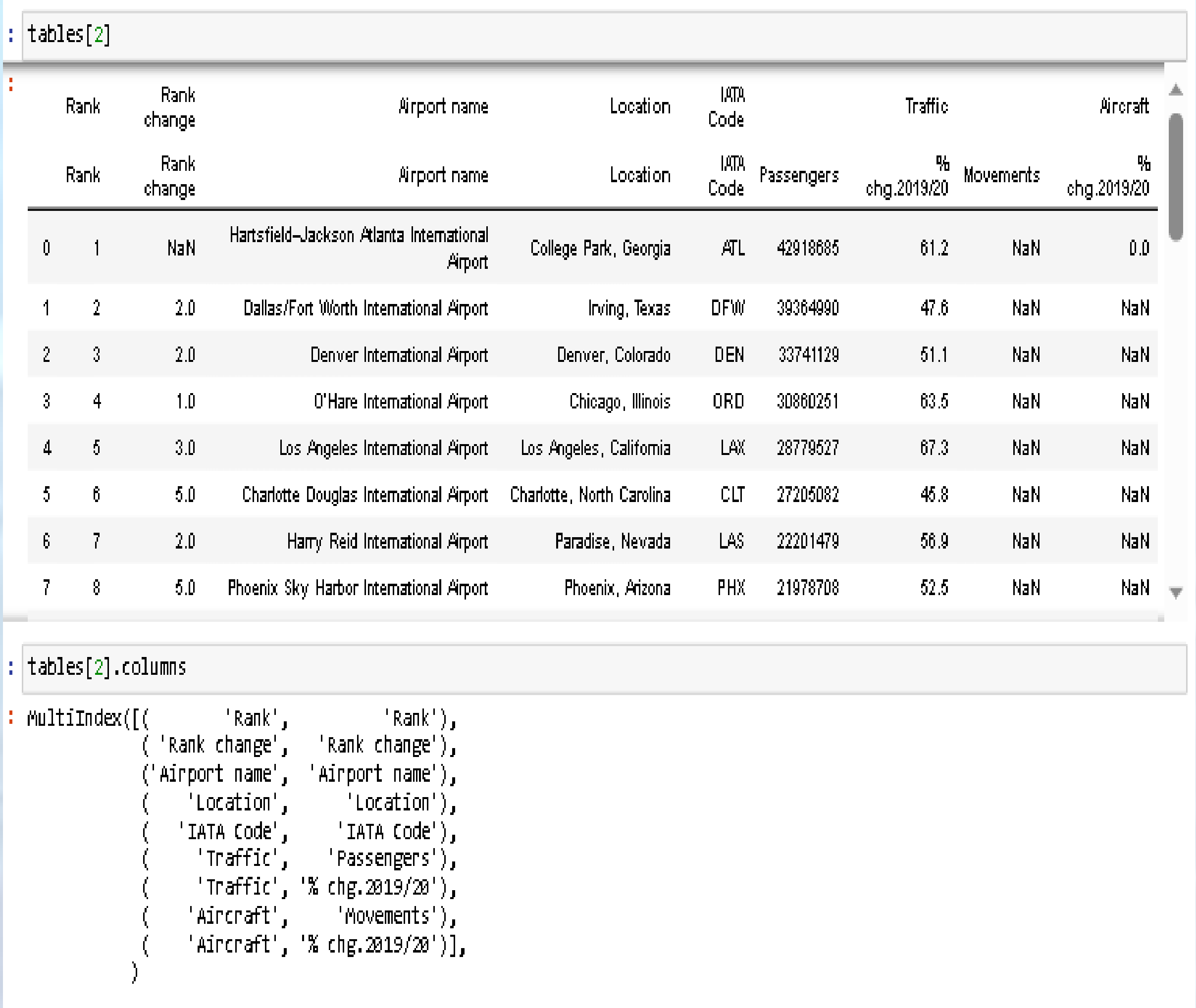
	Rank(2021)	Airports (medium hubs)	IATACode	City served	State	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]	20
0	31	Dallas Love Field	DAL	Dallas	TX	6487563	3669930	8408457	8134848	7876769	7554596	7040921	4522341	4023779	39
1	32	Daniel K. Inouye International Airport	HNL	Honolulu	HI	5830928	3126391	9988678	9578505	9743989	9656340	9656340	9463000	9466995	92
2	33	Portland International Airport	PDX	Portland	OR	5759879	3455877	9797408	9940866	9435473	9071154	8340234	7878760	7452603	71
3	34	William P. Hobby Airport	HOU	Houston	TX	5560780	3127178	7069614	6937061	6741870	6285181	5937944	5800726	5377050	50
4	35	Southwest Florida International Airport	RSW	Fort Myers	FL	5080805	2947139	5144467	4719568	4461304	4350650	4231134	4025959	3788870	36
		St. Louis Lambert													



```
tables[1]['Hubs'] = str('Medium Hub')
```

```
tables[1].head()
```

	Rank(2021)	Airports (medium hubs)	IATACode	City served	State	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]	2012[12]	Hubs
0	31	Dallas Love Field	DAL	Dallas	TX	6487563	3669930	8408457	8134848	7876769	7554596	7040921	4522341	4023779	3902628	Medium Hub
1	32	Daniel K. Inouye International Airport	HNL	Honolulu	HI	5830928	3126391	9988678	9578505	9743989	9656340	9656340	9463000	9466995	9225848	Medium Hub
2	33	Portland International Airport	PDX	Portland	OR	5759879	3455877	9797408	9940866	9435473	9071154	8340234	7878760	7452603	7142620	Medium Hub
3	34	William P. Hobby Airport	HOU	Houston	TX	5560780	3127178	7069614	6937061	6741870	6285181	5937944	5800726	5377050	5043737	Medium Hub
4	35	Southwest Florida International Airport	RSW	Fort Myers	FL	5080805	2947139	5144467	4719568	4461304	4350650	4231134	4025959	3788870	3634152	Medium Hub

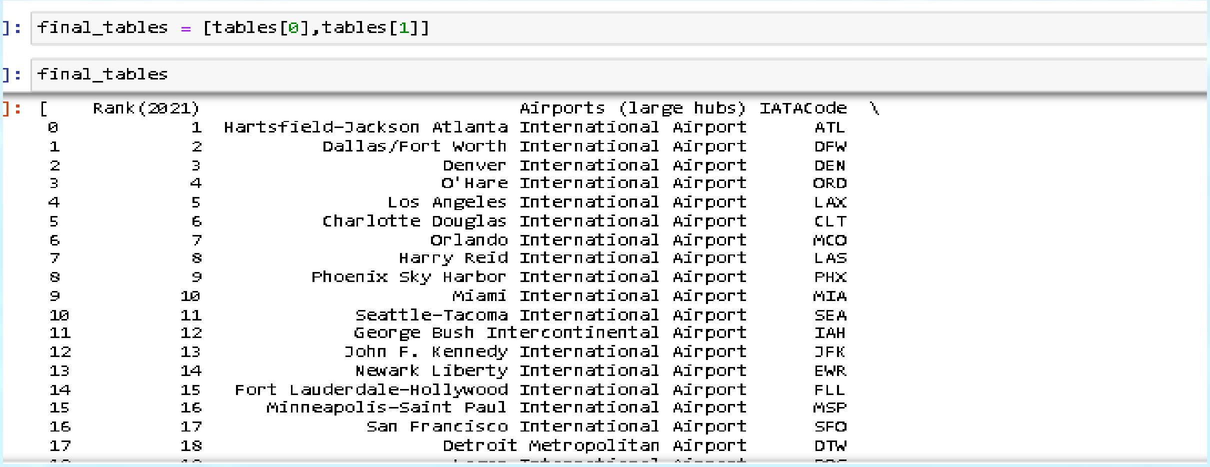


```
: tables[2]
```

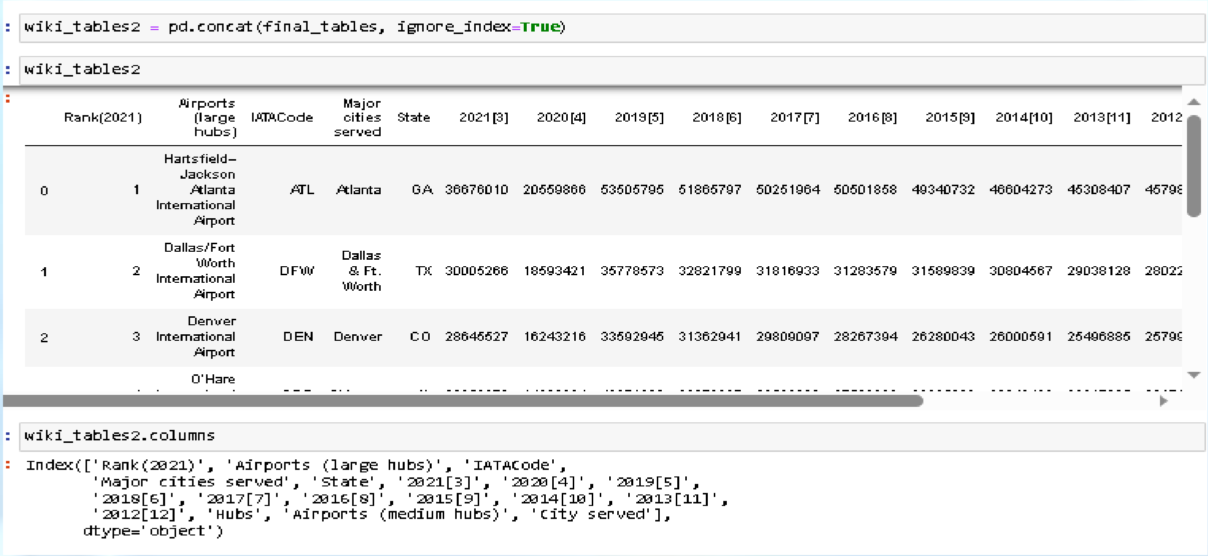
	Rank	Rank change	Airport name	Location	IATA Code	Traffic		Aircraft	
	Rank	Rank change	Airport name	Location	IATA Code	Passengers	% chg.2019/20	Movements	% chg.2019/20
0	1	NaN	Hartsfield–Jackson Atlanta International Airport	College Park, Georgia	ATL	42918685	61.2	NaN	0.0
1	2	2.0	Dallas/Fort Worth International Airport	Irving, Texas	DFW	39364990	47.6	NaN	NaN
2	3	2.0	Denver International Airport	Denver, Colorado	DEN	33741129	51.1	NaN	NaN
3	4	1.0	O'Hare International Airport	Chicago, Illinois	ORD	30860251	63.5	NaN	NaN
4	5	3.0	Los Angeles International Airport	Los Angeles, California	LAX	28779527	67.3	NaN	NaN
5	6	5.0	Charlotte Douglas International Airport	Charlotte, North Carolina	CLT	27205082	45.8	NaN	NaN
6	7	2.0	Harry Reid International Airport	Paradise, Nevada	LAS	22201479	56.9	NaN	NaN
7	8	5.0	Phoenix Sky Harbor International Airport	Phoenix, Arizona	PHX	21978708	52.5	NaN	NaN

```
: tables[2].columns
```

```
: MultiIndex([(        'Rank',          'Rank'),
            ( 'Rank change',   'Rank change'),
            ('Airport name',  'Airport name'),
            (    'Location',      'Location'),
            (   'IATA Code',     'IATA Code'),
            (     'Traffic',    'Passengers'),
            (     'Traffic', '% chg.2019/20'),
            (    'Aircraft',     'Movements'),
            (    'Aircraft', '% chg.2019/20')],
           )
```



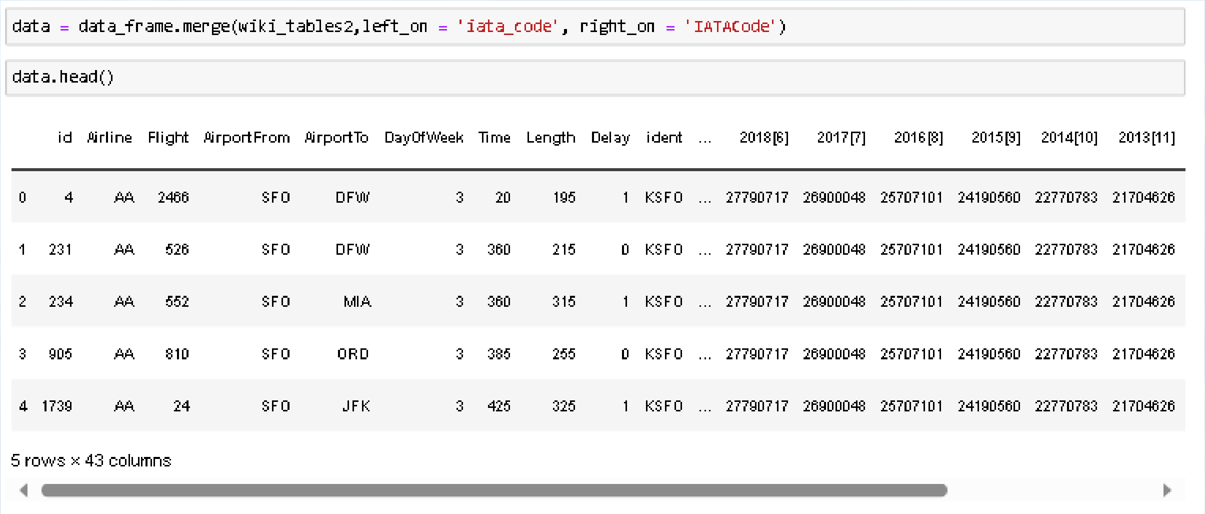
```
]: final_tables = [tables[0],tables[1]]

]: final_tables

]: [    Rank(2021)                          Airports (large hubs) IATACode  \
 0            1  Hartsfield-Jackson Atlanta International Airport      ATL
 1            2          Dallas/Fort Worth International Airport      DFW
 2            3                     Denver International Airport      DEN
 3            4                    O'Hare International Airport      ORD
 4            5                Los Angeles International Airport      LAX
 5            6           Charlotte Douglas International Airport      CLT
 6            7                    Orlando International Airport      MCO
 7            8                 Harry Reid International Airport      LAS
 8            9         Phoenix Sky Harbor International Airport      PHX
 9           10                      Miami International Airport      MIA
 10          11             Seattle-Tacoma International Airport      SEA
 11          12            George Bush Intercontinental Airport      IAH
 12          13              John F. Kennedy International Airport      JFK
 13          14             Newark Liberty International Airport      EWR
 14          15  Fort Lauderdale-Hollywood International Airport      FLL
 15          16        Minneapolis-Saint Paul International Airport      MSP
 16          17            San Francisco International Airport      SFO
 17          18               Detroit Metropolitan Airport      DTW
```



```
: wiki_tables2 = pd.concat(final_tables, ignore_index=True)

: wiki_tables2
```

	Rank(2021)	Airports (large hubs)	IATACode	Major cities served	State	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]	2012
0	1	Hartsfield–Jackson Atlanta International Airport	ATL	Atlanta	GA	36676010	20559866	53505795	51865797	50251964	50501858	49340732	46604273	45308407	45798
1	2	Dallas/Fort Worth International Airport	DFW	Dallas & Ft. Worth	TX	30005266	18593421	35778573	32821799	31816933	31283579	31589839	30804567	29038128	28022
2	3	Denver International Airport	DEN	Denver	CO	28645527	16243216	33592945	31362941	29809097	28267394	26280043	26000591	25496885	25799
		O'Hare													

```
: wiki_tables2.columns

: Index(['Rank(2021)', 'Airports (large hubs)', 'IATACode',
        'Major cities served', 'State', '2021[3]', '2020[4]', '2019[5]',
        '2018[6]', '2017[7]', '2016[8]', '2015[9]', '2014[10]', '2013[11]',
        '2012[12]', 'Hubs', 'Airports (medium hubs)', 'City served'],
       dtype='object')
```

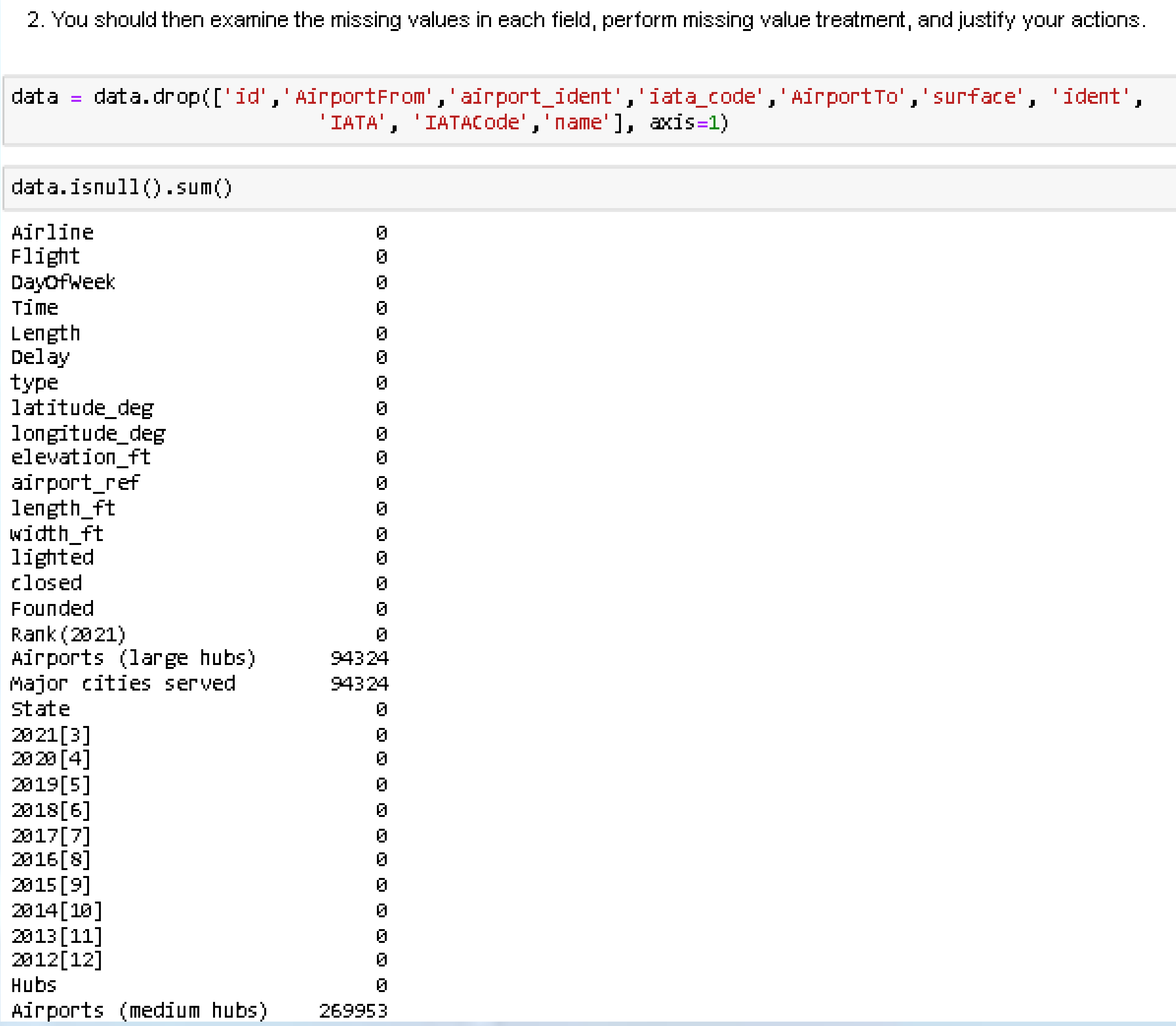


```
data = data_frame.merge(wiki_tables2,left_on = 'iata_code', right_on = 'IATACode')

data.head()
```

	id	Airline	Flight	AirportFrom	AirportTo	DayOfWeek	Time	Length	Delay	ident	...	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]
0	4	AA	2466	SFO	DFW	3	20	195	1	KSFO	...	27790717	26900048	25707101	24190560	22770783	21704626
1	231	AA	526	SFO	DFW	3	360	215	0	KSFO	...	27790717	26900048	25707101	24190560	22770783	21704626
2	234	AA	552	SFO	MIA	3	360	315	1	KSFO	...	27790717	26900048	25707101	24190560	22770783	21704626
3	905	AA	810	SFO	ORD	3	385	255	0	KSFO	...	27790717	26900048	25707101	24190560	22770783	21704626
4	1739	AA	24	SFO	JFK	3	425	325	1	KSFO	...	27790717	26900048	25707101	24190560	22770783	21704626

