00-eda

January 31, 2018

0.1 On Time Performance

0.1.1 Import Packages

```
In [1]: import numpy as np
        import pandas as pd

        import glob

        import datetime as dt
        from datetime import datetime, timedelta

        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style('darkgrid')

        %matplotlib inline
        pd.options.display.max_columns = None # Display all columns
```

0.1.2 Import Data

```
In [2]: # Create path/pattern for data import
        path = '../data/'
        pattern = '*.csv'
        path+pattern

Out[2]: '../data/*.csv'

In [3]: # Create csv_file list of data files
        csv_files = glob.glob(path + pattern)
        csv_files

Out[3]: ['../data/11-nov.csv', '../data/10-oct.csv', '../data/09-sep.csv']

In [4]: # Import and concatinate into one DataFrame
        df = pd.concat((pd.read_csv(f,low_memory=False) for f in csv_files))

In [5]: # Reverse dataframe
        df = df.iloc[::-1]
```

```
# Reset index due to DataFrame reversal
df = df.reset_index(drop=True)
# Save DataFrame as csv
df.to_csv('../data/df.csv')
```

0.1.3 Summary Statistics

```
In [6]: df.shape
```

```
Out[6]: (1392686, 18)
```

```
In [7]: df.head()
```

```
Out[7]:    YEAR  MONTH  DAY_OF_MONTH  DAY_OF_WEEK     FL_DATE CARRIER ORIGIN DEST  \
        0  2017      9            25            1  2017-09-25      B6    STT  SJU
        1  2017      9            25            1  2017-09-25      B6    SJU  MCO
        2  2017      9            25            1  2017-09-25      B6    BOS  HOU
        3  2017      9            25            1  2017-09-25      B6    BOS  BWI
        4  2017      9            25            1  2017-09-25      B6    DCA  MCO

           DEP_DELAY  TAXI_OUT  TAXI_IN  ARR_DELAY  CARRIER_DELAY  WEATHER_DELAY  \
        0        NaN       NaN      NaN        NaN            NaN            NaN
        1        NaN       NaN      NaN        NaN            NaN            NaN
        2       15.0      17.0      7.0      -15.0            NaN            NaN
        3       -8.0      13.0      5.0      -16.0            NaN            NaN
        4      -15.0      35.0     12.0       -7.0            NaN            NaN

           NAS_DELAY  SECURITY_DELAY  LATE_AIRCRAFT_DELAY  Unnamed: 17
        0        NaN             NaN                  NaN          NaN
        1        NaN             NaN                  NaN          NaN
        2        NaN             NaN                  NaN          NaN
        3        NaN             NaN                  NaN          NaN
        4        NaN             NaN                  NaN          NaN
```

```
In [8]: df.tail()
```

```
Out[8]:          YEAR  MONTH  DAY_OF_MONTH  DAY_OF_WEEK     FL_DATE CARRIER ORIGIN  \
        1392681  2017     11            18            6  2017-11-18      AA    MSP
        1392682  2017     11            17            5  2017-11-17      AA    MSP
        1392683  2017     11            16            4  2017-11-16      AA    MSP
        1392684  2017     11            15            3  2017-11-15      AA    MSP
        1392685  2017     11            14            2  2017-11-14      AA    MSP

                DEST  DEP_DELAY  TAXI_OUT  TAXI_IN  ARR_DELAY  CARRIER_DELAY  \
        1392681  PHL       -3.0      12.0     13.0      -22.0            NaN
        1392682  PHL        1.0      13.0      8.0      -29.0            NaN
        1392683  PHL       -2.0      21.0     14.0      -17.0            NaN
        1392684  PHL      -10.0      10.0     17.0      -32.0            NaN
        1392685  PHL      -12.0      10.0      6.0      -44.0            NaN
```

```
         WEATHER_DELAY  NAS_DELAY  SECURITY_DELAY  LATE_AIRCRAFT_DELAY  \
1392681            NaN        NaN             NaN                  NaN
1392682            NaN        NaN             NaN                  NaN
1392683            NaN        NaN             NaN                  NaN
1392684            NaN        NaN             NaN                  NaN
1392685            NaN        NaN             NaN                  NaN

         Unnamed: 17
1392681          NaN
1392682          NaN
1392683          NaN
1392684          NaN
1392685          NaN
```

```
In [9]: df.sample(5)
```

```
Out[9]:         YEAR  MONTH  DAY_OF_MONTH  DAY_OF_WEEK     FL_DATE CARRIER ORIGIN  \
        287019  2017      9             2            6  2017-09-02      AA    PHX
        946320  2017     11             3            5  2017-11-03      AA    CLT
        561769  2017     10            31            2  2017-10-31      OO    SFO
        694749  2017     10             4            3  2017-10-04      UA    IAH
        562052  2017     10            31            2  2017-10-31      OO    SEA

               DEST  DEP_DELAY  TAXI_OUT  TAXI_IN  ARR_DELAY  CARRIER_DELAY  \
        287019  DFW        6.0      13.0     10.0        1.0            NaN
        946320  MIA       -4.0      18.0      7.0      -24.0            NaN
        561769  RNO       -8.0      24.0      7.0       -4.0            NaN
        694749  SFO       -5.0      15.0      9.0      -14.0            NaN
        562052  EUG      -14.0      20.0      3.0      -17.0            NaN

                WEATHER_DELAY  NAS_DELAY  SECURITY_DELAY  LATE_AIRCRAFT_DELAY  \
        287019            NaN        NaN             NaN                  NaN
        946320            NaN        NaN             NaN                  NaN
        561769            NaN        NaN             NaN                  NaN
        694749            NaN        NaN             NaN                  NaN
        562052            NaN        NaN             NaN                  NaN

                Unnamed: 17
        287019          NaN
        946320          NaN
        561769          NaN
        694749          NaN
        562052          NaN
```

```
In [10]: df.columns.tolist()
```

```
Out[10]: ['YEAR',
          'MONTH',
          'DAY_OF_MONTH',
          'DAY_OF_WEEK',
          'FL_DATE',
          'CARRIER',
          'ORIGIN',
          'DEST',
          'DEP_DELAY',
          'TAXI_OUT',
          'TAXI_IN',
          'ARR_DELAY',
          'CARRIER_DELAY',
          'WEATHER_DELAY',
          'NAS_DELAY',
          'SECURITY_DELAY',
          'LATE_AIRCRAFT_DELAY',
          'Unnamed: 17']
```

```
In [11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1392686 entries, 0 to 1392685
Data columns (total 18 columns):
YEAR                       1392686 non-null int64
MONTH                      1392686 non-null int64
DAY_OF_MONTH               1392686 non-null int64
DAY_OF_WEEK                1392686 non-null int64
FL_DATE                    1392686 non-null object
CARRIER                    1392686 non-null object
ORIGIN                     1392686 non-null object
DEST                       1392686 non-null object
DEP_DELAY                  1372912 non-null float64
TAXI_OUT                   1372727 non-null float64
TAXI_IN                    1372290 non-null float64
ARR_DELAY                  1370514 non-null float64
CARRIER_DELAY              179239 non-null float64
WEATHER_DELAY              179239 non-null float64
NAS_DELAY                  179239 non-null float64
SECURITY_DELAY             179239 non-null float64
LATE_AIRCRAFT_DELAY        179239 non-null float64
Unnamed: 17                0 non-null float64
dtypes: float64(10), int64(4), object(4)
memory usage: 191.3+ MB
```

```
In [12]: df.describe()
```