5 rows × 43 columns

2.You should then examine the missing values in each field, perform missing value treatment, and justify your actions

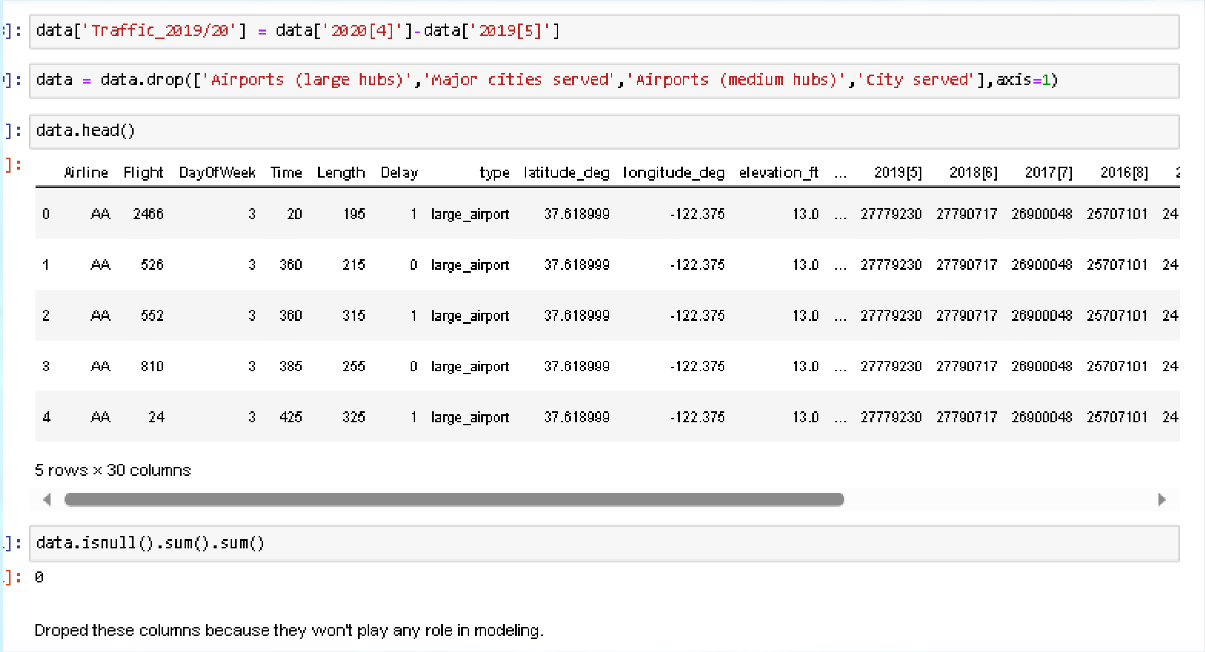


2. You should then examine the missing values in each field, perform missing value treatment, and justify your actions.

```
data = data.drop(['id','AirportFrom','airport_ident','iata_code','AirportTo','surface', 'ident',
                          'IATA', 'IATACode','name'], axis=1)
```

```
data.isnull().sum()
```

```
Airline                      0
Flight                       0
DayOfWeek                    0
Time                         0
Length                       0
Delay                        0
type                         0
latitude_deg                 0
longitude_deg                0
elevation_ft                 0
airport_ref                  0
length_ft                    0
width_ft                     0
lighted                      0
closed                       0
Founded                      0
Rank(2021)                   0
Airports (large hubs)    94324
Major cities served      94324
State                        0
2021[3]                      0
2020[4]                      0
2019[5]                      0
2018[6]                      0
2017[7]                      0
2016[8]                      0
2015[9]                      0
2014[10]                     0
2013[11]                     0
2012[12]                     0
Hubs                         0
Airports (medium hubs)  269953
```



```
data['Traffic_2019/20'] = data['2020[4]']-data['2019[5]']
```

```
data = data.drop(['Airports (large hubs)','Major cities served','Airports (medium hubs)','City served'],axis=1)
```

```
data.head()
```

	Airline	Flight	DayOfWeek	Time	Length	Delay	type	latitude_deg	longitude_deg	elevation_ft	...	2019[5]	2018[6]	2017[7]	2016[8]	2
0	AA	2466	3	20	195	1	large_airport	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	24
1	AA	526	3	360	215	0	large_airport	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	24
2	AA	552	3	360	315	1	large_airport	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	24
3	AA	810	3	385	255	0	large_airport	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	24
4	AA	24	3	425	325	1	large_airport	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	24

5 rows × 30 columns

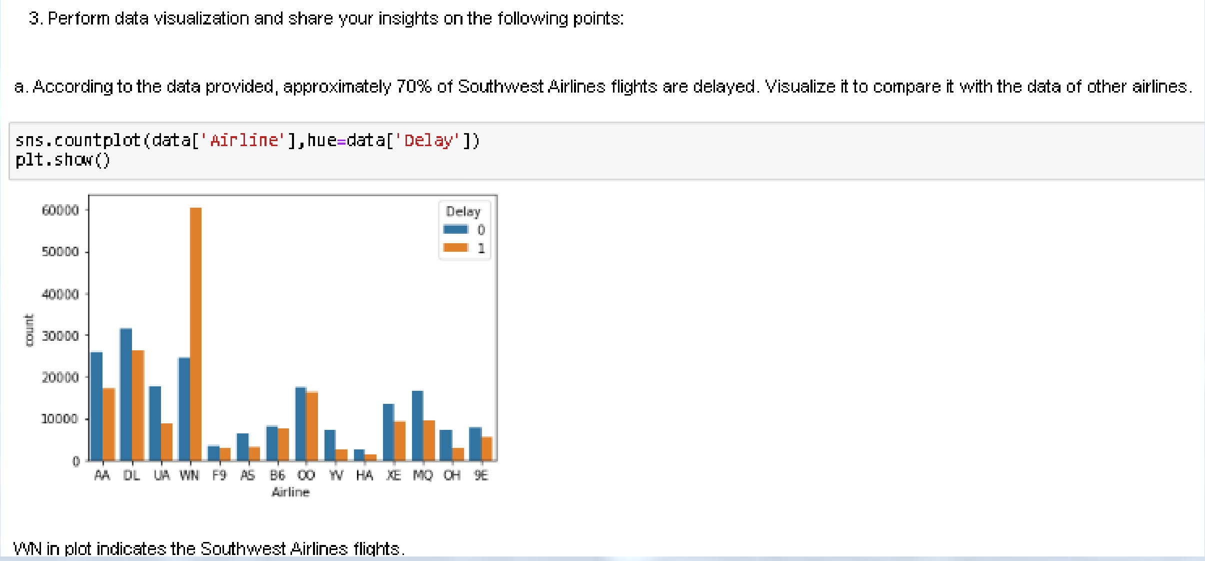
```
data.isnull().sum().sum()
```

```
0
```

Droped these columns because they won't play any role in modeling.

3.Perform data visualization and share your insights on the following points:

a. According to the data provided, approximately 70% of Southwest Airlines flights are delayed. Visualize it to compare it with the data of other airlines.



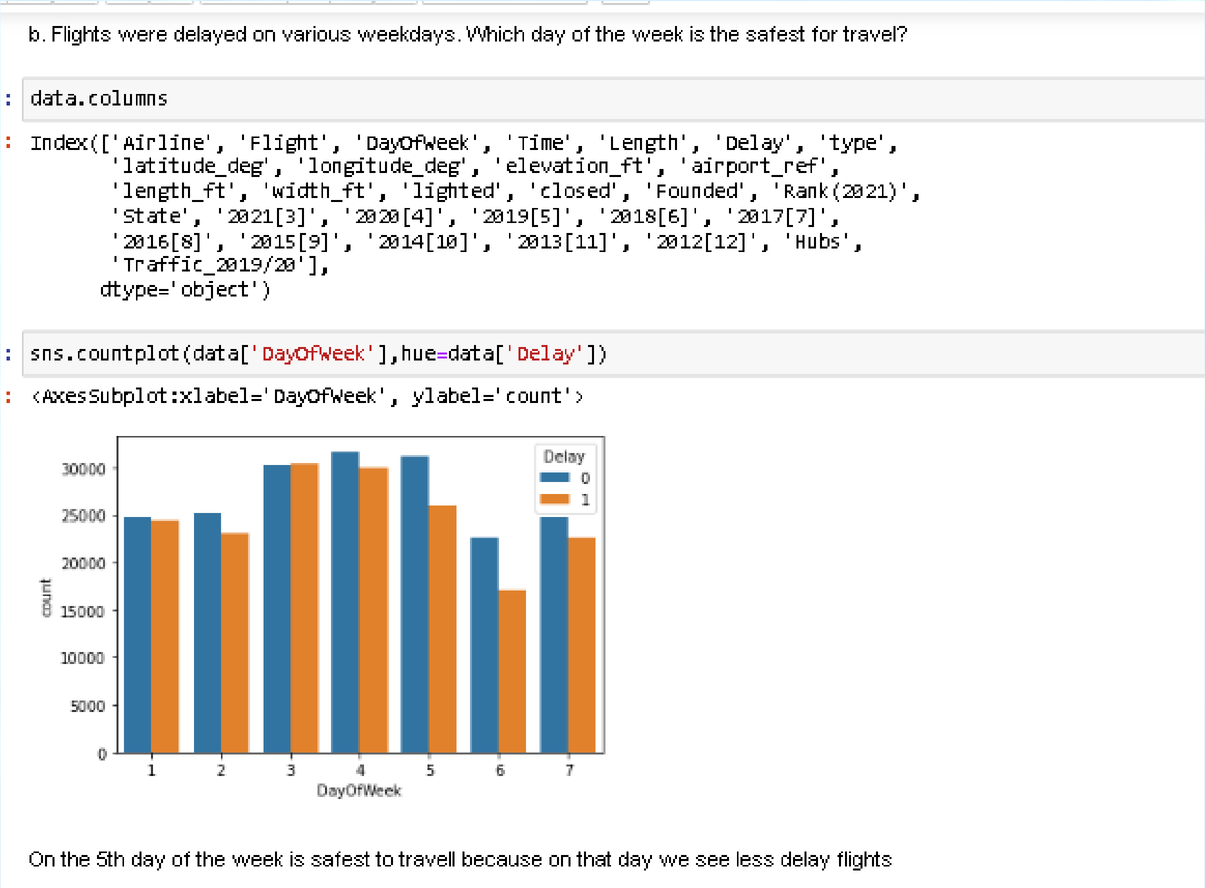
3. Perform data visualization and share your insights on the following points:

a. According to the data provided, approximately 70% of Southwest Airlines flights are delayed. Visualize it to compare it with the data of other airlines.

```
sns.countplot(data['Airline'],hue=data['Delay'])
plt.show()
```

WN in plot indicates the Southwest Airlines flights.

b. Flights were delayed on various weekdays. Which day of the week is the safest for travel?



b. Flights were delayed on various weekdays. Which day of the week is the safest for travel?

```
: data.columns
```

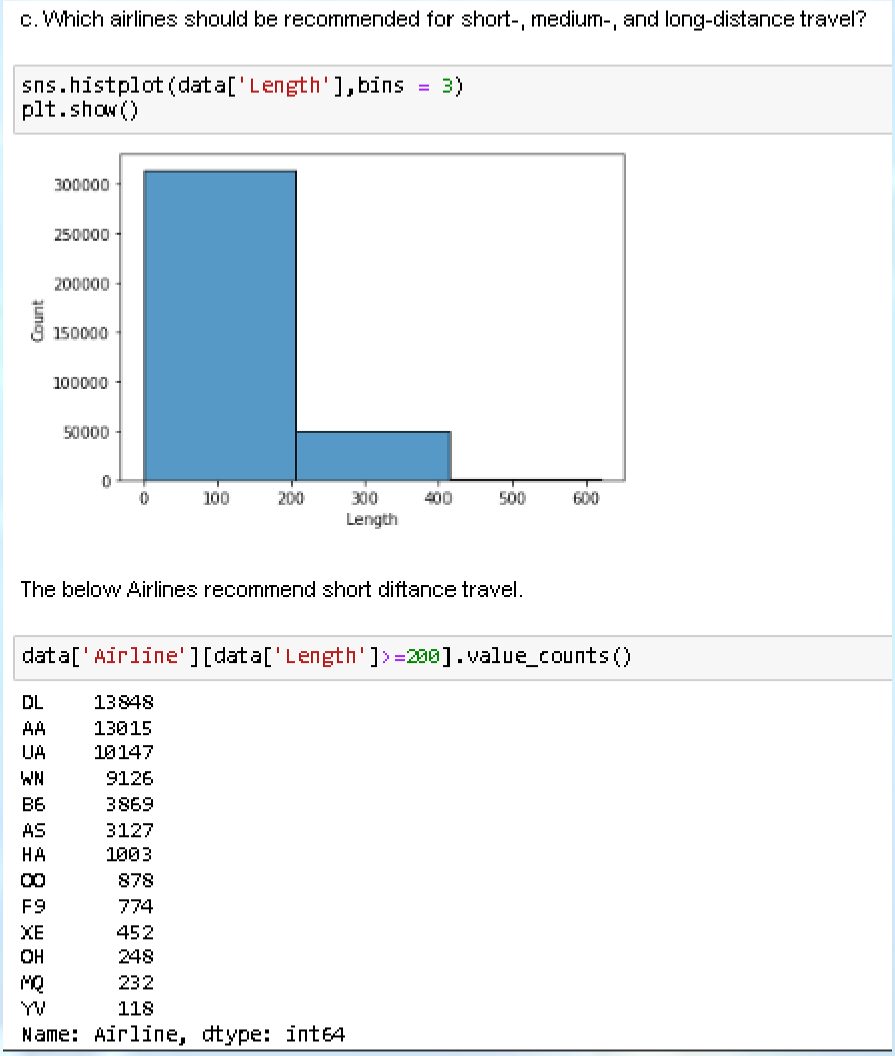
```
: Index(['Airline', 'Flight', 'DayOfWeek', 'Time', 'Length', 'Delay', 'type',
         'latitude_deg', 'longitude_deg', 'elevation_ft', 'airport_ref',
         'length_ft', 'width_ft', 'lighted', 'closed', 'Founded', 'Rank(2021)',
         'State', '2021[3]', '2020[4]', '2019[5]', '2018[6]', '2017[7]',
         '2016[8]', '2015[9]', '2014[10]', '2013[11]', '2012[12]', 'Hubs',
         'Traffic_2019/20'],
        dtype='object')
```

```
: sns.countplot(data['DayOfWeek'],hue=data['Delay'])
```

```
: <AxesSubplot:xlabel='DayOfWeek', ylabel='count'>
```

On the 5th day of the week is safest to travell because on that day we see less delay flights

c. Which airlines should be recommended for short-, medium-, and long-distance travel?



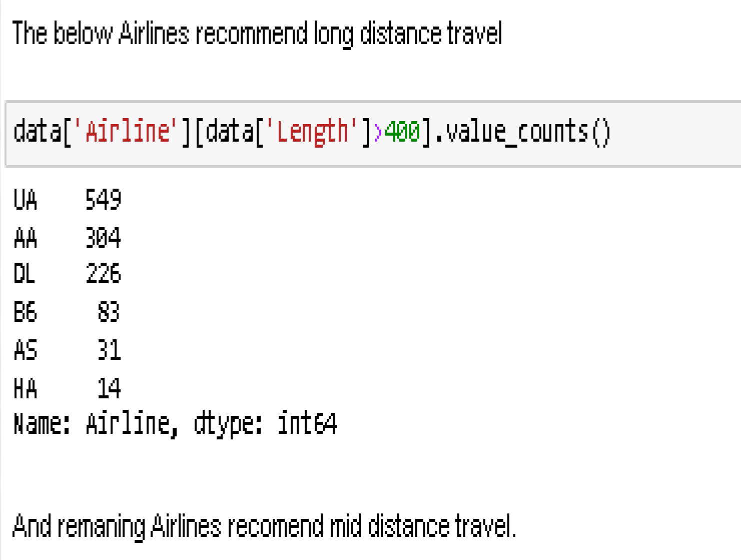
c. Which airlines should be recommended for short-, medium-, and long-distance travel?

```
sns.histplot(data['Length'],bins = 3)
plt.show()
```

The below Airlines recommend short diftance travel.

```
data['Airline'][data['Length']>=200].value_counts()
```

```
DL    13848
AA    13015
UA    10147
WN     9126
B6     3869
AS     3127
HA     1003
OO      878
F9      774
XE      452
OH      248
MQ      232
YV      118
Name: Airline, dtype: int64
```



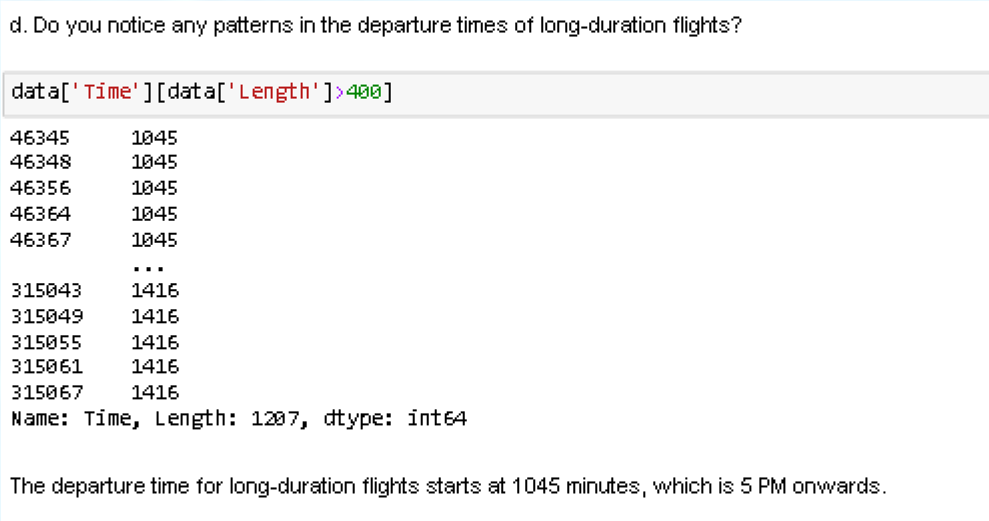
The below Airlines recommend long distance travel

```
data['Airline'][data['Length']>400].value_counts()
```

```
UA    549
AA    304
DL    226
B6     83
AS     31
HA     14
Name: Airline, dtype: int64
```

And remaning Airlines recomend mid distance travel.

d. Do you notice any patterns in the departure times of long-duration flights?



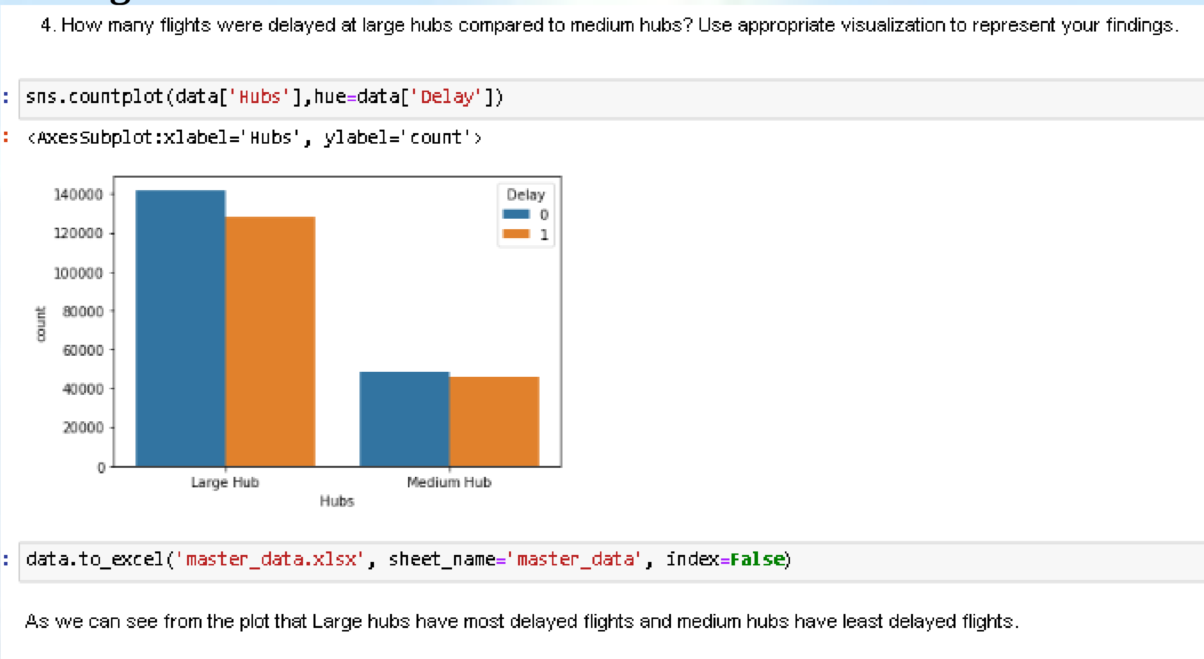
d. Do you notice any patterns in the departure times of long-duration flights?

```
data['Time'][data['Length']>400]
```

```
46345     1045
46348     1045
46356     1045
46364     1045
46367     1045
          ...
315043    1416
315049    1416
315055    1416
315061    1416
315067    1416
Name: Time, Length: 1207, dtype: int64
```

The departure time for long-duration flights starts at 1045 minutes, which is 5 PM onwards.

4. How many flights were delayed at large hubs compared to medium hubs? Use appropriate visualization to represent your findings.



4. How many flights were delayed at large hubs compared to medium hubs? Use appropriate visualization to represent your findings.

```
: sns.countplot(data['Hubs'],hue=data['Delay'])
```

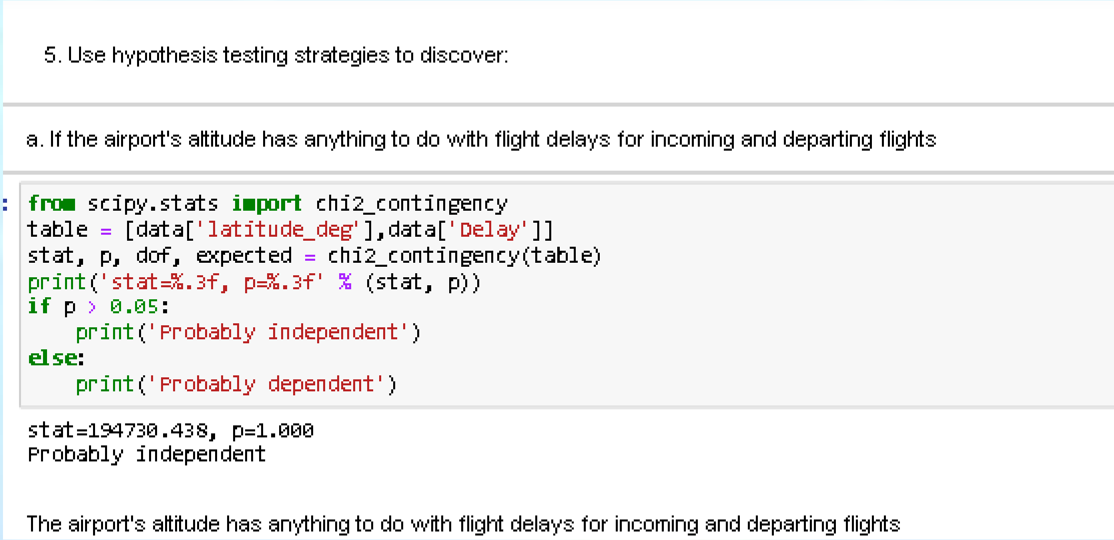
```
: <AxesSubplot:xlabel='Hubs', ylabel='count'>
```

```
: data.to_excel('master_data.xlsx', sheet_name='master_data', index=False)
```

As we can see from the plot that Large hubs have most delayed flights and medium hubs have least delayed flights.

5. Use hypothesis testing strategies to discover:

a. If the airport's altitude has anything to do with flight delays for incoming and departing flights



5. Use hypothesis testing strategies to discover:

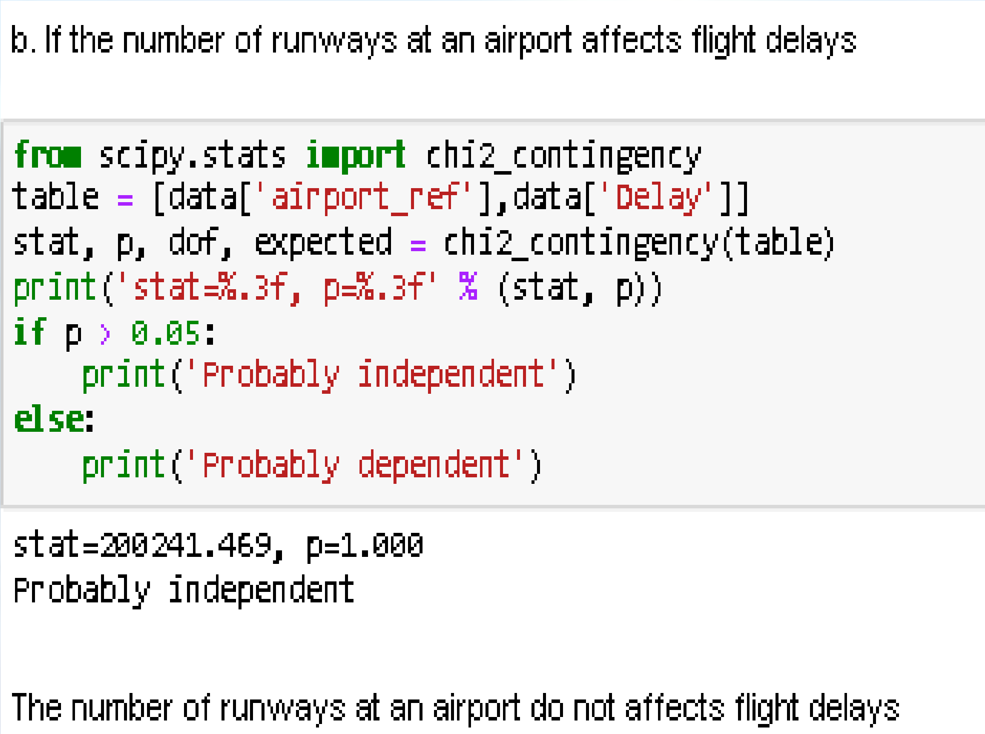
a. If the airport's altitude has anything to do with flight delays for incoming and departing flights

```
from scipy.stats import chi2_contingency
table = [data['latitude_deg'],data['Delay']]
stat, p, dof, expected = chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=194730.438, p=1.000
Probably independent
```

The airport's altitude has anything to do with flight delays for incoming and departing flights

b. If the number of runways at an airport affects flight delays



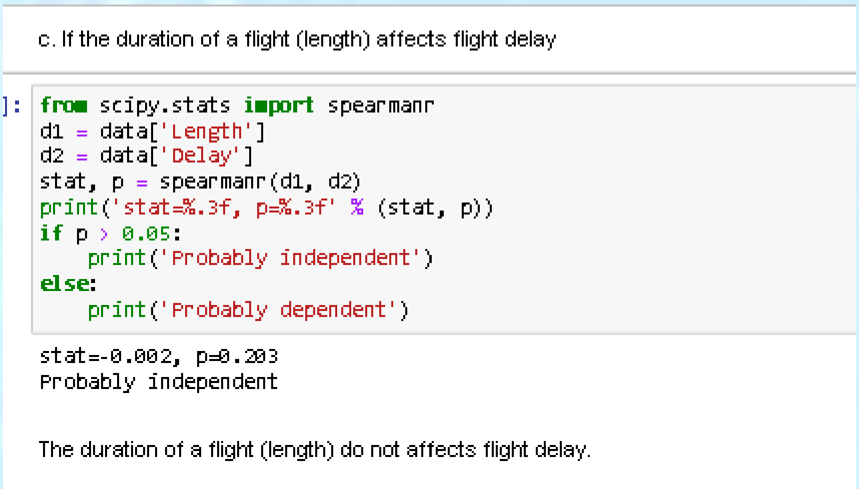
b. If the number of runways at an airport affects flight delays

```
from scipy.stats import chi2_contingency
table = [data['airport_ref'],data['Delay']]
stat, p, dof, expected = chi2_contingency(table)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=200241.469, p=1.000
Probably independent
```

The number of runways at an airport do not affects flight delays

d. If the duration of a flight (length) affects flight delay



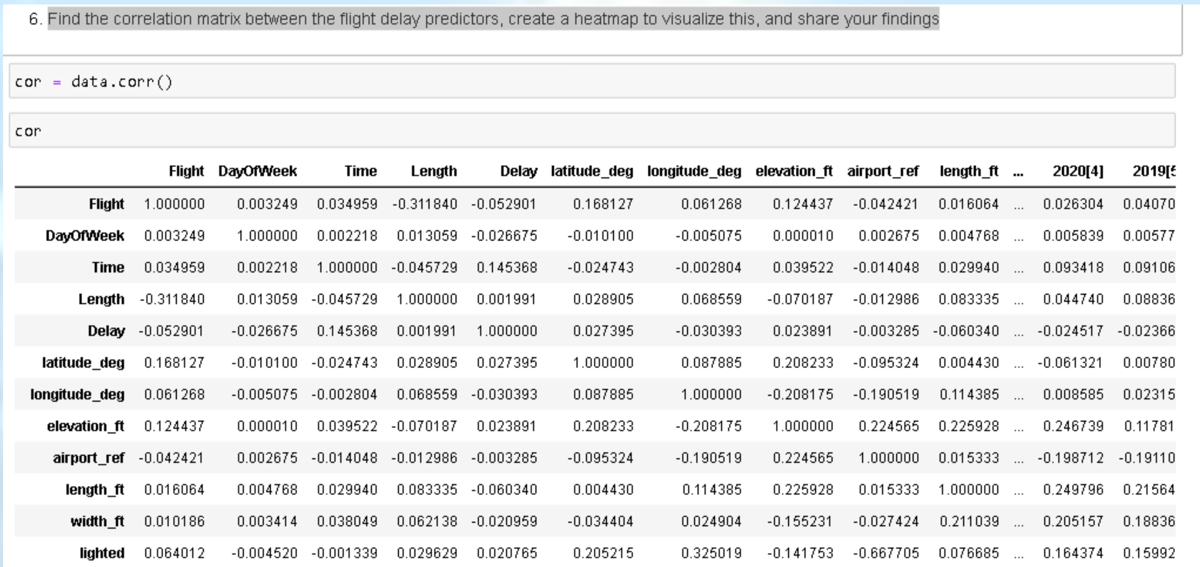
c. If the duration of a flight (length) affects flight delay

```
from scipy.stats import spearmanr
d1 = data['Length']
d2 = data['Delay']
stat, p = spearmanr(d1, d2)
print('stat=%.3f, p=%.3f' % (stat, p))
if p > 0.05:
    print('Probably independent')
else:
    print('Probably dependent')
```

```
stat=-0.002, p=0.203
Probably independent
```

The duration of a flight (length) do not affects flight delay.

6. Find the correlation matrix between the flight delay predictors, create a heatmap to visualize this, and share your findings

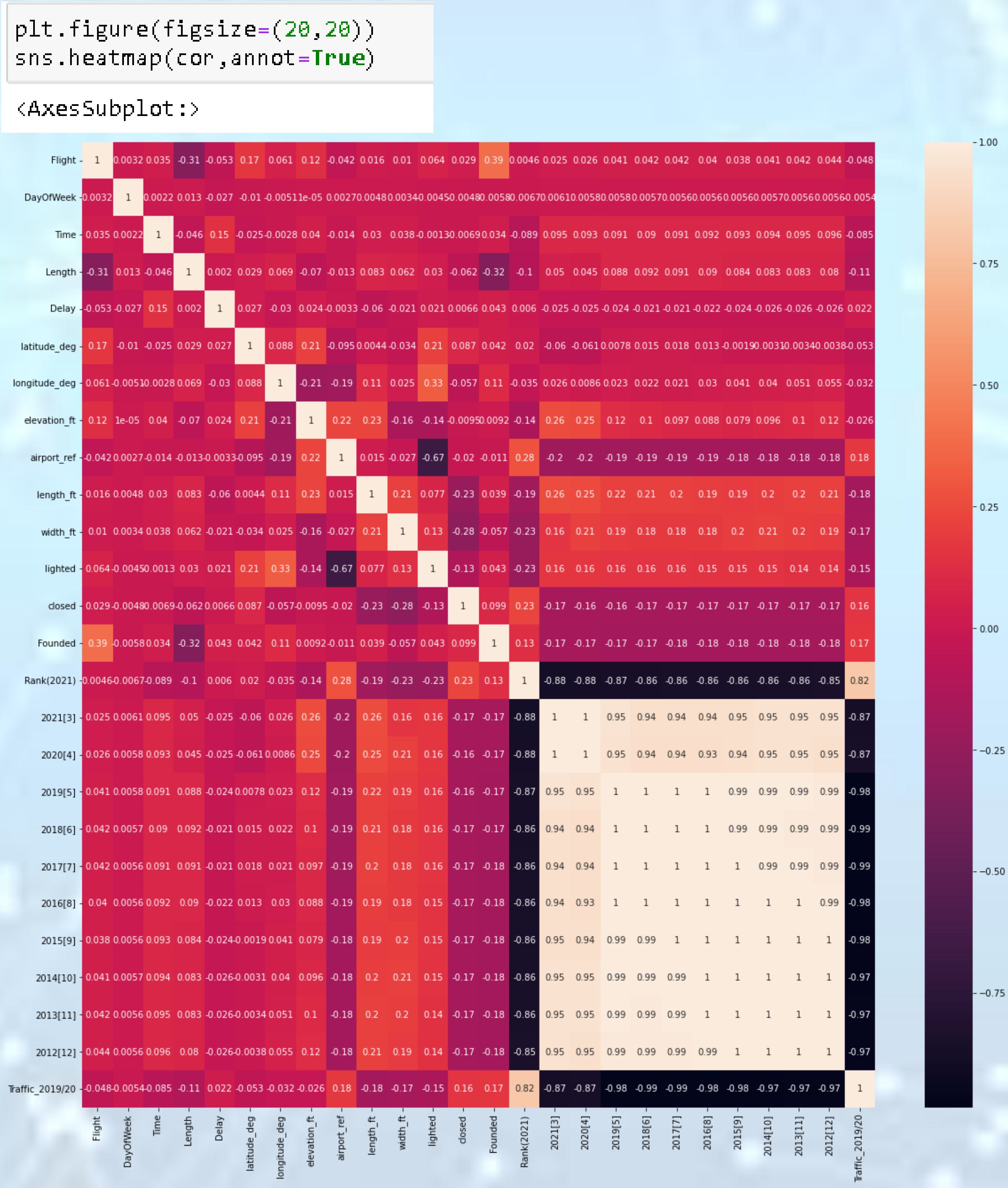


6. Find the correlation matrix between the flight delay predictors, create a heatmap to visualize this, and share your findings

```
cor = data.corr()
```

```
cor
```