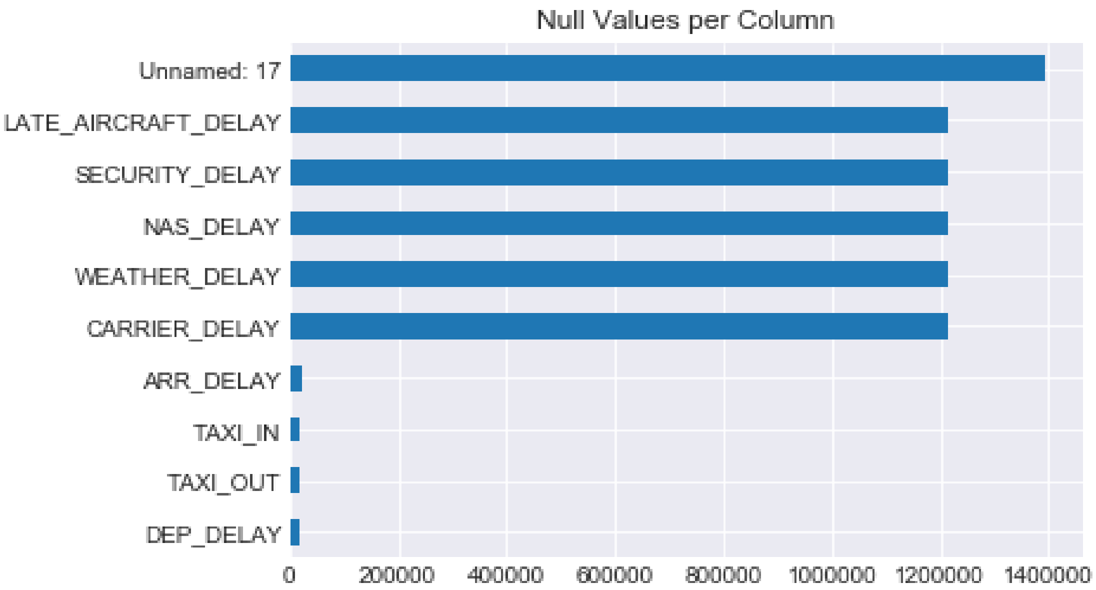
```
Out[12]:             YEAR         MONTH  DAY_OF_MONTH   DAY_OF_WEEK     DEP_DELAY  \
         count  1392686.0  1.392686e+06  1.392686e+06  1.392686e+06  1.372912e+06
         mean      2017.0  9.996722e+00  1.566044e+01  3.929426e+00  5.449386e+00
```

```
std          0.0  8.096158e-01  8.728560e+00  1.996739e+00  3.605892e+01
min       2017.0  9.000000e+00  1.000000e+00  1.000000e+00 -2.340000e+02
25%       2017.0  9.000000e+00  8.000000e+00  2.000000e+00 -6.000000e+00
50%       2017.0  1.000000e+01  1.600000e+01  4.000000e+00 -3.000000e+00
75%       2017.0  1.100000e+01  2.300000e+01  6.000000e+00  2.000000e+00
max       2017.0  1.100000e+01  3.100000e+01  7.000000e+00  1.816000e+03

           TAXI_OUT       TAXI_IN     ARR_DELAY  CARRIER_DELAY  WEATHER_DELAY  \
count  1.372727e+06  1.372290e+06  1.370514e+06  179239.000000  179239.000000
mean   1.640912e+01  7.157819e+00 -8.190059e-01      20.596427       1.769648
std    8.329690e+00  5.364448e+00  3.786952e+01      62.421050      20.740136
min    0.000000e+00  0.000000e+00 -2.380000e+02       0.000000       0.000000
25%    1.100000e+01  4.000000e+00 -1.600000e+01       0.000000       0.000000
50%    1.400000e+01  6.000000e+00 -8.000000e+00       1.000000       0.000000
75%    1.900000e+01  8.000000e+00  2.000000e+00      18.000000       0.000000
max    1.830000e+02  1.770000e+02  1.810000e+03    1810.000000    1336.000000

           NAS_DELAY  SECURITY_DELAY  LATE_AIRCRAFT_DELAY  Unnamed: 17
count  179239.000000   179239.000000        179239.000000          0.0
mean       14.907263        0.162504            22.079514          NaN
std        34.093446        4.902098            44.851681          NaN
min         0.000000        0.000000             0.000000          NaN
25%         0.000000        0.000000             0.000000          NaN
50%         2.000000        0.000000             0.000000          NaN
75%        18.000000        0.000000            26.000000          NaN
max      1549.000000      827.000000          1509.000000          NaN
```

```
In [13]: # Columns with null (np.nan) values
         nan_col_list = df.columns[df.isnull().any()].tolist()

         # Sum of nan values of each column
         nulls = df[nan_col_list].isnull().sum()
         nulls
```

```
Out[13]: DEP_DELAY                19774
         TAXI_OUT                 19959
         TAXI_IN                  20396
         ARR_DELAY                22172
         CARRIER_DELAY          1213447
         WEATHER_DELAY          1213447
         NAS_DELAY              1213447
         SECURITY_DELAY         1213447
         LATE_AIRCRAFT_DELAY    1213447
         Unnamed: 17            1392686
         dtype: int64
```

```
In [14]: nulls.plot(kind='barh', title='Null Values per Column')
         plt.savefig('../assets/png/01-nulls.png')
         plt.show()
```



0.1.4 Data Cleaning

```
In [15]: # Drop 'Unnamed: 17'
         df.drop('Unnamed: 17', axis=1, inplace=True)
```

```
In [16]: # Change date column to datetime format
         df['FL_DATE'] = pd.to_datetime(df['FL_DATE'])
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1392686 entries, 0 to 1392685
Data columns (total 17 columns):
YEAR                   1392686 non-null int64
MONTH                  1392686 non-null int64
DAY_OF_MONTH           1392686 non-null int64
DAY_OF_WEEK            1392686 non-null int64
FL_DATE                1392686 non-null datetime64[ns]
CARRIER                1392686 non-null object
ORIGIN                 1392686 non-null object
DEST                   1392686 non-null object
DEP_DELAY              1372912 non-null float64
TAXI_OUT               1372727 non-null float64
TAXI_IN                1372290 non-null float64
ARR_DELAY              1370514 non-null float64
CARRIER_DELAY          179239 non-null float64
WEATHER_DELAY          179239 non-null float64
NAS_DELAY              179239 non-null float64
```

```
SECURITY_DELAY         179239 non-null float64
LATE_AIRCRAFT_DELAY    179239 non-null float64
dtypes: datetime64[ns](1), float64(9), int64(4), object(3)
memory usage: 180.6+ MB
```

```
In [17]: # Convert int64 columns to object/string
         convert_list_int = ['YEAR','MONTH', 'DAY_OF_MONTH']
         df[convert_list_int] = df[convert_list_int].astype('object')
```

```
In [18]: # Rename the carrier codes to carrier name:
         print(df['CARRIER'].value_counts())
         carrier_dict = {'WN': 'Southwest',
                         'DL': 'Delta',
                         'AA': 'American',
                         'OO': 'SkyWest',
                         'UA': 'United',
                         'B6': 'JetBlue',
                         'EV': 'ExpressJet',
                         'AS': 'Alaska',
                         'NK': 'Spirit',
                         'F9': 'Frontier',
                         'HA': 'Hawaiian',
                         'VX': 'Virgin America'}
         df['CARRIER'] = df['CARRIER'].replace(carrier_dict)
```

```
WN    324892
DL    229329
AA    217450
OO    183785
UA    148513
B6     71906
EV     68257
AS     44648
NK     38805
F9     27097
HA     19815
VX     18189
Name: CARRIER, dtype: int64
```

```
In [19]: print(df['DAY_OF_WEEK'].value_counts())
         weekday_dict = {1: 'Monday',
                         2: 'Tuesday',
                         3: 'Wednesday',
                         4: 'Thursday',
                         5: 'Friday',
                         6: 'Saturday',
                     7: 'Sunday'}
         df['DAY_OF_WEEK'] = df['DAY_OF_WEEK'].replace(weekday_dict)
```

```
1    209595
5    207377
3    204561
4    204121
2    201251
7    198621
6    167160
Name: DAY_OF_WEEK, dtype: int64
```

```
In [20]: # Create list of numeric columns
         numeric_columns = df.select_dtypes(exclude=['object','datetime64']).columns
         numeric_columns
```

```
Out[20]: Index(['DEP_DELAY', 'TAXI_OUT', 'TAXI_IN', 'ARR_DELAY', 'CARRIER_DELAY',
                'WEATHER_DELAY', 'NAS_DELAY', 'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY'],
               dtype='object')
```

```
In [21]: # Create list of categorical columns
         cat_columns = df.select_dtypes(include=['object']).columns
         cat_columns
```

```
Out[21]: Index(['YEAR', 'MONTH', 'DAY_OF_MONTH', 'DAY_OF_WEEK', 'CARRIER', 'ORIGIN',
                'DEST'],
               dtype='object')
```

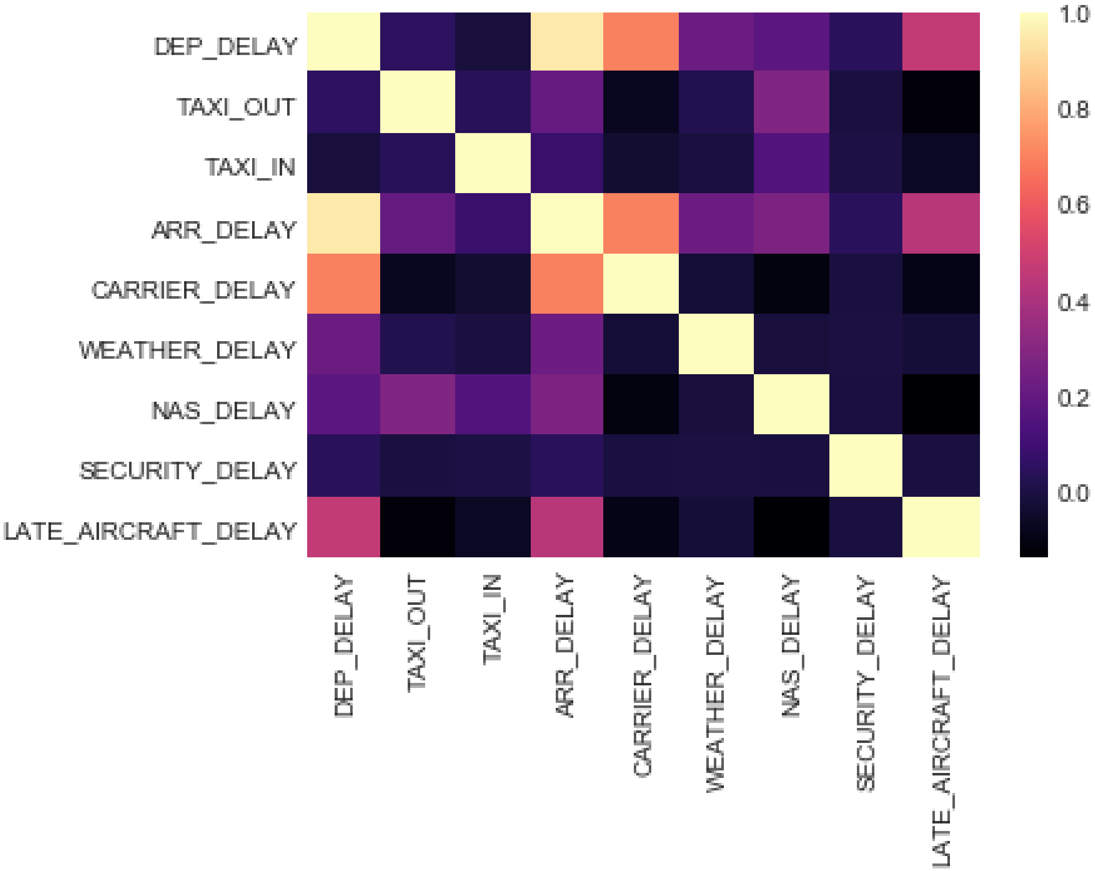
```
In [22]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1392686 entries, 0 to 1392685
Data columns (total 17 columns):
YEAR                   1392686 non-null object
MONTH                  1392686 non-null object
DAY_OF_MONTH           1392686 non-null object
DAY_OF_WEEK            1392686 non-null object
FL_DATE                1392686 non-null datetime64[ns]
CARRIER                1392686 non-null object
ORIGIN                 1392686 non-null object
DEST                   1392686 non-null object
DEP_DELAY              1372912 non-null float64
TAXI_OUT               1372727 non-null float64
TAXI_IN                1372290 non-null float64
ARR_DELAY              1370514 non-null float64
CARRIER_DELAY          179239 non-null float64
WEATHER_DELAY          179239 non-null float64
NAS_DELAY              179239 non-null float64
```

```
SECURITY_DELAY         179239 non-null float64
LATE_AIRCRAFT_DELAY    179239 non-null float64
dtypes: datetime64[ns](1), float64(9), object(7)
memory usage: 180.6+ MB
```