	Flight	DayOfWeek	Time	Length	Delay	latitude_deg	longitude_deg	elevation_ft	airport_ref	length_ft	...	2020[4]	2019[5
Flight	1.000000	0.003249	0.034959	-0.311840	-0.052901	0.168127	0.061268	0.124437	-0.042421	0.016064	...	0.026304	0.04070
DayOfWeek	0.003249	1.000000	0.002218	0.013059	-0.026675	-0.010100	-0.005075	0.000010	0.002675	0.004768	...	0.005839	0.00577
Time	0.034959	0.002218	1.000000	-0.045729	0.145368	-0.024743	-0.002804	0.039522	-0.014048	0.029940	...	0.093418	0.09106
Length	-0.311840	0.013059	-0.045729	1.000000	0.001991	0.028905	0.068559	-0.070187	-0.012986	0.083335	...	0.044740	0.08836
Delay	-0.052901	-0.026675	0.145368	0.001991	1.000000	0.027395	-0.030393	0.023891	-0.003285	-0.060340	...	-0.024517	-0.02366
latitude_deg	0.168127	-0.010100	-0.024743	0.028905	0.027395	1.000000	0.087885	0.208233	-0.095324	0.004430	...	-0.061321	0.00780
longitude_deg	0.061268	-0.005075	-0.002804	0.068559	-0.030393	0.087885	1.000000	-0.208175	-0.190519	0.114385	...	0.008585	0.02315
elevation_ft	0.124437	0.000010	0.039522	-0.070187	0.023891	0.208233	-0.208175	1.000000	0.224565	0.225928	...	0.246739	0.11781
airport_ref	-0.042421	0.002675	-0.014048	-0.012986	-0.003285	-0.095324	-0.190519	0.224565	1.000000	0.015333	...	-0.198712	-0.19110
length_ft	0.016064	0.004768	0.029940	0.083335	-0.060340	0.004430	0.114385	0.225928	0.015333	1.000000	...	0.249796	0.21564
width_ft	0.010186	0.003414	0.038049	0.062138	-0.020959	-0.034404	0.024904	-0.155231	-0.027424	0.211039	...	0.205157	0.18836
lighted	0.064012	-0.004520	-0.001339	0.029629	0.020765	0.205215	0.325019	-0.141753	-0.667705	0.076685	...	0.164374	0.15992



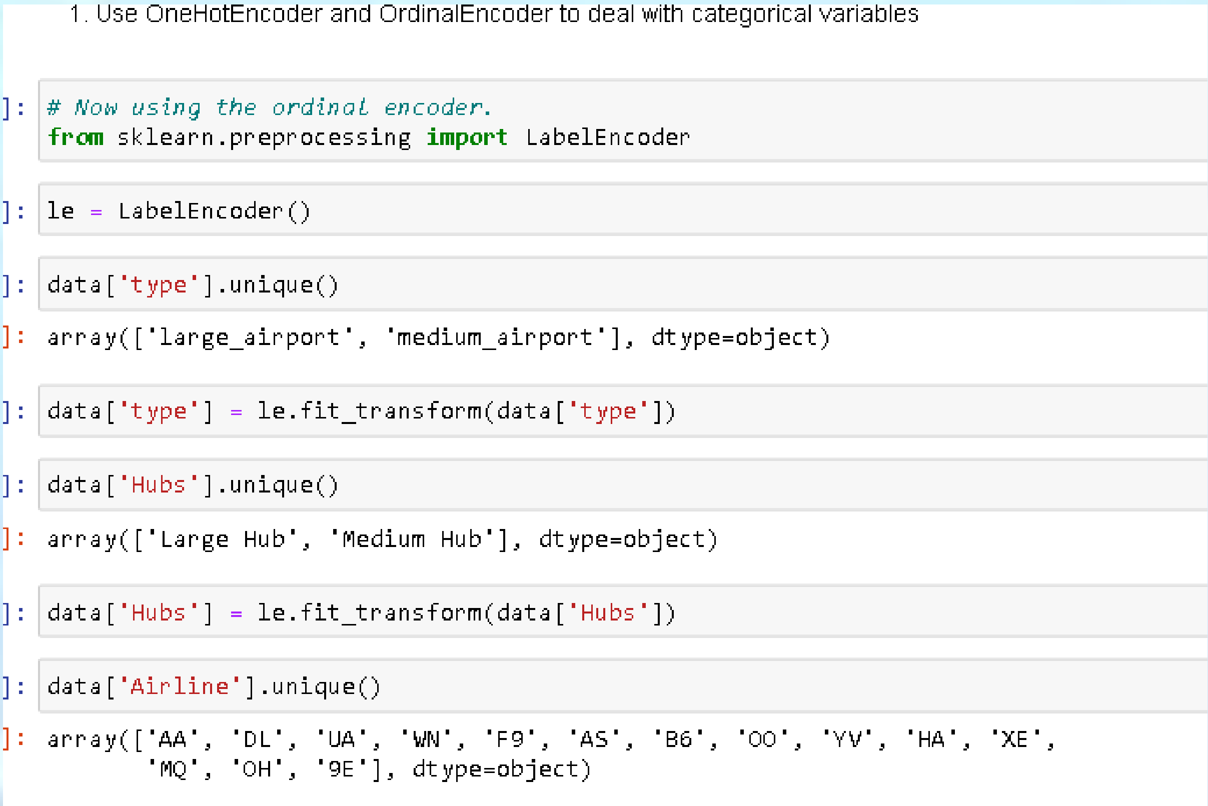
```
plt.figure(figsize=(20,20))
sns.heatmap(cor,annot=True)
```

```
<AxesSubplot:>
```

	Flight	DayOfWeek	Time	Length	Delay	latitude_deg	longitude_deg	elevation_ft	airport_ref	length_ft	width_ft	lighted	closed	Founded	Rank(2021)	2021[3]	2020[4]	2019[5]	2018[6]	2017[7]	2016[8]	2015[9]	2014[10]	2013[11]	2012[12]	Traffic_2019/20
Flight	1	0.0032	0.035	-0.31	-0.053	0.17	0.061	0.12	-0.042	0.016	0.01	0.064	0.029	0.39	0.0046	0.025	0.026	0.041	0.042	0.042	0.04	0.038	0.041	0.042	0.044	-0.048
DayOfWeek	0.0032	1	0.0022	0.013	-0.027	-0.01	-0.0051	1e-05	0.0027	0.0048	0.0034	-0.0045	-0.0048	-0.0058	-0.0067	0.0061	0.0058	0.0058	0.0057	0.0056	0.0056	0.0056	0.0057	0.0056	0.0056	-0.0054
Time	0.035	0.0022	1	-0.046	0.15	-0.025	-0.0028	0.04	-0.014	0.03	0.038	-0.0013	-0.0069	0.034	-0.089	0.095	0.093	0.091	0.09	0.091	0.092	0.093	0.094	0.095	0.096	-0.085
Length	-0.31	0.013	-0.046	1	0.002	0.029	0.069	-0.07	-0.013	0.083	0.062	0.03	-0.062	-0.32	-0.1	0.05	0.045	0.088	0.092	0.091	0.09	0.084	0.083	0.083	0.08	-0.11
Delay	-0.053	-0.027	0.15	0.002	1	0.027	-0.03	0.024	-0.0033	-0.06	-0.021	0.021	0.0066	0.043	0.006	-0.025	-0.025	-0.024	-0.021	-0.021	-0.022	-0.024	-0.026	-0.026	-0.026	0.022
latitude_deg	0.17	-0.01	-0.025	0.029	0.027	1	0.088	0.21	-0.095	0.0044	-0.034	0.21	0.087	0.042	0.02	-0.06	-0.061	0.0078	0.015	0.018	0.013	-0.0019	-0.0031	-0.0034	-0.0038	-0.053
longitude_deg	0.061	-0.0051	-0.0028	0.069	-0.03	0.088	1	-0.21	-0.19	0.11	0.025	0.33	-0.057	0.11	-0.035	0.026	0.0086	0.023	0.022	0.021	0.03	0.041	0.04	0.051	0.055	-0.032
elevation_ft	0.12	1e-05	0.04	-0.07	0.024	0.21	-0.21	1	0.22	0.23	-0.16	-0.14	-0.0095	0.0092	-0.14	0.26	0.25	0.12	0.1	0.097	0.088	0.079	0.096	0.1	0.12	-0.026
airport_ref	-0.042	0.0027	-0.014	-0.013	-0.0033	-0.095	-0.19	0.22	1	0.015	-0.027	-0.67	-0.02	-0.011	0.28	-0.2	-0.2	-0.19	-0.19	-0.19	-0.19	-0.18	-0.18	-0.18	-0.18	0.18
length_ft	0.016	0.0048	0.03	0.083	-0.06	0.0044	0.11	0.23	0.015	1	0.21	0.077	-0.23	0.039	-0.19	0.26	0.25	0.22	0.21	0.2	0.19	0.19	0.2	0.2	0.21	-0.18
width_ft	0.01	0.0034	0.038	0.062	-0.021	-0.034	0.025	-0.16	-0.027	0.21	1	0.13	-0.28	-0.057	-0.23	0.16	0.21	0.19	0.18	0.18	0.18	0.2	0.21	0.2	0.19	-0.17
lighted	0.064	-0.0045	-0.0013	0.03	0.021	0.21	0.33	-0.14	-0.67	0.077	0.13	1	-0.13	0.043	-0.23	0.16	0.16	0.16	0.16	0.16	0.15	0.15	0.15	0.14	0.14	-0.15
closed	0.029	-0.0048	-0.0069	-0.062	0.0066	0.087	-0.057	-0.0095	-0.02	-0.23	-0.28	-0.13	1	0.099	0.23	-0.17	-0.16	-0.16	-0.17	-0.17	-0.17	-0.17	-0.17	-0.17	-0.17	0.16
Founded	0.39	-0.0058	0.034	-0.32	0.043	0.042	0.11	0.0092	-0.011	0.039	-0.057	0.043	0.099	1	0.13	-0.17	-0.17	-0.17	-0.17	-0.18	-0.18	-0.18	-0.18	-0.18	-0.18	0.17
Rank(2021)	0.0046	-0.0067	-0.089	-0.1	0.006	0.02	-0.035	-0.14	0.28	-0.19	-0.23	-0.23	0.23	0.13	1	-0.88	-0.88	-0.87	-0.86	-0.86	-0.86	-0.86	-0.86	-0.86	-0.85	0.82
2021[3]	0.025	0.0061	0.095	0.05	-0.025	-0.06	0.026	0.26	-0.2	0.26	0.16	0.16	-0.17	-0.17	-0.88	1	1	0.95	0.94	0.94	0.94	0.95	0.95	0.95	0.95	-0.87
2020[4]	0.026	0.0058	0.093	0.045	-0.025	-0.061	0.0086	0.25	-0.2	0.25	0.21	0.16	-0.16	-0.17	-0.88	1	1	0.95	0.94	0.94	0.93	0.94	0.95	0.95	0.95	-0.87
2019[5]	0.041	0.0058	0.091	0.088	-0.024	0.0078	0.023	0.12	-0.19	0.22	0.19	0.16	-0.16	-0.17	-0.87	0.95	0.95	1	1	1	1	0.99	0.99	0.99	0.99	-0.98
2018[6]	0.042	0.0057	0.09	0.092	-0.021	0.015	0.022	0.1	-0.19	0.21	0.18	0.16	-0.17	-0.17	-0.86	0.94	0.94	1	1	1	1	0.99	0.99	0.99	0.99	-0.99
2017[7]	0.042	0.0056	0.091	0.091	-0.021	0.018	0.021	0.097	-0.19	0.2	0.18	0.16	-0.17	-0.18	-0.86	0.94	0.94	1	1	1	1	1	0.99	0.99	0.99	-0.99
2016[8]	0.04	0.0056	0.092	0.09	-0.022	0.013	0.03	0.088	-0.19	0.19	0.18	0.15	-0.17	-0.18	-0.86	0.94	0.93	1	1	1	1	1	1	1	0.99	-0.98
2015[9]	0.038	0.0056	0.093	0.084	-0.024	-0.0019	0.041	0.079	-0.18	0.19	0.2	0.15	-0.17	-0.18	-0.86	0.95	0.94	0.99	0.99	1	1	1	1	1	1	-0.98
2014[10]	0.041	0.0057	0.094	0.083	-0.026	-0.0031	0.04	0.096	-0.18	0.2	0.21	0.15	-0.17	-0.18	-0.86	0.95	0.95	0.99	0.99	0.99	1	1	1	1	1	-0.97
2013[11]	0.042	0.0056	0.095	0.083	-0.026	-0.0034	0.051	0.1	-0.18	0.2	0.2	0.14	-0.17	-0.18	-0.86	0.95	0.95	0.99	0.99	0.99	1	1	1	1	1	-0.97
2012[12]	0.044	0.0056	0.096	0.08	-0.026	-0.0038	0.055	0.12	-0.18	0.21	0.19	0.14	-0.17	-0.18	-0.85	0.95	0.95	0.99	0.99	0.99	0.99	1	1	1	1	-0.97
Traffic_2019/20	-0.048	-0.0054	-0.085	-0.11	0.022	-0.053	-0.032	-0.026	0.18	-0.18	-0.17	-0.15	0.16	0.17	0.82	-0.87	-0.87	-0.98	-0.99	-0.99	-0.98	-0.98	-0.97	-0.97	-0.97	1

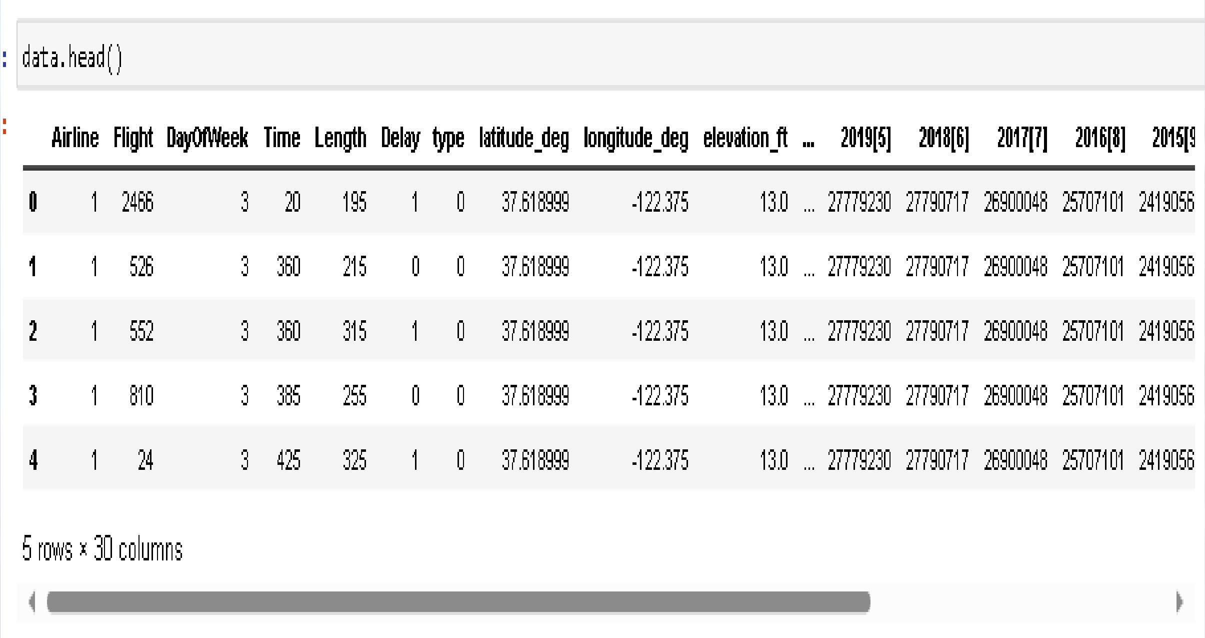
Machine learning

1. Use OneHotEncoder and OrdinalEncoder to deal with categorical variables



1. Use OneHotEncoder and OrdinalEncoder to deal with categorical variables

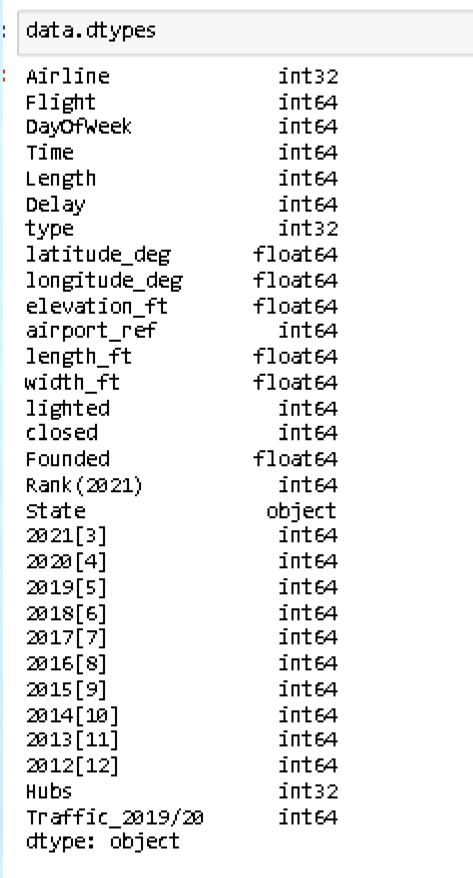
```
]: # Now using the ordinal encoder.
   from sklearn.preprocessing import LabelEncoder

]: le = LabelEncoder()

]: data['type'].unique()

]: array(['large_airport', 'medium_airport'], dtype=object)

]: data['type'] = le.fit_transform(data['type'])

]: data['Hubs'].unique()

]: array(['Large Hub', 'Medium Hub'], dtype=object)

]: data['Hubs'] = le.fit_transform(data['Hubs'])

]: data['Airline'].unique()

]: array(['AA', 'DL', 'UA', 'WN', 'F9', 'AS', 'B6', 'OO', 'YV', 'HA', 'XE',
          'MQ', 'OH', '9E'], dtype=object)
```



```
: data.head()
```

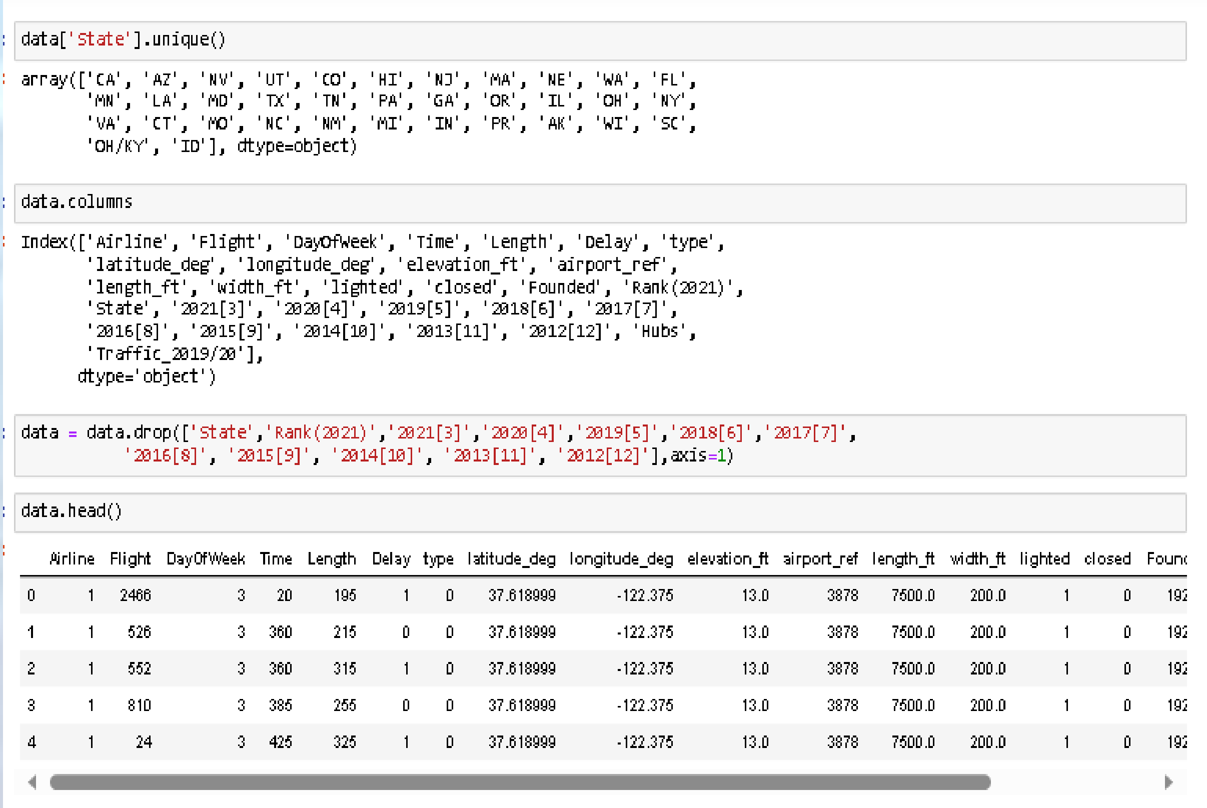
	Airline	Flight	DayOfWeek	Time	Length	Delay	type	latitude_deg	longitude_deg	elevation_ft	...	2019[5]	2018[6]	2017[7]	2016[8]	2015[9
0	1	2466	3	20	195	1	0	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	2419056
1	1	526	3	360	215	0	0	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	2419056
2	1	552	3	360	315	1	0	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	2419056
3	1	810	3	385	255	0	0	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	2419056
4	1	24	3	425	325	1	0	37.618999	-122.375	13.0	...	27779230	27790717	26900048	25707101	2419056

5 rows × 30 columns



```
data.dtypes
```

```
Airline               int32
Flight                int64
DayOfWeek             int64
Time                  int64
Length                int64
Delay                 int64
type                  int32
latitude_deg        float64
longitude_deg       float64
elevation_ft        float64
airport_ref           int64
length_ft           float64
width_ft            float64
lighted               int64
closed                int64
Founded             float64
Rank(2021)            int64
State                object
2021[3]               int64
2020[4]               int64
2019[5]               int64
2018[6]               int64
2017[7]               int64
2016[8]               int64
2015[9]               int64
2014[10]              int64
2013[11]              int64
2012[12]              int64
Hubs                  int32
Traffic_2019/20       int64
dtype: object
```



```
data['State'].unique()
```

```
array(['CA', 'AZ', 'NV', 'UT', 'CO', 'HI', 'NJ', 'MA', 'NE', 'WA', 'FL',
       'MN', 'LA', 'MD', 'TX', 'TN', 'PA', 'GA', 'OR', 'IL', 'OH', 'NY',
       'VA', 'CT', 'MO', 'NC', 'NM', 'MI', 'IN', 'PR', 'AK', 'WI', 'SC',
       'OH/KY', 'ID'], dtype=object)
```

```
data.columns
```

```
Index(['Airline', 'Flight', 'DayOfWeek', 'Time', 'Length', 'Delay', 'type',
       'latitude_deg', 'longitude_deg', 'elevation_ft', 'airport_ref',
       'length_ft', 'width_ft', 'lighted', 'closed', 'Founded', 'Rank(2021)',
       'State', '2021[3]', '2020[4]', '2019[5]', '2018[6]', '2017[7]',
       '2016[8]', '2015[9]', '2014[10]', '2013[11]', '2012[12]', 'Hubs',
       'Traffic_2019/20'],
      dtype='object')
```

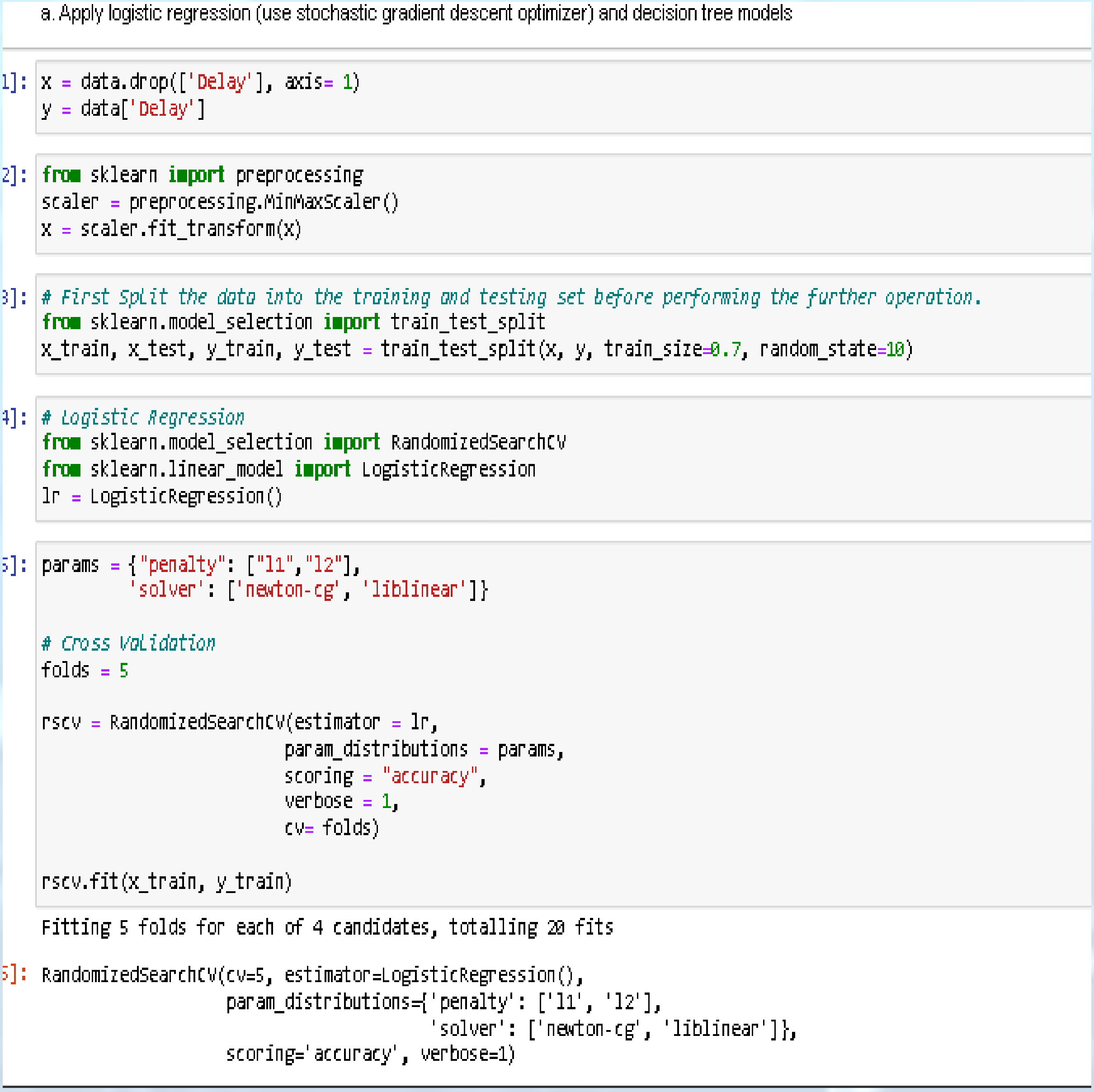
```
data = data.drop(['State','Rank(2021)','2021[3]','2020[4]','2019[5]','2018[6]','2017[7]',
            '2016[8]', '2015[9]', '2014[10]', '2013[11]', '2012[12]'],axis=1)
```

```
data.head()
```

	Airline	Flight	DayOfWeek	Time	Length	Delay	type	latitude_deg	longitude_deg	elevation_ft	airport_ref	length_ft	width_ft	lighted	closed	Founc
0	1	2466	3	20	195	1	0	37.618999	-122.375	13.0	3878	7500.0	200.0	1	0	192
1	1	526	3	360	215	0	0	37.618999	-122.375	13.0	3878	7500.0	200.0	1	0	192
2	1	552	3	360	315	1	0	37.618999	-122.375	13.0	3878	7500.0	200.0	1	0	192
3	1	810	3	385	255	0	0	37.618999	-122.375	13.0	3878	7500.0	200.0	1	0	192
4	1	24	3	425	325	1	0	37.618999	-122.375	13.0	3878	7500.0	200.0	1	0	192

2. Perform the following model building steps:

a. Apply logistic regression (use stochastic gradient descent optimizer) and decision tree models



a. Apply logistic regression (use stochastic gradient descent optimizer) and decision tree models

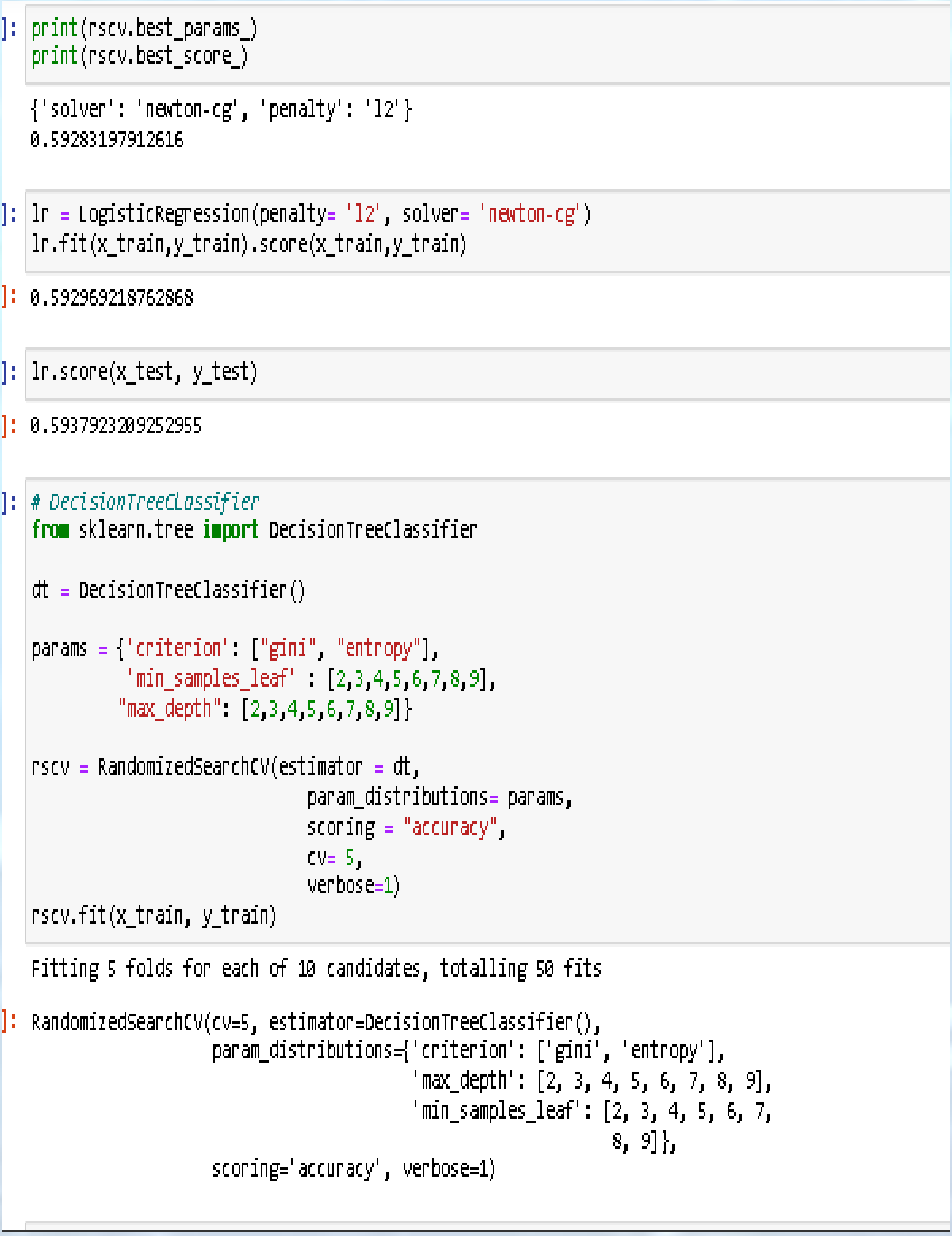
```
1]: x = data.drop(['Delay'], axis= 1)
    y = data['Delay']
```

```
2]: from sklearn import preprocessing
    scaler = preprocessing.MinMaxScaler()
    x = scaler.fit_transform(x)
```

```
3]: # First Split the data into the training and testing set before performing the further operation.
    from sklearn.model_selection import train_test_split
    x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.7, random_state=10)
```

```
4]: # Logistic Regression
    from sklearn.model_selection import RandomizedSearchCV
    from sklearn.linear_model import LogisticRegression
    lr = LogisticRegression()
```

```
5]: params = {"penalty": ["l1","l2"],
             'solver': ['newton-cg', 'liblinear']}

    # Cross Validation
    folds = 5

    rscv = RandomizedSearchCV(estimator = lr,
                              param_distributions = params,
                              scoring = "accuracy",
                              verbose = 1,
                              cv= folds)

    rscv.fit(x_train, y_train)

    Fitting 5 folds for each of 4 candidates, totalling 20 fits

5]: RandomizedSearchCV(cv=5, estimator=LogisticRegression(),
                       param_distributions={'penalty': ['l1', 'l2'],
                                            'solver': ['newton-cg', 'liblinear']},
                       scoring='accuracy', verbose=1)
```



```
print(rscv.best_params_)
print(rscv.best_score_)
```

```
{'solver': 'newton-cg', 'penalty': 'l2'}
0.59283197912616
```

```
lr = LogisticRegression(penalty= 'l2', solver= 'newton-cg')
lr.fit(x_train,y_train).score(x_train,y_train)
```

```
0.592969218762868
```

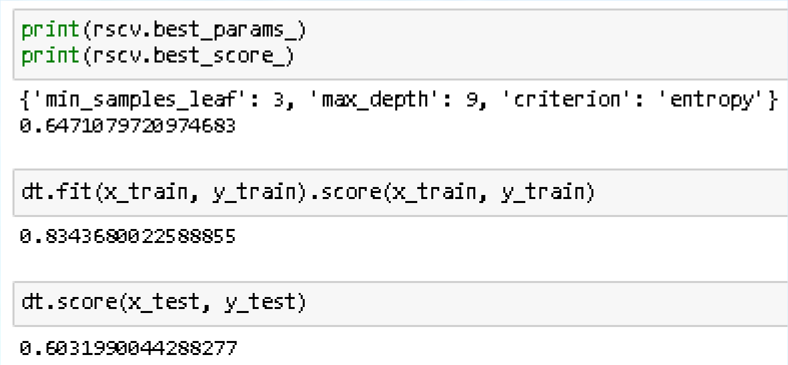
```
lr.score(x_test, y_test)
```

```
0.5937923209252955
```

```
# DecisionTreeClassifier
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()

params = {'criterion': ["gini", "entropy"],
          'min_samples_leaf' : [2,3,4,5,6,7,8,9],
         "max_depth": [2,3,4,5,6,7,8,9]}

rscv = RandomizedSearchCV(estimator = dt,
                          param_distributions= params,
                          scoring = "accuracy",
                          cv= 5,
                          verbose=1)
rscv.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

```
RandomizedSearchCV(cv=5, estimator=DecisionTreeClassifier(),
                   param_distributions={'criterion': ['gini', 'entropy'],
                                        'max_depth': [2, 3, 4, 5, 6, 7, 8, 9],
                                        'min_samples_leaf': [2, 3, 4, 5, 6, 7,
                                                             8, 9]},
                   scoring='accuracy', verbose=1)
```



```
print(rscv.best_params_)
print(rscv.best_score_)
```

```
{'min_samples_leaf': 3, 'max_depth': 9, 'criterion': 'entropy'}
0.6471079720974683
```

```
dt.fit(x_train, y_train).score(x_train, y_train)
```

```
0.8343680022588855
```

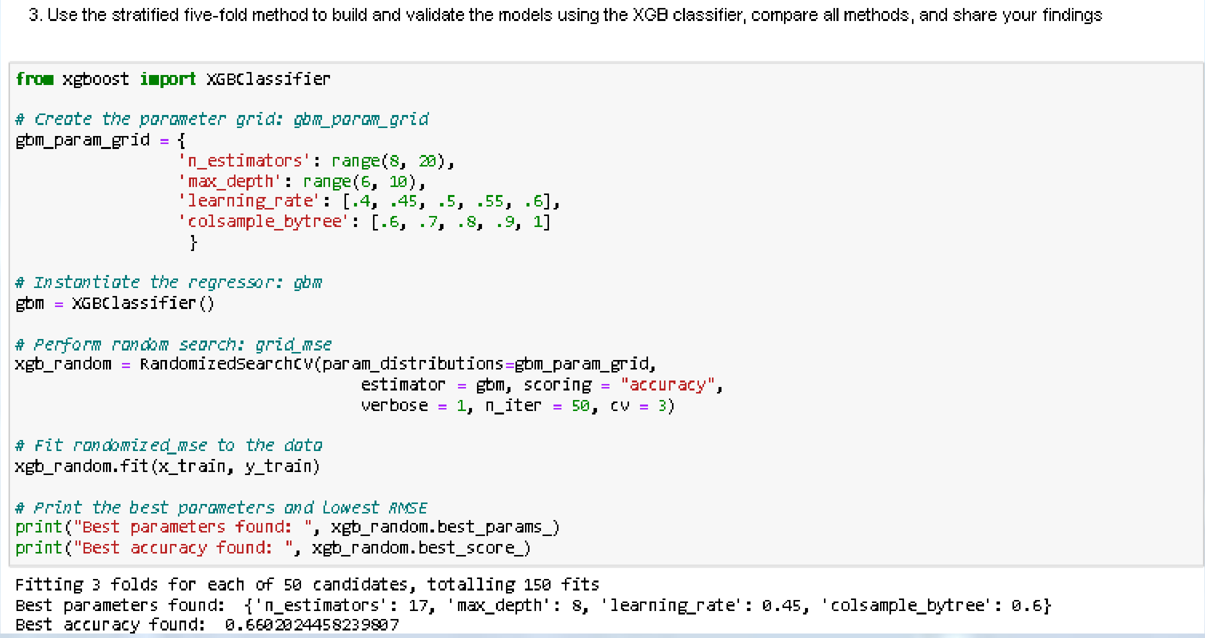
```
dt.score(x_test, y_test)
```