0.1.5 EDA Visualization

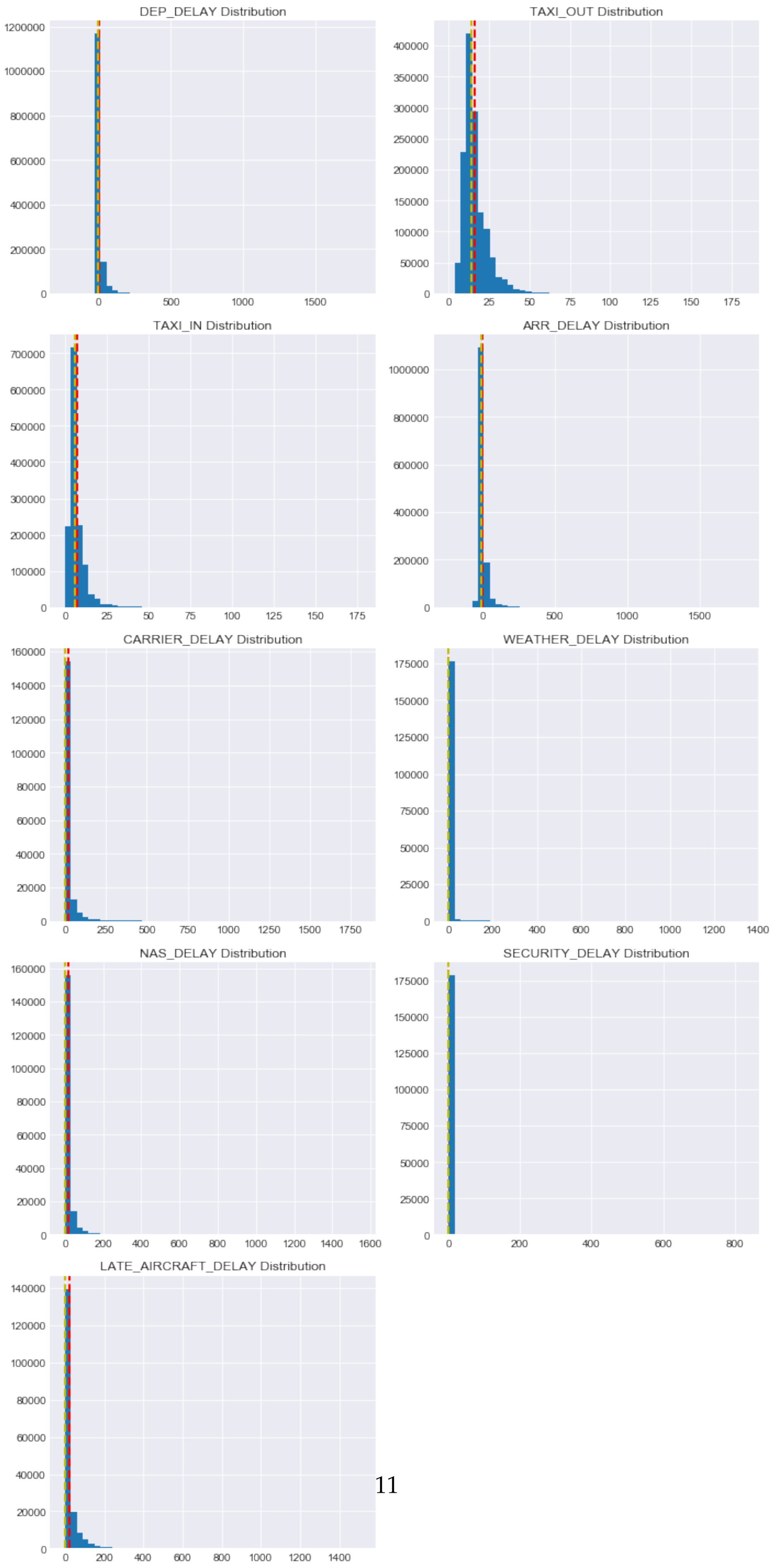
```
In [23]: corr = df.corr()
         sns.heatmap(corr,xticklabels=corr.columns.values,yticklabels=corr.columns.values,cmap=
         plt.savefig('../assets/png/02-correlations.png')
```



```
In [24]: def draw_histograms(df, variables, n_rows, n_cols):
             fig=plt.figure(figsize=(10,20))
             for i, var_name in enumerate(variables):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 df[var_name].hist(ax=ax, bins=50)
                 plt.axvline(df[var_name].mean(), color='r', linestyle='dashed', linewidth=2)
                 plt.axvline(df[var_name].median(), color='y', linestyle='dashed', linewidth=2
                 ax.set_title(var_name+" Distribution")
```

```
            fig.tight_layout()
            plt.savefig('../assets/png/03-histograms.png')
            plt.show()
```

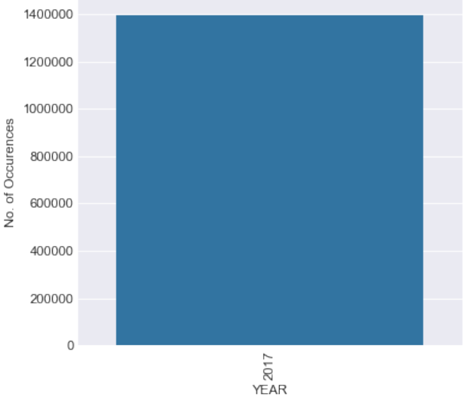
```
In [25]: draw_histograms(df, numeric_columns, int(len(numeric_columns)/2)+1, 2)
```


DEP_DELAY Distribution
1200000
1000000
800000
600000
400000
200000
0
0
500
1000
1500
TAXI_OUT Distribution
400000
350000
300000
250000
200000
150000
100000
50000
0
0
25
50
75
100
125
150
175
TAXI_IN Distribution
700000
600000
500000
400000
300000
200000
100000
0
0
25
50
75
100
125
150
175
ARR_DELAY Distribution
1000000
800000
600000
400000
200000
0
0
500
1000
1500
CARRIER_DELAY Distribution
160000
140000
120000
100000
80000
60000
40000
20000
0
0
250
500
750
1000
1250
1500
1750
WEATHER_DELAY Distribution
175000
150000
125000
100000
75000
50000
25000
0
0
200
400
600
800
1000
1200
1400
NAS_DELAY Distribution
160000
140000
120000
100000
80000
60000
40000
20000
0
0
200
400
600
800
1000
1200
1400
1600
SECURITY_DELAY Distribution
175000
150000
125000
100000
75000
50000
25000
0
0
200
400
600
800
LATE_AIRCRAFT_DELAY Distribution
140000
120000
100000
80000
60000
40000
20000
0
0
200
400
600
800
1000
1200
1400

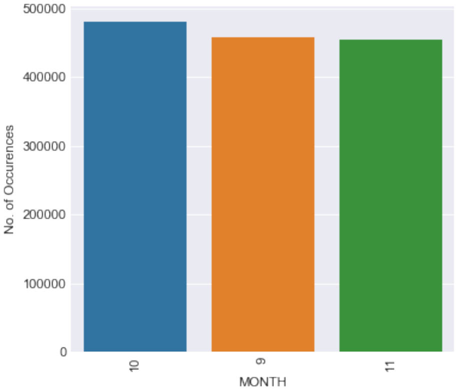
```
In [26]: def draw_cat_countplots(df,columns,n_rows,n_cols):
             fig=plt.figure(figsize=(10,20), dpi=80)
             for i, col in enumerate(cat_columns):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 sns.countplot(x=col, data=df, order=df[col].value_counts().index)
                 plt.xticks(rotation=90)
                 ax.set_title('DEPARTURE DELAYS')
                 plt.ylabel('No. of Occurences')
             plt.title('Departure Delays')
             fig.tight_layout()
             plt.savefig('../assets/png/04-cat-countplots.png')
             plt.show()
```

```
In [27]: draw_cat_countplots(df, cat_columns, int(len(cat_columns)/2)+1, 2)
```