```
0.6031990044288277
```

****g. Compare the results of logistic regression and decision tree classifier**

After comparing the results, we can conclude that Decission Tree Algorithm is optimal

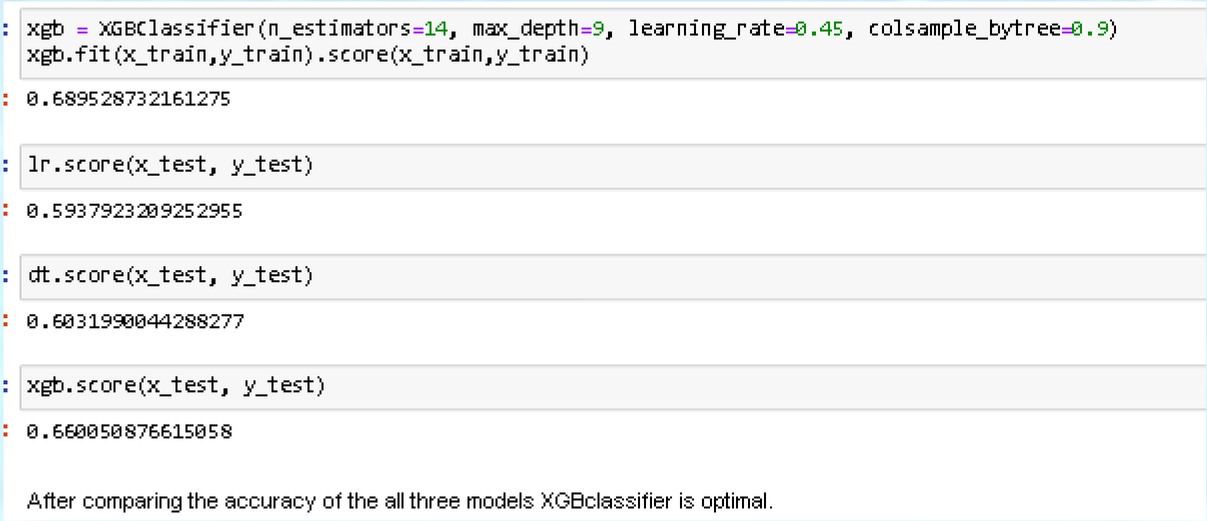
3. Use the stratified five-fold method to build and validate the models using the XGB classifier, compare all methods, and share your findings



3. Use the stratified five-fold method to build and validate the models using the XGB classifier, compare all methods, and share your findings

```
from xgboost import XGBClassifier

# Create the parameter grid: gbm_param_grid
gbm_param_grid = {
                 'n_estimators': range(8, 20),
                 'max_depth': range(6, 10),
                 'learning_rate': [.4, .45, .5, .55, .6],
                 'colsample_bytree': [.6, .7, .8, .9, 1]
                  }

# Instantiate the regressor: gbm
gbm = XGBClassifier()

# Perform random search: grid_mse
xgb_random = RandomizedSearchCV(param_distributions=gbm_param_grid,
                                    estimator = gbm, scoring = "accuracy",
                                    verbose = 1, n_iter = 50, cv = 3)

# Fit randomized_mse to the data
xgb_random.fit(x_train, y_train)

# Print the best parameters and lowest RMSE
print("Best parameters found: ", xgb_random.best_params_)
print("Best accuracy found: ", xgb_random.best_score_)
```

```
Fitting 3 folds for each of 50 candidates, totalling 150 fits
Best parameters found:  {'n_estimators': 17, 'max_depth': 8, 'learning_rate': 0.45, 'colsample_bytree': 0.6}
Best accuracy found:  0.6602024458239807
```



```
xgb = XGBClassifier(n_estimators=14, max_depth=9, learning_rate=0.45, colsample_bytree=0.9)
xgb.fit(x_train,y_train).score(x_train,y_train)
```

```
0.689528732161275
```

```
lr.score(x_test, y_test)
```

```
0.5937923209252955
```

```
dt.score(x_test, y_test)
```

```
0.6031990044288277
```

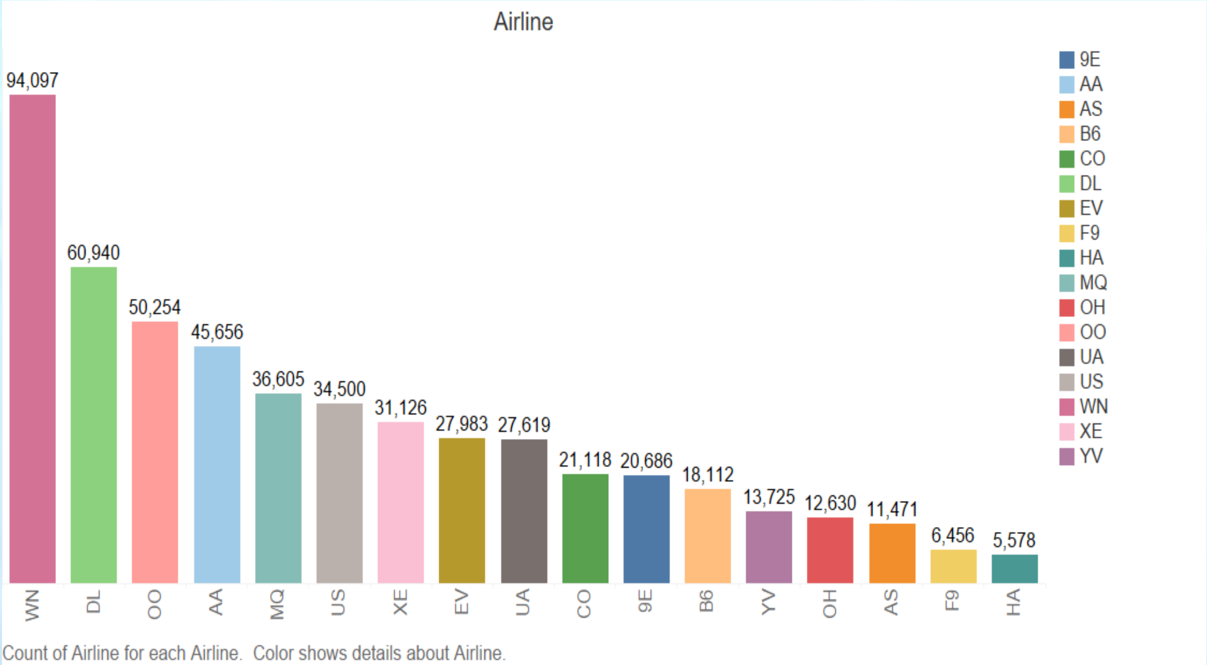
```
xgb.score(x_test, y_test)
```

```
0.660050876615058
```

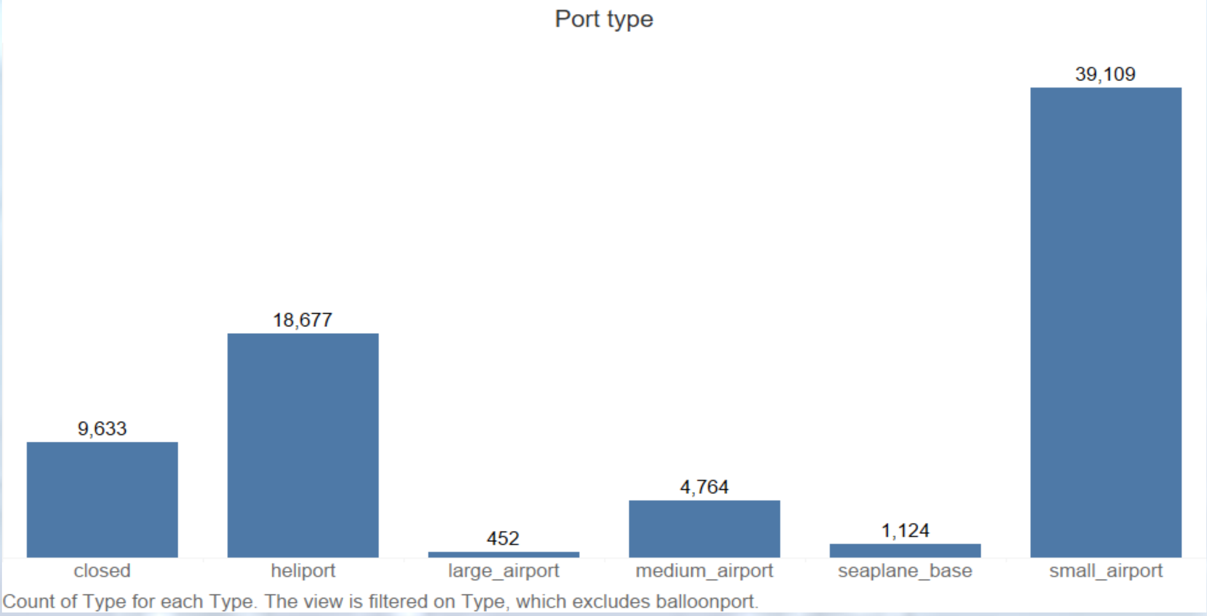
After comparing the accuracy of the all three models XGBclassifier is optimal.

Tableau:

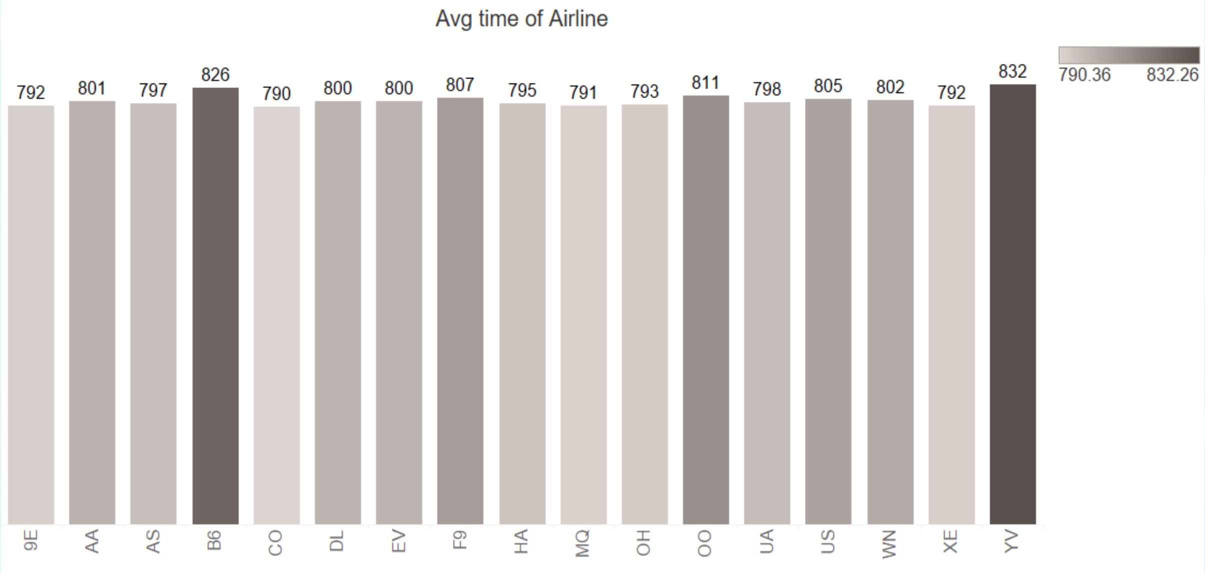
Airline:



Port type:

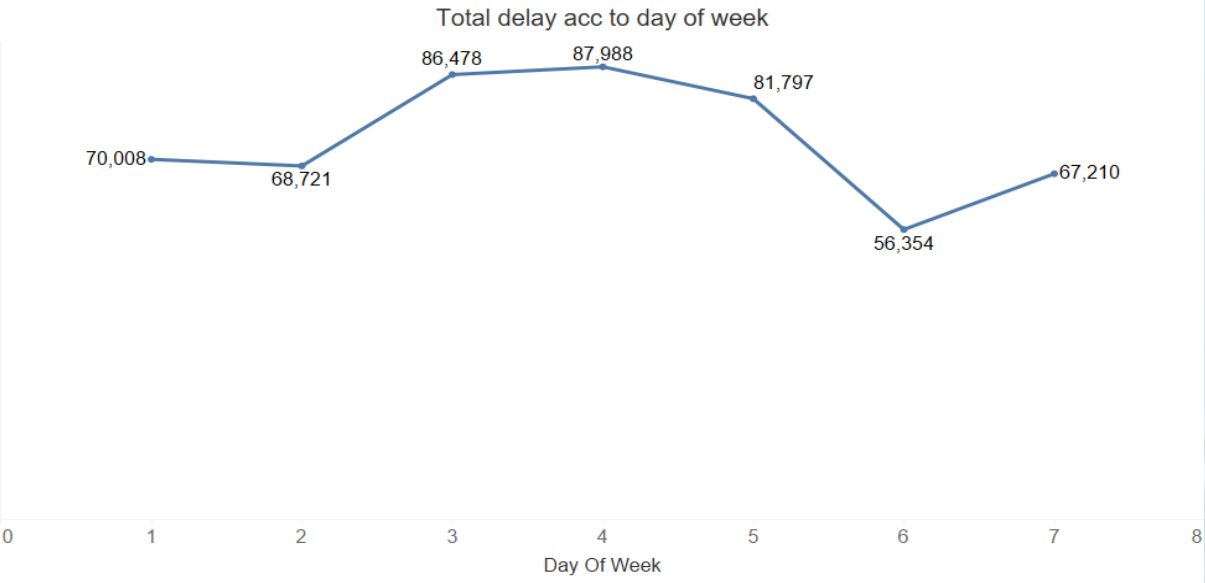


Avg time of Airline:



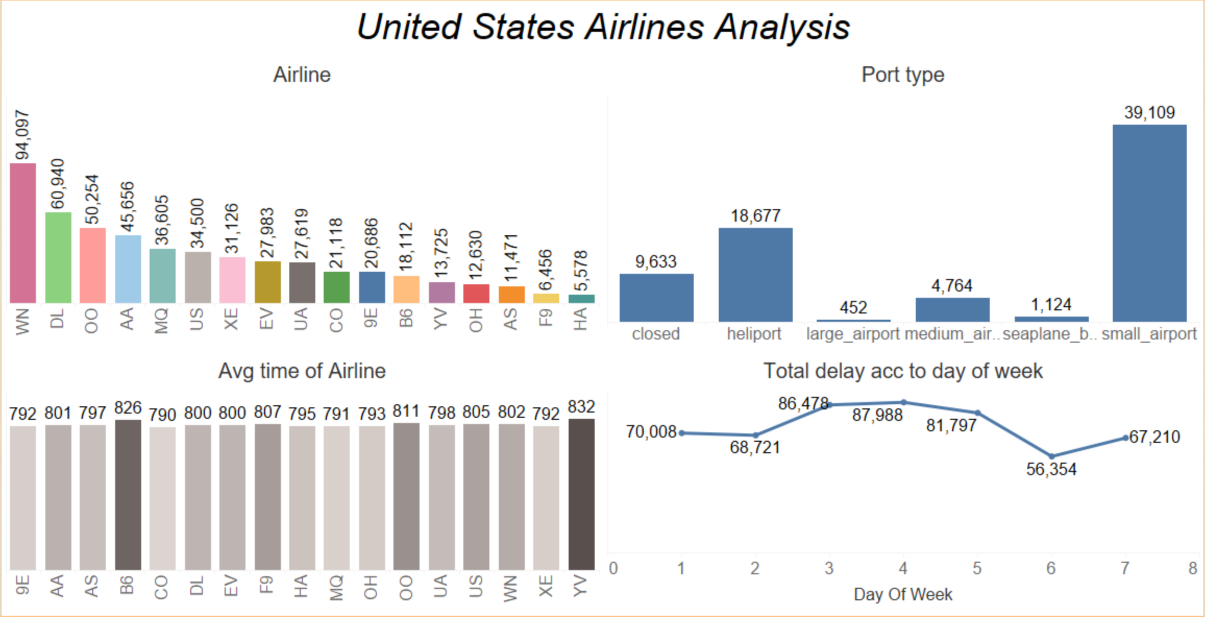
Average of Time for each Airline. Color shows average of Time.

Total delay acc to day of week:

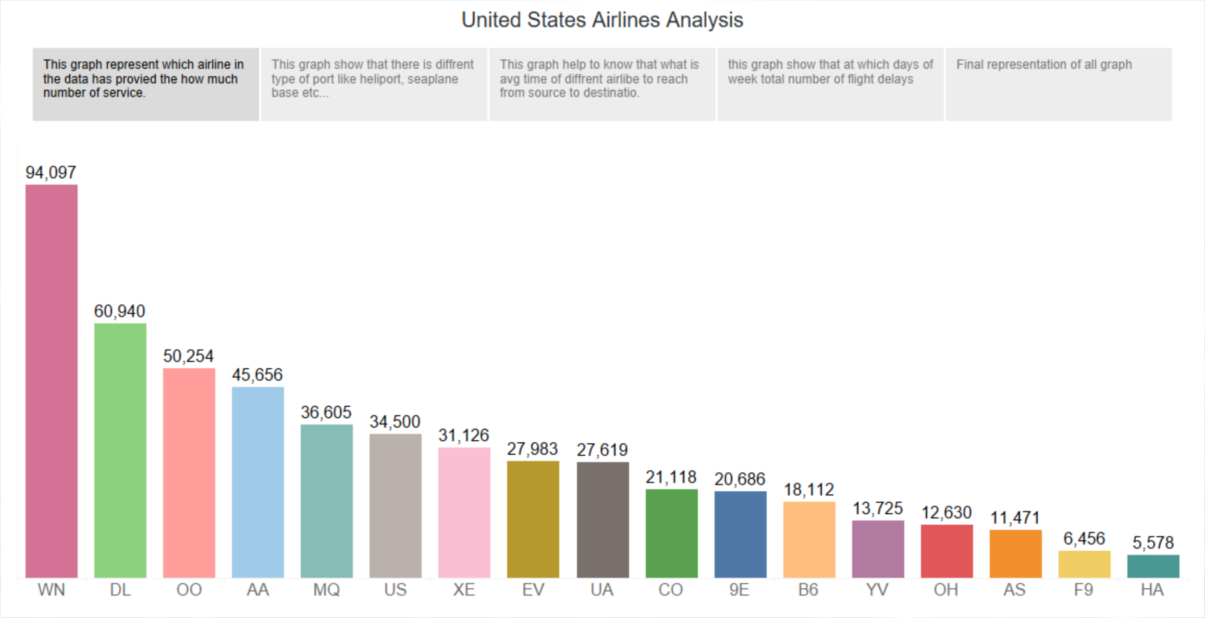


The trend of count of Delay for Day Of Week.

Dashboard:



Story:


United States Airlines Analysis

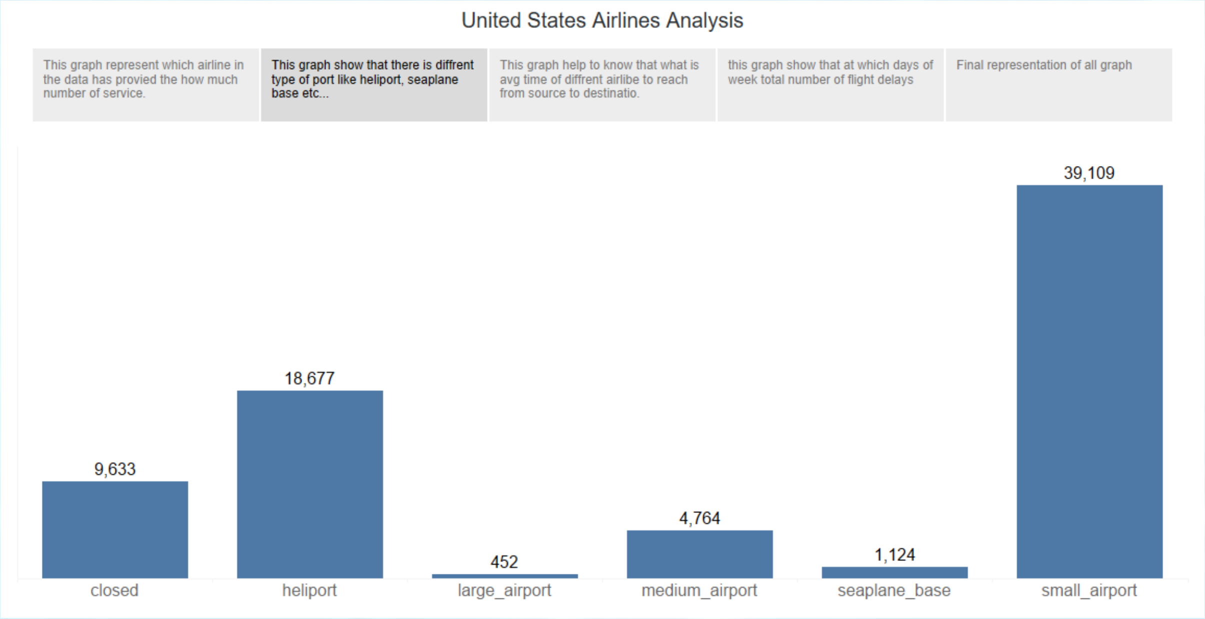
This graph represent which airline in the data has provied the how much number of service.

This graph show that there is diffrent type of port like heliport, seaplane base etc...

This graph help to know that what is avg time of diffrent airlibe to reach from source to destinatio.

this graph show that at which days of week total number of flight delays

Final representation of all graph



United States Airlines Analysis

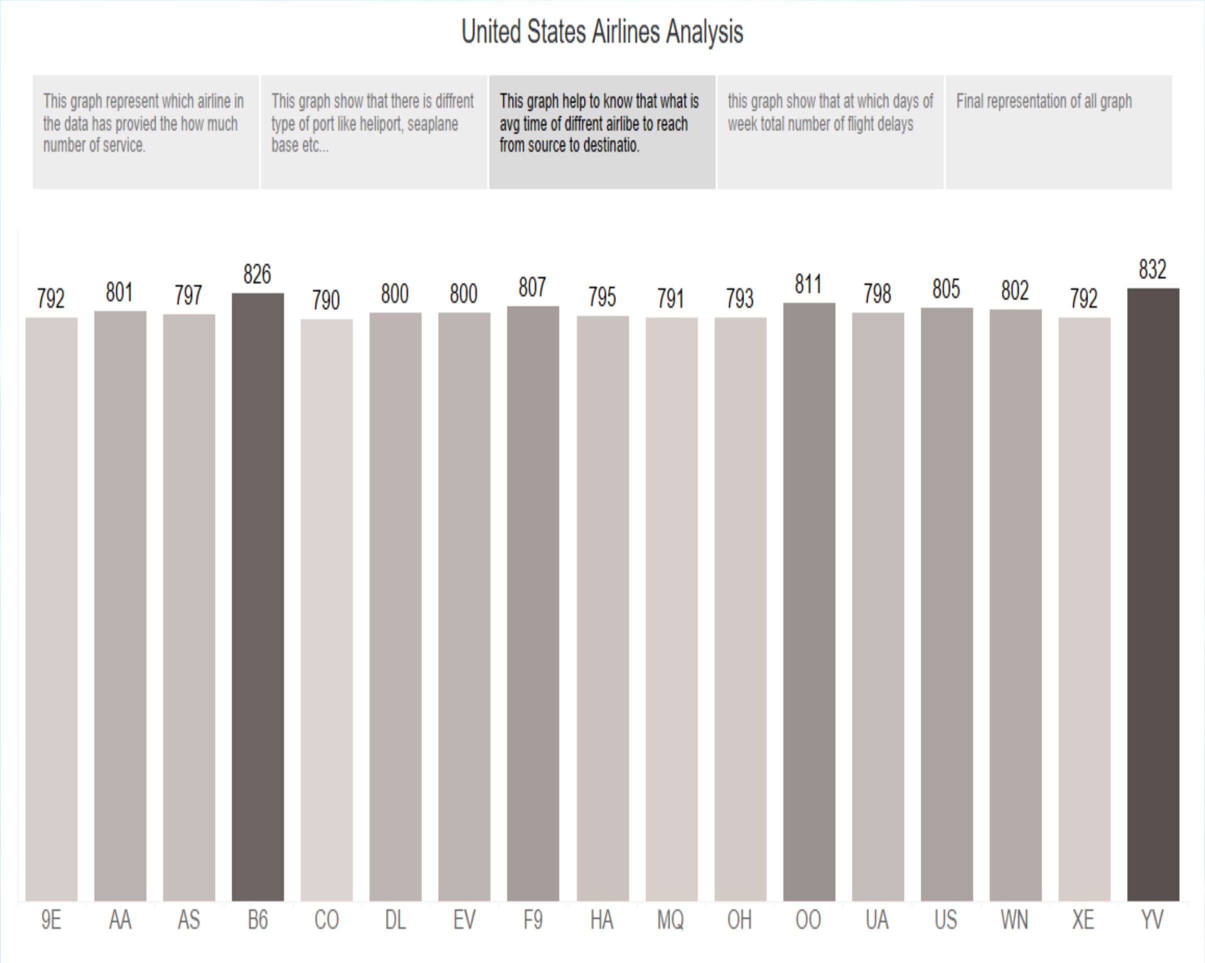
This graph represent which airline in the data has provied the how much number of service.

This graph show that there is diffrent type of port like heliport, seaplane base etc...

This graph help to know that what is avg time of diffrent airlibe to reach from source to destinatio.

this graph show that at which days of week total number of flight delays

Final representation of all graph



United States Airlines Analysis

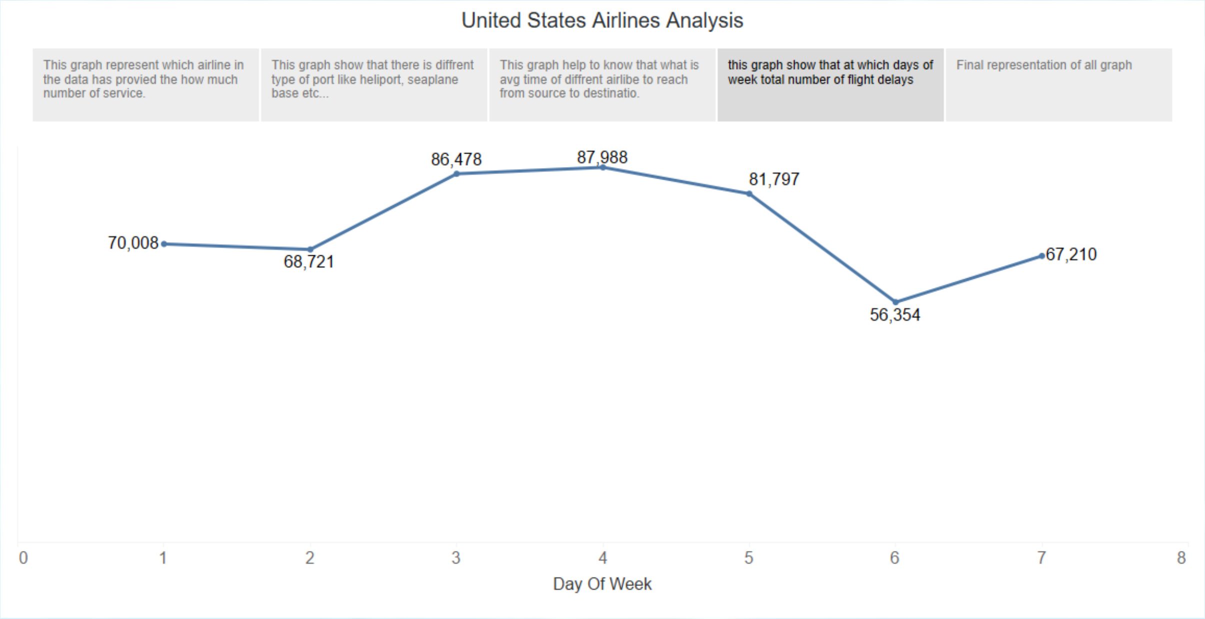
This graph represent which airline in the data has provied the how much number of service.

This graph show that there is diffrent type of port like heliport, seaplane base etc...

This graph help to know that what is avg time of diffrent airlibe to reach from source to destinatio.

this graph show that at which days of week total number of flight delays

Final representation of all graph



United States Airlines Analysis

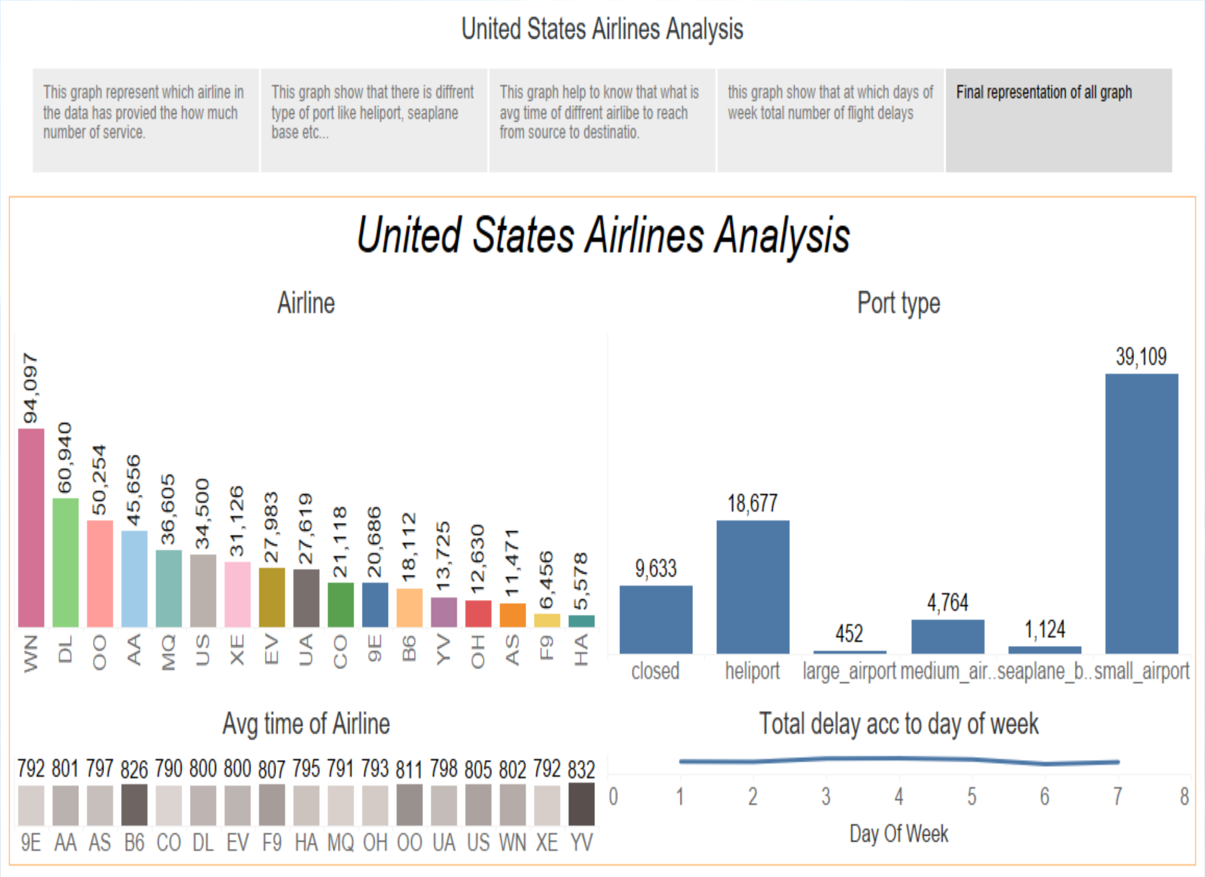
This graph represent which airline in the data has provied the how much number of service.

This graph show that there is diffrent type of port like heliport, seaplane base etc...

This graph help to know that what is avg time of diffrent airlibe to reach from source to destinatio.

this graph show that at which days of week total number of flight delays

Final representation of all graph



United States Airlines Analysis

This graph represent which airline in the data has provied the how much number of service.

This graph show that there is diffrent type of port like heliport, seaplane base etc...

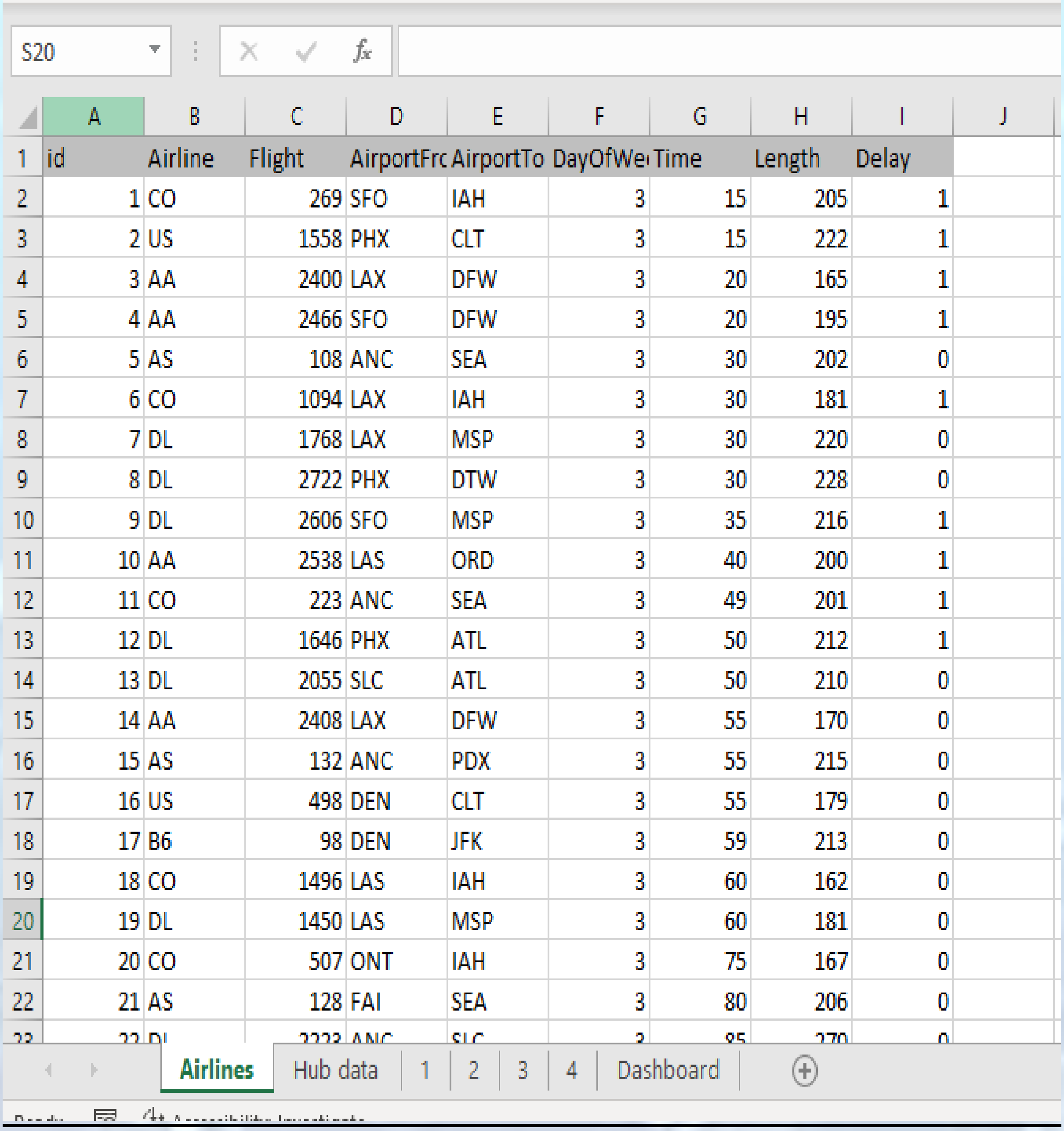
This graph help to know that what is avg time of diffrent airlibe to reach from source to destinatio.

this graph show that at which days of week total number of flight delays

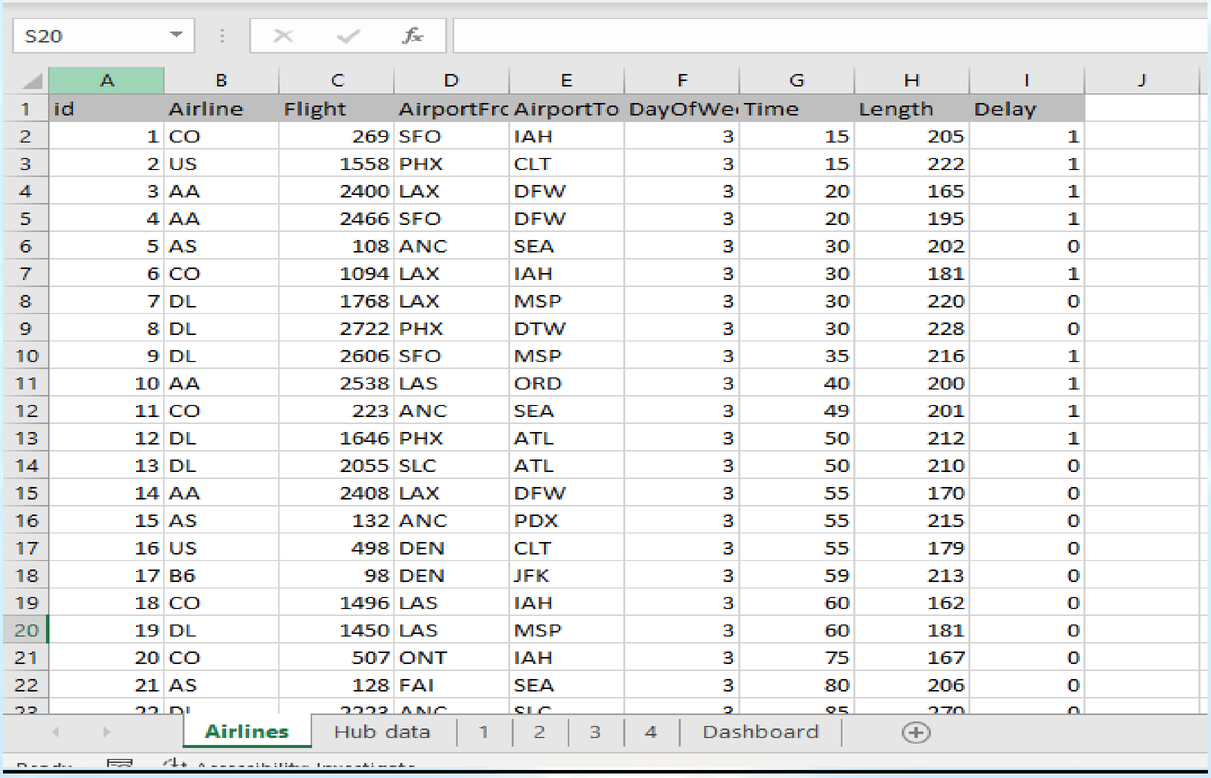
Final representation of all graph

Excel

Create an Excel dashboard showcasing the following (use form controls to make a dynamic chart):

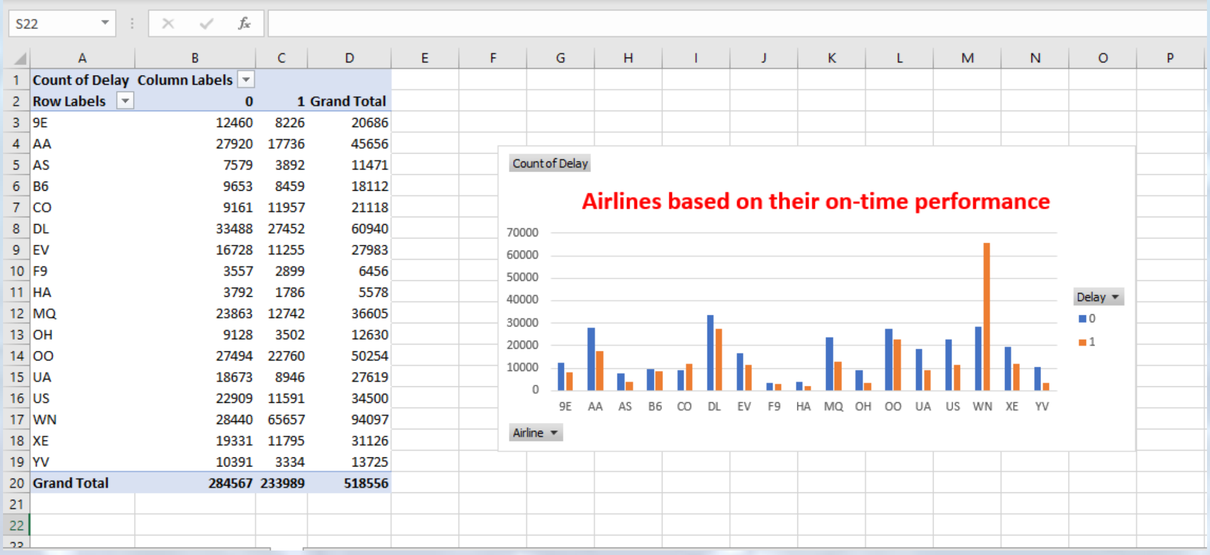


	A	B	C	D	E	F	G	H	I	J