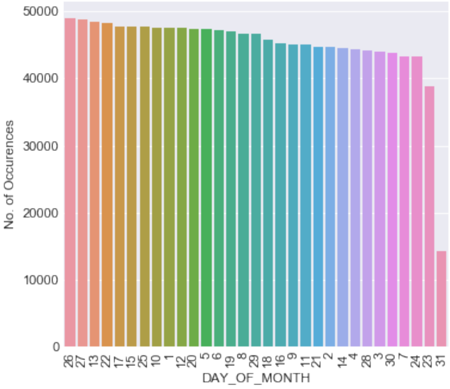
DEPARTURE DELAYS



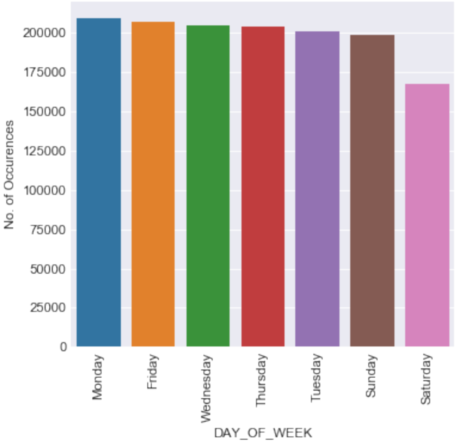
DEPARTURE DELAYS



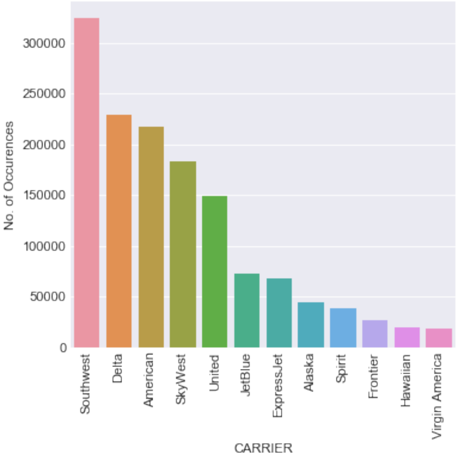
DEPARTURE DELAYS



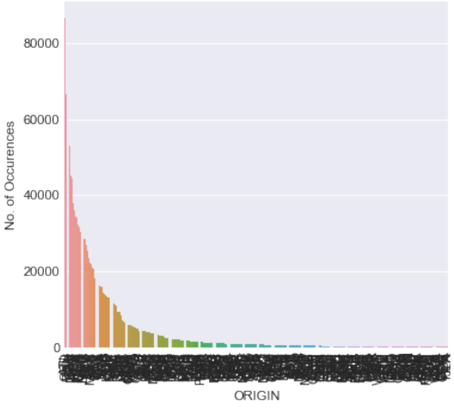
DEPARTURE DELAYS



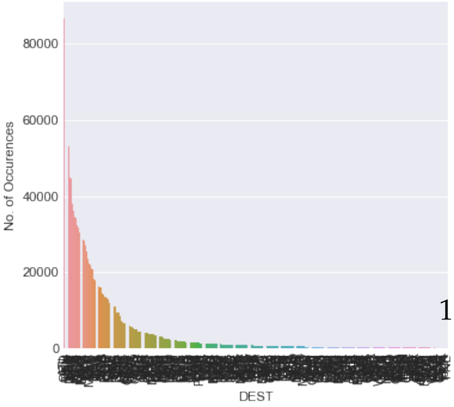
DEPARTURE DELAYS



DEPARTURE DELAYS

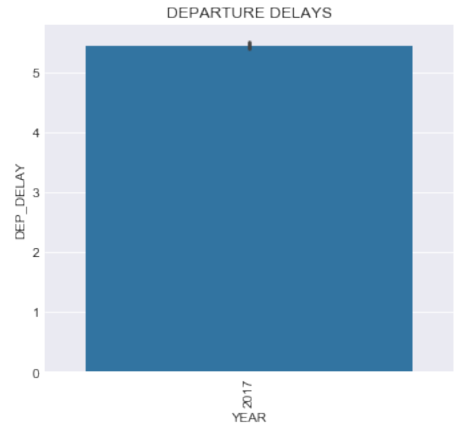


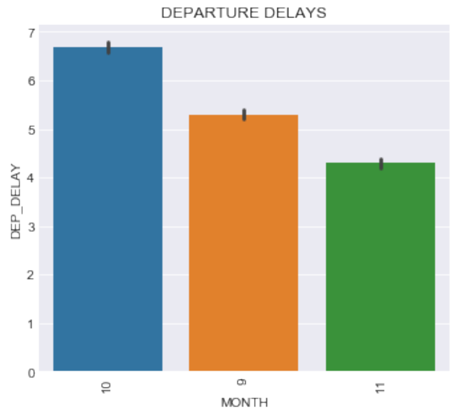
Departure Delays

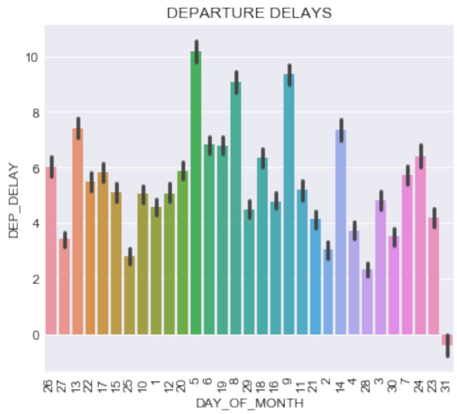


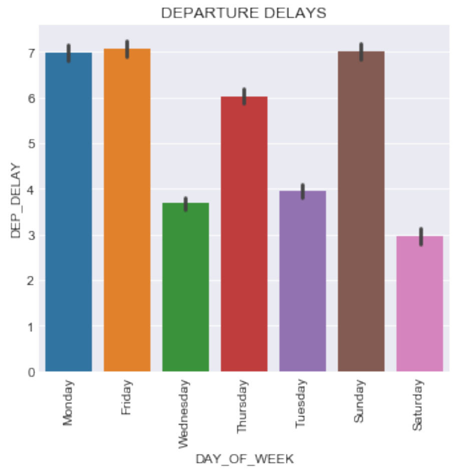
```
In [28]: # Draw bar plots of categorical data departures
         def draw_bars_dep(df, variables, n_rows, n_cols):
             fig=plt.figure(figsize=(10,20))
             for i, var_name in enumerate(variables):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 sns.barplot(x=var_name,y='DEP_DELAY',data=df,order=df[var_name].value_counts()
                 plt.xticks(rotation=90)
                 ax.set_title('DEPARTURE DELAYS')
             fig.tight_layout()
             plt.savefig('../assets/png/05-bars-departures.png')
             plt.show()
```

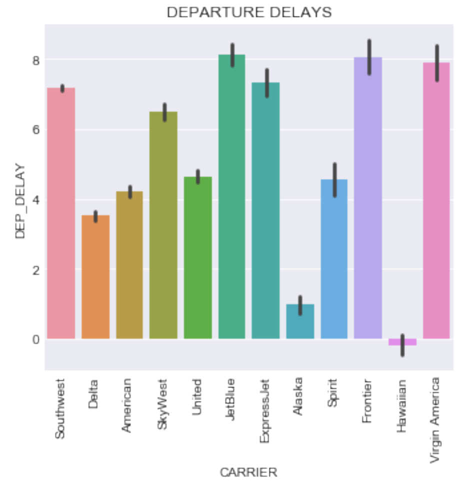
```
In [29]: draw_bars_dep(df, cat_columns, int(len(cat_columns)/2)+1, 2)
```

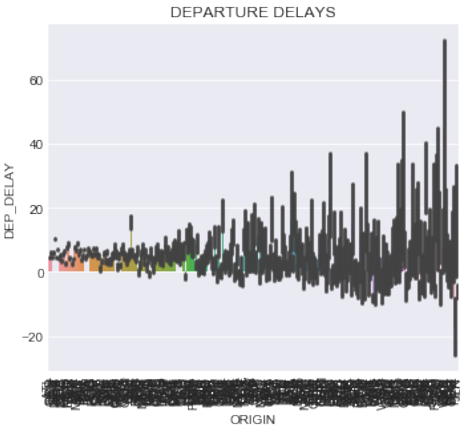

DEPARTURE DELAYS
DEP_DELAY
0
1
2
3
4
5
2017
YEAR

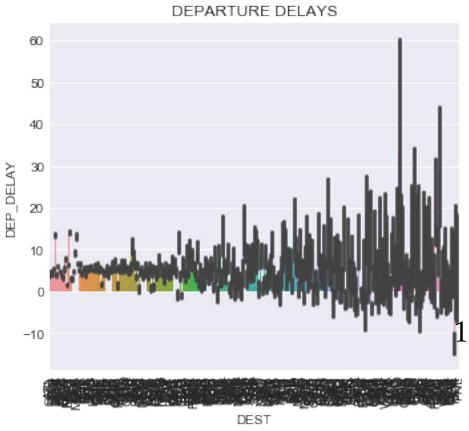

DEPARTURE DELAYS
DEP_DELAY
0
1
2
3
4
5
6
7
10
9
11
MONTH


DEPARTURE DELAYS
DEP_DELAY
0
2
4
6
8
10
26
27
13
22
17
15
25
10
1
12
20
5
6
19
8
29
18
16
9
11
21
2
14
4
28
3
30
7
24
23
31
DAY_OF_MONTH


DEPARTURE DELAYS
DEP_DELAY
0
1
2
3
4
5
6
7
Monday
Friday
Wednesday
Thursday
Tuesday
Sunday
Saturday
DAY_OF_WEEK

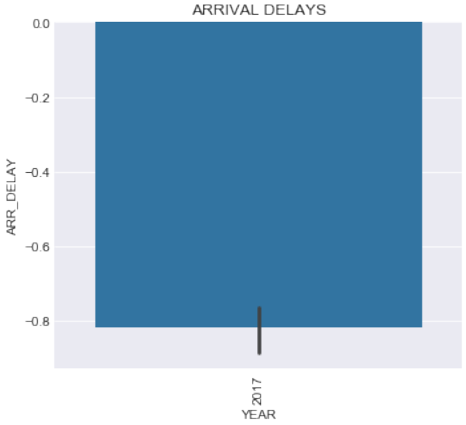

DEPARTURE DELAYS
DEP_DELAY
0
2
4
6
8
Southwest
Delta
American
SkyWest
United
JetBlue
ExpressJet
Alaska
Spirit
Frontier
Hawaiian
Virgin America
CARRIER

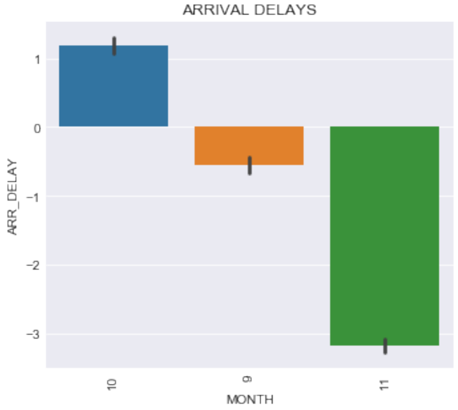

DEPARTURE DELAYS
DEP_DELAY
−20
0
20
40
60
ORIGIN

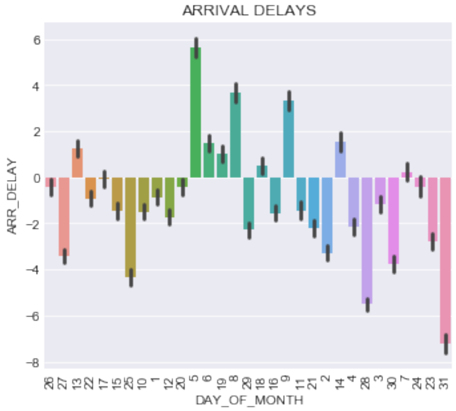

DEPARTURE DELAYS
DEP_DELAY
−10
0
10
20
30
40
50
60
DEST

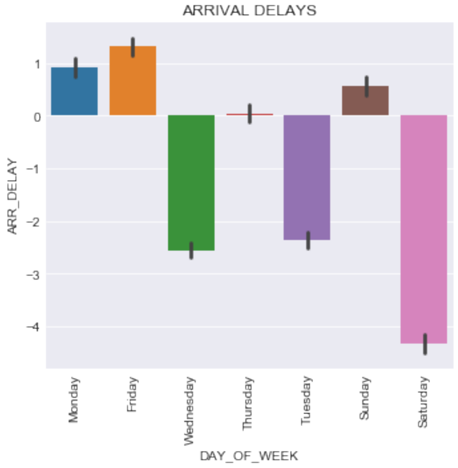
```
In [30]: # Draw bar plots of categorical data arrivals
         def draw_bars_arr(df, variables, n_rows, n_cols):
             fig=plt.figure(figsize=(10,20))
             for i, var_name in enumerate(variables):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 sns.barplot(x=var_name,y='ARR_DELAY',data=df,order=df[var_name].value_counts()
                 plt.xticks(rotation=90)
                 ax.set_title('ARRIVAL DELAYS')
             fig.tight_layout()
             plt.savefig('../assets/06-bars-arrivals.png')
             plt.show()
```

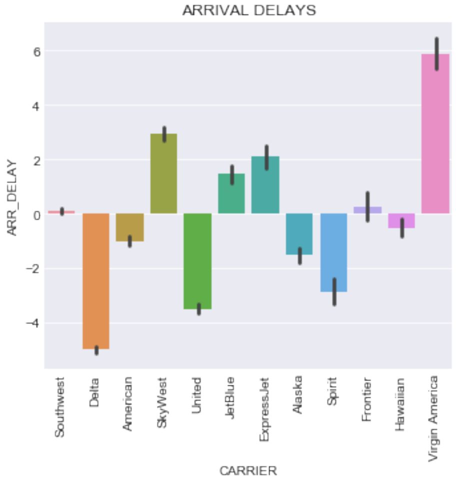
```
In [31]: draw_bars_arr(df, cat_columns, int(len(cat_columns)/2)+1, 2)
```