1	id	Airline	Flight	AirportFrc	AirportTo	DayOfWe	Time	Length	Delay	
2	1	CO	269	SFO	IAH	3	15	205	1	
3	2	US	1558	PHX	CLT	3	15	222	1	
4	3	AA	2400	LAX	DFW	3	20	165	1	
5	4	AA	2466	SFO	DFW	3	20	195	1	
6	5	AS	108	ANC	SEA	3	30	202	0	
7	6	CO	1094	LAX	IAH	3	30	181	1	
8	7	DL	1768	LAX	MSP	3	30	220	0	
9	8	DL	2722	PHX	DTW	3	30	228	0	
10	9	DL	2606	SFO	MSP	3	35	216	1	
11	10	AA	2538	LAS	ORD	3	40	200	1	
12	11	CO	223	ANC	SEA	3	49	201	1	
13	12	DL	1646	PHX	ATL	3	50	212	1	
14	13	DL	2055	SLC	ATL	3	50	210	0	
15	14	AA	2408	LAX	DFW	3	55	170	0	
16	15	AS	132	ANC	PDX	3	55	215	0	
17	16	US	498	DEN	CLT	3	55	179	0	
18	17	B6	98	DEN	JFK	3	59	213	0	
19	18	CO	1496	LAS	IAH	3	60	162	0	
20	19	DL	1450	LAS	MSP	3	60	181	0	
21	20	CO	507	ONT	IAH	3	75	167	0	
22	21	AS	128	FAI	SEA	3	80	206	0	
23	22	DL	2223	ANC	SLC	3	85	270	0	



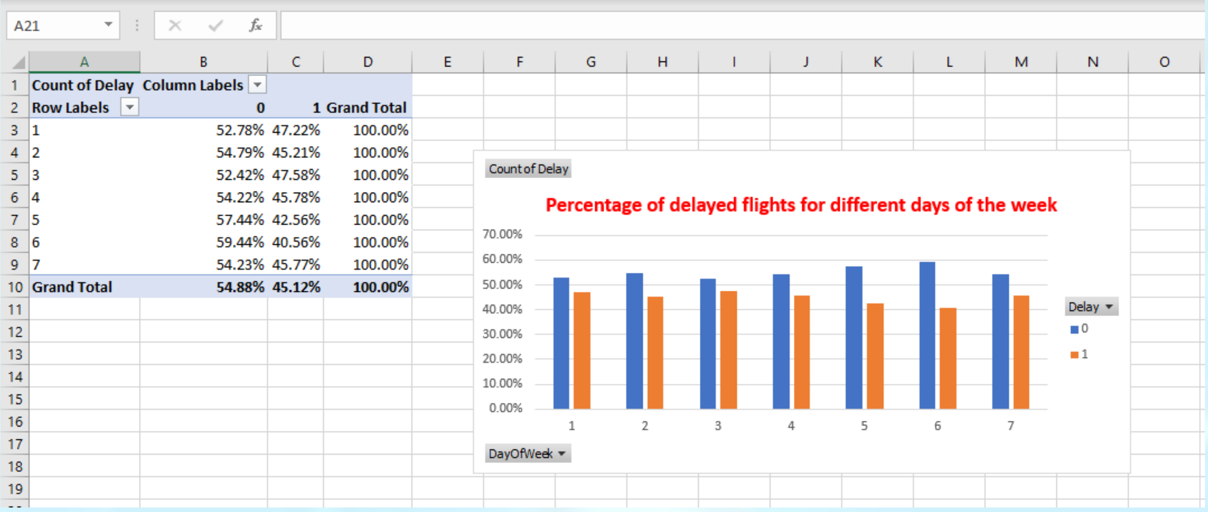
id	Airline	Flight	AirportFrc	AirportTo	DayOfWe	Time	Length	Delay
1	CO	269	SFO	IAH	3	15	205	1
2	US	1558	PHX	CLT	3	15	222	1
3	AA	2400	LAX	DFW	3	20	165	1
4	AA	2466	SFO	DFW	3	20	195	1
5	AS	108	ANC	SEA	3	30	202	0
6	CO	1094	LAX	IAH	3	30	181	1
7	DL	1768	LAX	MSP	3	30	220	0
8	DL	2722	PHX	DTW	3	30	228	0
9	DL	2606	SFO	MSP	3	35	216	1
10	AA	2538	LAS	ORD	3	40	200	1
11	CO	223	ANC	SEA	3	49	201	1
12	DL	1646	PHX	ATL	3	50	212	1
13	DL	2055	SLC	ATL	3	50	210	0
14	AA	2408	LAX	DFW	3	55	170	0
15	AS	132	ANC	PDX	3	55	215	0
16	US	498	DEN	CLT	3	55	179	0
17	B6	98	DEN	JFK	3	59	213	0
18	CO	1496	LAS	IAH	3	60	162	0
19	DL	1450	LAS	MSP	3	60	181	0
20	CO	507	ONT	IAH	3	75	167	0
21	AS	128	FAI	SEA	3	80	206	0

a. Compare different airlines based on their on-time performance



Count of Delay	Column Labels		
Row Labels	0	1	Grand Total
9E	12460	8226	20686
AA	27920	17736	45656
AS	7579	3892	11471
B6	9653	8459	18112
CO	9161	11957	21118
DL	33488	27452	60940
EV	16728	11255	27983
F9	3557	2899	6456
HA	3792	1786	5578
MQ	23863	12742	36605
OH	9128	3502	12630
OO	27494	22760	50254
UA	18673	8946	27619
US	22909	11591	34500
WN	28440	65657	94097
XE	19331	11795	31126
YV	10391	3334	13725
Grand Total	284567	233989	518556

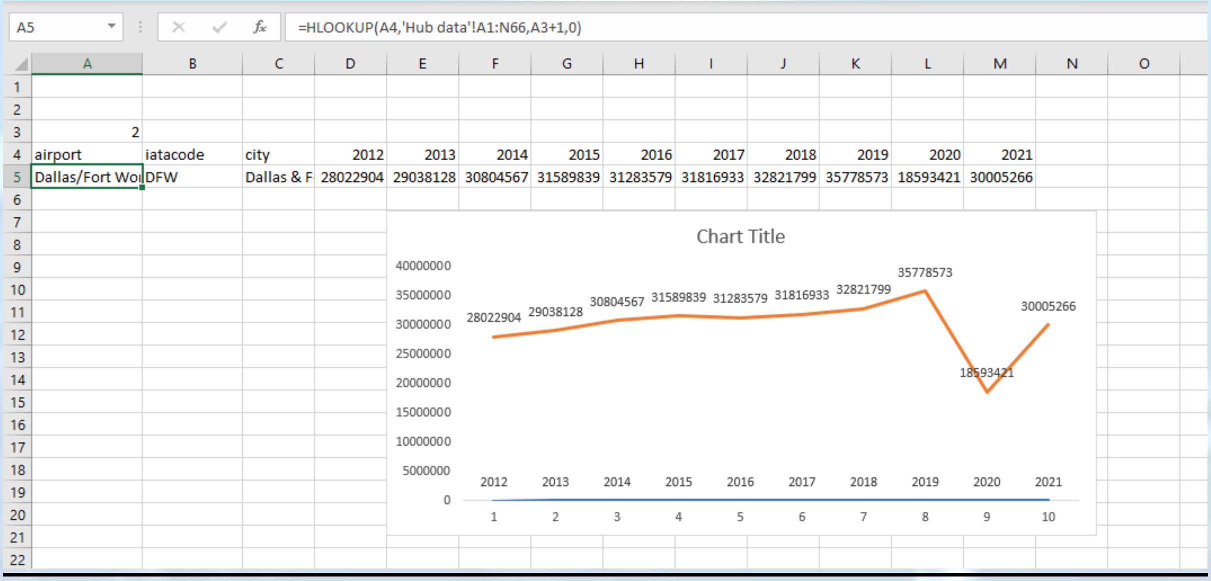
b. Compare the percentage of delayed flights for different days of the week



A21

Count of Delay	Column Labels		
Row Labels	0	1	Grand Total
1	52.78%	47.22%	100.00%
2	54.79%	45.21%	100.00%
3	52.42%	47.58%	100.00%
4	54.22%	45.78%	100.00%
5	57.44%	42.56%	100.00%
6	59.44%	40.56%	100.00%
7	54.23%	45.77%	100.00%
Grand Total	54.88%	45.12%	100.00%

c. Create a trend chart for the number of passengers at large and medium hubs



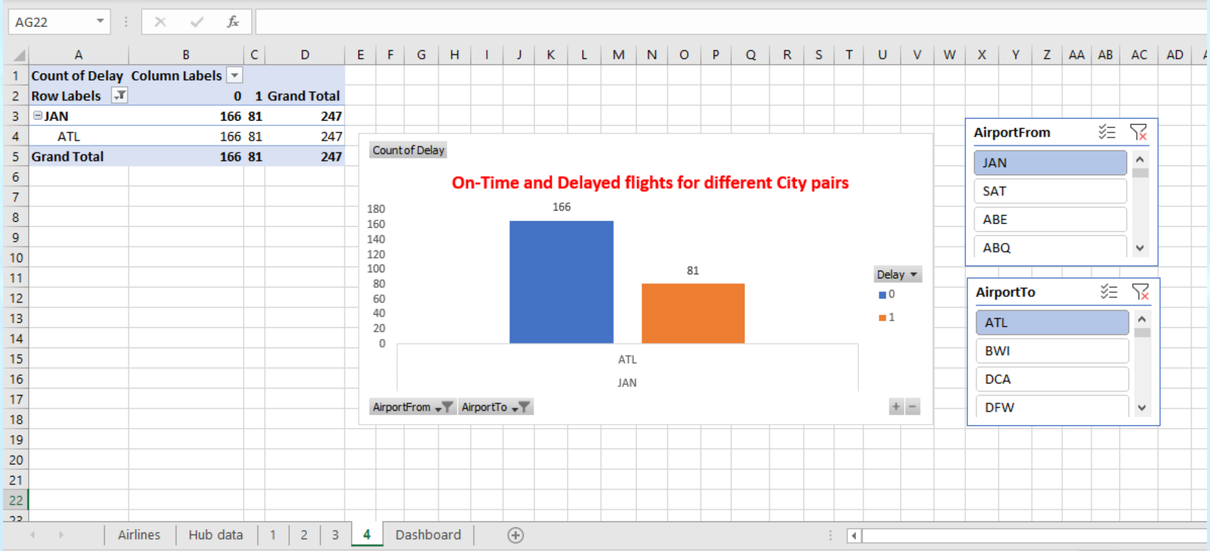
A5

=HLOOKUP(A4,'Hub data'!A1:N66,A3+1,0)

2											
airport	iatacode	city	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Dallas/Fort Wo	DFW	Dallas & F	28022904	29038128	30804567	31589839	31283579	31816933	32821799	35778573	18593421	30005266

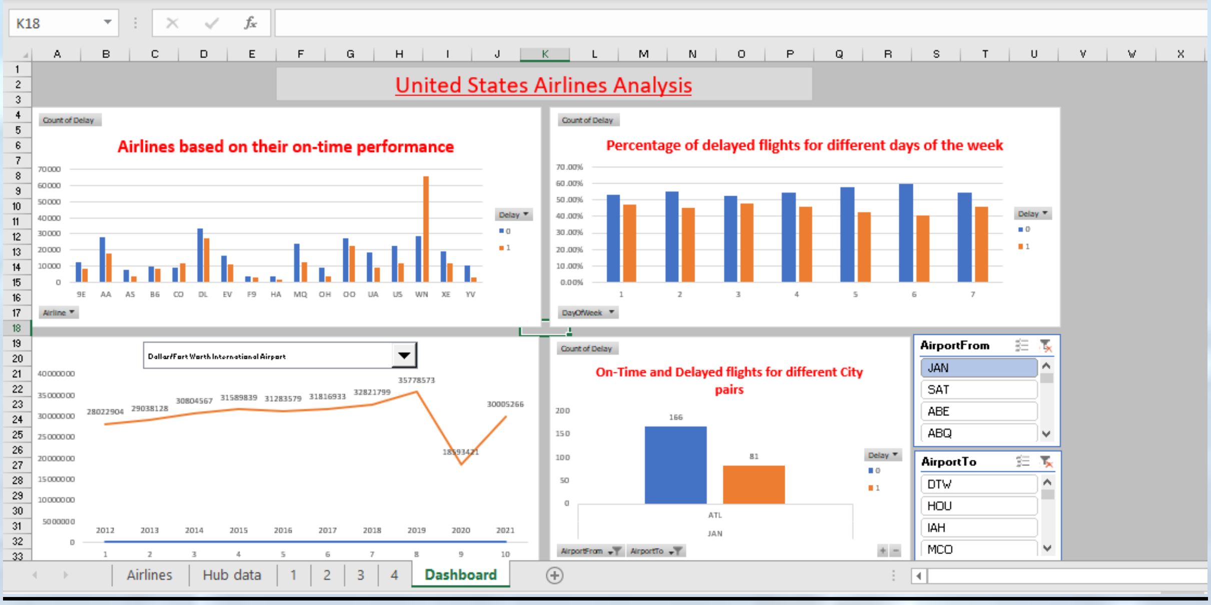
e. **Visualize the count of delayed and on-time flights for different pairs of source and destination airports**

***Create a dynamic chart that allows users to select a source and destination airport.**



Count of Delay	Column Labels		
Row Labels	0	1	Grand Total
JAN	166	81	247
ATL	166	81	247
Grand Total	166	81	247

Dashboard:



THANK YOU