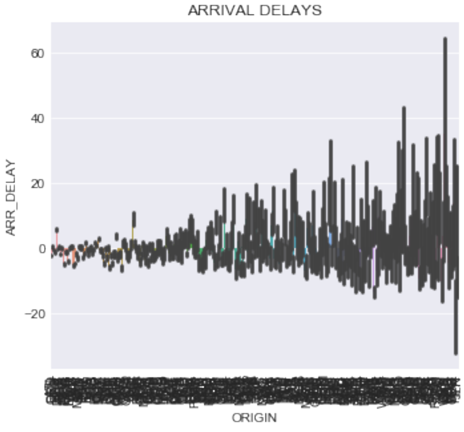

ARRIVAL DELAYS
0.0
−0.2
−0.4
−0.6
−0.8
ARR_DELAY
2017
YEAR

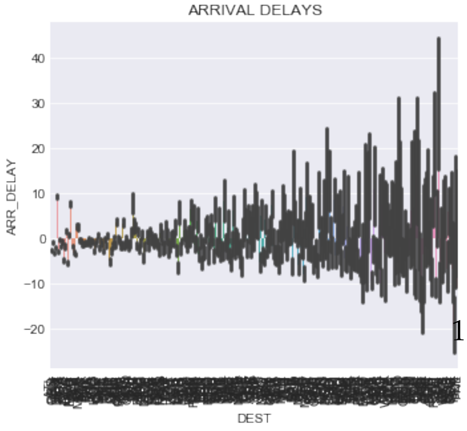

ARRIVAL DELAYS
1
0
−1
−2
−3
ARR_DELAY
10
9
11
MONTH


ARRIVAL DELAYS
6
4
2
0
−2
−4
−6
−8
ARR_DELAY
26
27
13
22
17
15
25
10
1
12
20
5
6
19
8
29
18
16
9
11
21
2
14
4
28
3
30
7
24
23
31
DAY_OF_MONTH


ARRIVAL DELAYS
1
0
−1
−2
−3
−4
ARR_DELAY
Monday
Friday
Wednesday
Thursday
Tuesday
Sunday
Saturday
DAY_OF_WEEK

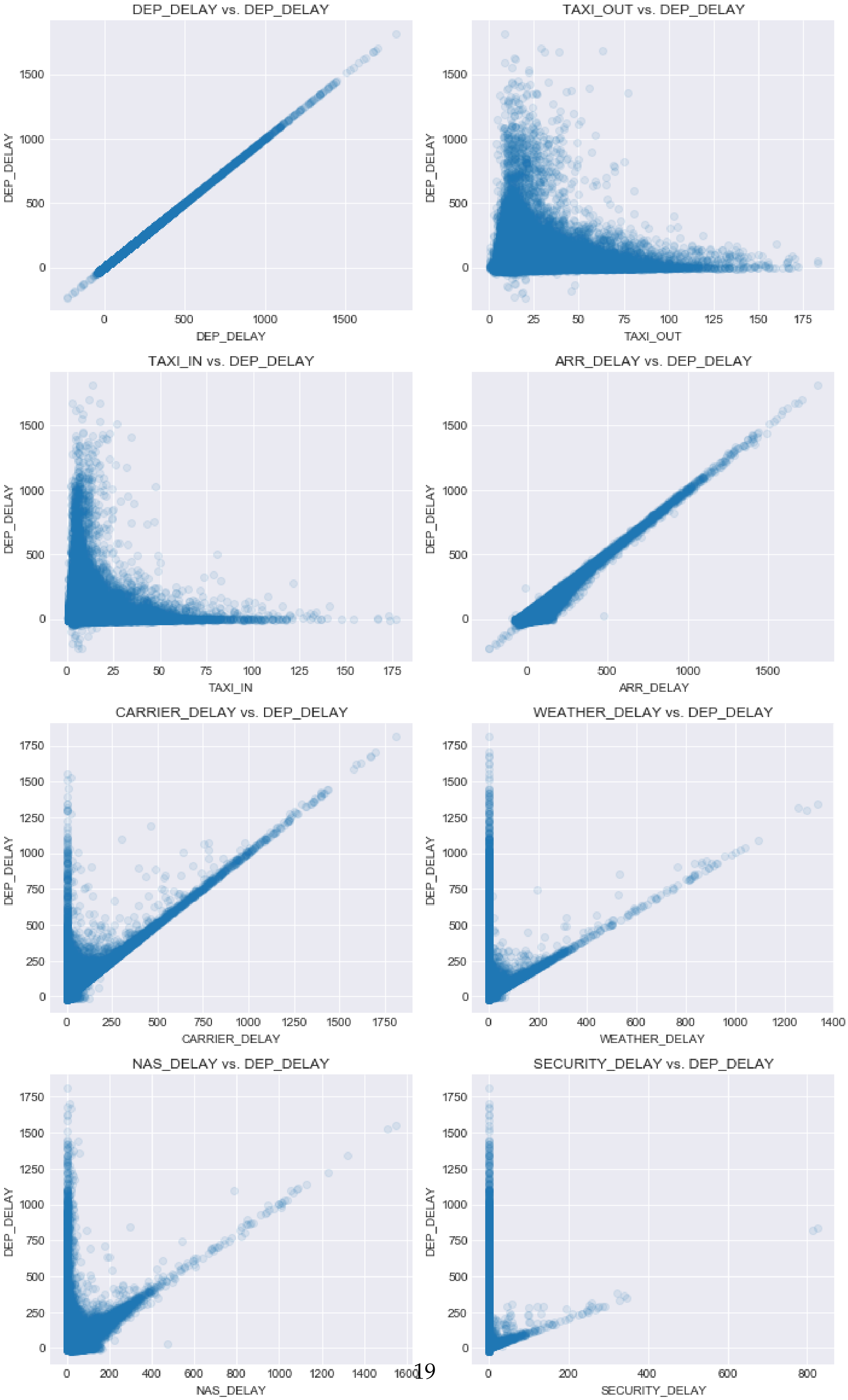

ARRIVAL DELAYS
6
4
2
0
−2
−4
ARR_DELAY
Southwest
Delta
American
SkyWest
United
JetBlue
ExpressJet
Alaska
Spirit
Frontier
Hawaiian
Virgin America
CARRIER


ARRIVAL DELAYS
60
40
20
0
−20
ARR_DELAY
ORIGIN


ARRIVAL DELAYS
40
30
20
10
0
−10
−20
ARR_DELAY
DEST

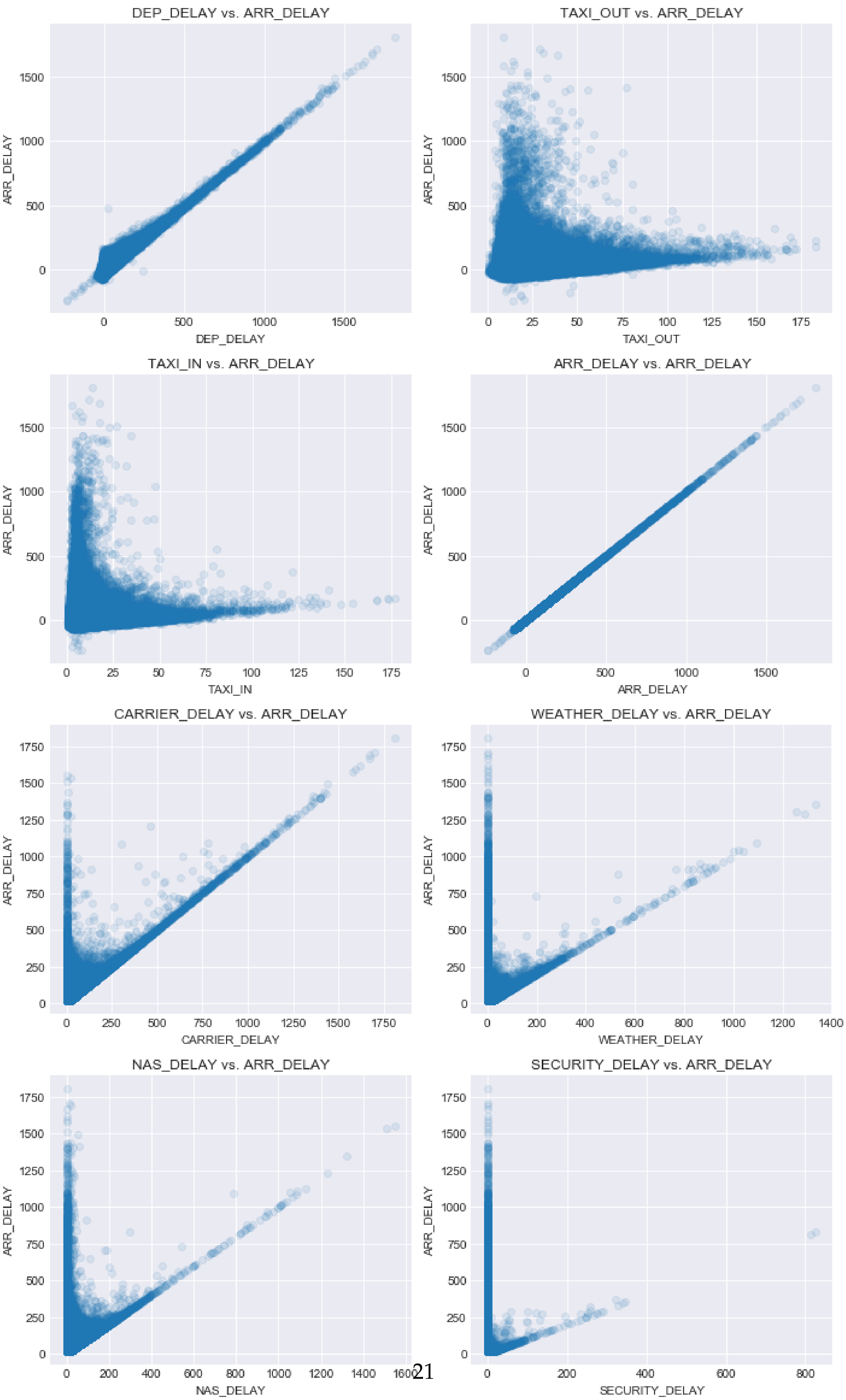
```
In [32]: # Draw scatter plots of numerical columns departures
         def draw_scatters_dep(df, variables, n_rows, n_cols):
             fig=plt.figure(figsize=(10,20))
             for i, var_name in enumerate(variables):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 sns.regplot(x=var_name,y='DEP_DELAY',data=df,fit_reg=False,scatter_kws={'alpha
                 ax.set_title(var_name +" vs. DEP_DELAY")
             fig.tight_layout()
             plt.savefig('../assets/png/07-scatters-departures.png')
             plt.show()
```

```
In [33]: draw_scatters_dep(df, numeric_columns[:-1], int(len(numeric_columns[:-1])/2)+1, 2)
```


DEP_DELAY vs. DEP_DELAY
DEP_DELAY
DEP_DELAY
0
500
1000
1500
TAXI_OUT vs. DEP_DELAY
TAXI_OUT
0
25
50
75
100
125
150
175
TAXI_IN vs. DEP_DELAY
TAXI_IN
ARR_DELAY vs. DEP_DELAY
ARR_DELAY
CARRIER_DELAY vs. DEP_DELAY
CARRIER_DELAY
250
750
1250
1750
WEATHER_DELAY vs. DEP_DELAY
WEATHER_DELAY
200
400
600
800
1200
1400
NAS_DELAY vs. DEP_DELAY
NAS_DELAY
1600
SECURITY_DELAY vs. DEP_DELAY
SECURITY_DELAY

```
In [34]: # Draw scatter plots of numerical columns arrivals
         def draw_scatters_arr(df, variables, n_rows, n_cols):
             fig=plt.figure(figsize=(10,20))
             for i, var_name in enumerate(variables):
                 ax=fig.add_subplot(n_rows,n_cols,i+1)
                 sns.regplot(x=var_name,y='ARR_DELAY',data=df,fit_reg=False,scatter_kws={'alpha
                 ax.set_title(var_name +" vs. ARR_DELAY")
             fig.tight_layout()
             plt.savefig('../assets/png/08-scatters-arrivals.png')
             plt.show()
```

```
In [35]: draw_scatters_arr(df, numeric_columns[:-1], int(len(numeric_columns[:-1])/2)+1, 2)
```


DEP_DELAY vs. ARR_DELAY
ARR_DELAY
DEP_DELAY
-0
500
1000
1500
0
500
1000
1500
TAXI_OUT vs. ARR_DELAY
ARR_DELAY
TAXI_OUT
0
500
1000
1500
0
25
50
75
100
125
150
175
TAXI_IN vs. ARR_DELAY
ARR_DELAY
TAXI_IN
0
500
1000
1500
0
25
50
75
100
125
150
175
ARR_DELAY vs. ARR_DELAY
ARR_DELAY
ARR_DELAY
0
500
1000
1500
0
500
1000
1500
CARRIER_DELAY vs. ARR_DELAY
ARR_DELAY
CARRIER_DELAY
0
250
500
750
1000
1250
1500
1750
0
250
500
750
1000
1250
1500
1750
WEATHER_DELAY vs. ARR_DELAY
ARR_DELAY
WEATHER_DELAY
0
250
500
750
1000
1250
1500
1750
0
200
400
600
800
1000
1200
1400
NAS_DELAY vs. ARR_DELAY
ARR_DELAY
NAS_DELAY
0
250
500
750
1000
1250
1500
1750
0
200
400
600
800
1000
1200
1400
1600
SECURITY_DELAY vs. ARR_DELAY
ARR_DELAY
SECURITY_DELAY
0
250
500
750
1000
1250
1500
1750
0
200
400
600
800

```
In [36]: # Pickle DataFrame
         df.to_pickle('../data/df.p')
```