(Flights Dataset Exploration)

by (Abdelrahman Nasr)

Preliminary Wrangling

> This dataset reports flights in the United States, including carriers, arrival and departure delays, and reasons for delays, from 2010 to 2020.

In [1]:
```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.colors import to_rgb
import seaborn as sb

%matplotlib inline
```

In [2]:
```
# load in the dataset into a pandas dataframe, print statistics
df = pd.read_csv('flights dataset.csv')
```

In [3]:
```
# high-level overview of data shape and composition
print(df.shape)
print(df.dtypes)
```

```
(159766, 22)
year                    int64
 month                  int64
carrier                object
carrier_name           object
airport                object
airport_name           object
arr_flights           float64
arr_del15             float64
carrier_ct            float64
 weather_ct           float64
nas_ct                float64
security_ct           float64
late_aircraft_ct      float64
arr_cancelled         float64
arr_diverted          float64
 arr_delay            float64
 carrier_delay        float64
weather_delay         float64
nas_delay             float64
security_delay        float64
late_aircraft_delay   float64
Unnamed: 21           float64
dtype: object
```

In [4]:
```
df.head(10)
```

Out[4]:

	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15	carrier_ct	weather_ct	...	late_aircraft_ct	arr_cancelled	arr_
0	2011	12	DL	Delta Air Lines Inc.	STL	St. Louis, MO: St Louis Lambert International	396.0	34.0	13.90	0.68	...	12.89	0.0	
1	2011	12	DL	Delta Air Lines Inc.	STT	Charlotte Amalie, VI: Cyril E King	42.0	4.0	3.09	0.00	...	0.00	0.0	
2	2011	12	DL	Delta Air Lines Inc.	STX	Christiansted, VI: Henry E. Rohlsen	3.0	0.0	0.00	0.00	...	0.00	0.0	
3	2011	12	DL	Delta Air Lines Inc.	SYR	Syracuse, NY: Syracuse Hancock International	55.0	5.0	2.50	0.00	...	1.00	0.0	
4	2011	12	DL	Delta Air Lines Inc.	TLH	Tallahassee, FL: Tallahassee International	31.0	5.0	1.25	0.00	...	2.74	0.0	
5	2011	12	DL	Delta Air Lines Inc.	TPA	Tampa, FL: Tampa International	893.0	90.0	30.77	1.37	...	33.12	3.0	

6	2011	12	DL	Delta Air Lines Inc.	TUL	Tulsa, OK: Tulsa International	28.0	3.0	1.00	0.00	...	0.87	0.0
7	2011	12	DL	Delta Air Lines Inc.	TUS	Tucson, AZ: Tucson International	89.0	18.0	4.26	0.24	...	6.70	0.0
8	2011	12	DL	Delta Air Lines Inc.	TYS	Knoxville, TN: McGhee Tyson	28.0	6.0	1.38	0.00	...	3.82	0.0
9	2011	12	DL	Delta Air Lines Inc.	VPS	Valparaiso, FL: Eglin AFB Destin Fort Walton B...	26.0	2.0	1.00	0.00	...	0.44	0.0

10 rows × 22 columns

In [5]:
```
df.describe()
```

Out[5]:

	year	month	arr_flights	arr_del15	carrier_ct	weather_ct	nas_ct	security_ct	late_aircraft_ct
count	159766.00000	159766.000000	159531.000000	159335.000000	159531.000000	159531.000000	159531.000000	159531.000000	159531.000000
mean	2015.63747	6.527353	385.477149	69.258104	19.932764	2.145940	21.490234	0.131169	25.472960
std	3.04247	3.470750	1030.703698	182.280781	44.974315	6.718006	66.755794	0.578655	74.978867
min	2010.00000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2013.00000	3.000000	55.000000	7.000000	2.450000	0.000000	1.400000	0.000000	1.710000
50%	2016.00000	7.000000	115.000000	20.000000	6.950000	0.360000	4.700000	0.000000	5.900000
75%	2018.00000	10.000000	274.000000	51.000000	17.890000	1.800000	13.490000	0.000000	16.990000
max	2020.00000	12.000000	21977.000000	5268.000000	1242.160000	295.280000	2401.790000	26.070000	1849.000000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 159766 entries, 0 to 159765
Data columns (total 22 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   year                 159766 non-null  int64  
 1    month               159766 non-null  int64  
 2   carrier              159766 non-null  object 
 3   carrier_name         159766 non-null  object 
 4   airport              159766 non-null  object 
 5   airport_name         159766 non-null  object 
 6   arr_flights          159531 non-null  float64
 7   arr_del15            159335 non-null  float64
 8   carrier_ct           159531 non-null  float64
 9    weather_ct          159531 non-null  float64
 10  nas_ct               159531 non-null  float64
 11  security_ct          159531 non-null  float64
 12  late_aircraft_ct     159531 non-null  float64
 13  arr_cancelled        159531 non-null  float64
 14  arr_diverted         159531 non-null  float64
 15   arr_delay           159531 non-null  float64
 16   carrier_delay       159531 non-null  float64
 17  weather_delay        159531 non-null  float64
 18  nas_delay            159531 non-null  float64
 19  security_delay       159531 non-null  float64
 20  late_aircraft_delay  159531 non-null  float64
 21  Unnamed: 21          0 non-null       float64
dtypes: float64(16), int64(2), object(4)
memory usage: 26.8+ MB
```

The data need some simple cleaning

In [7]:
```
# drop the last column and drop the nan values
df = df.iloc[:, :-1]
df.dropna()
```

Out[7]:

	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15	carrier_ct	weather_ct	...	security_ct	late_aircraft_ct
				Delta Air		St. Louis, MO: St Louis							

0	2011	12	DL	Lines Inc.	STL	Lambert International	396.0	34.0	13.90	0.68	...	0.0	12.89
1	2011	12	DL	Delta Air Lines Inc.	STT	Charlotte Amalie, VI: Cyril E King	42.0	4.0	3.09	0.00	...	0.0	0.00
2	2011	12	DL	Delta Air Lines Inc.	STX	Christiansted, VI: Henry E. Rohlsen	3.0	0.0	0.00	0.00	...	0.0	0.00
3	2011	12	DL	Delta Air Lines Inc.	SYR	Syracuse, NY: Syracuse Hancock International	55.0	5.0	2.50	0.00	...	0.0	1.00
4	2011	12	DL	Delta Air Lines Inc.	TLH	Tallahassee, FL: Tallahassee International	31.0	5.0	1.25	0.00	...	0.0	2.74
...	...	...	...	...	...	...	...	...	...	...	...	...	...
159761	2019	1	MQ	Envoy Air	RIC	Richmond, VA: Richmond International	195.0	68.0	12.12	1.87	...	0.0	36.04
159762	2019	1	MQ	Envoy Air	ROA	Roanoke, VA: Roanoke Blacksburg Regional Woodr...	52.0	14.0	2.74	0.69	...	0.0	8.11
159763	2019	1	MQ	Envoy Air	ROC	Rochester, NY: Greater Rochester International	106.0	26.0	4.67	2.26	...	0.0	7.26
159764	2019	1	MQ	Envoy Air	RST	Rochester, MN: Rochester International	116.0	35.0	6.83	6.92	...	0.0	9.75
159765	2019	1	MQ	Envoy Air	SAT	San Antonio, TX: San Antonio International	26.0	4.0	1.16	0.64	...	0.0	0.29

159335 rows × 21 columns

In [8]:
```
# remove space from columns names
df.columns = df.columns.str.lstrip()
```

In [9]:
```
df.columns
```

Out[9]:
```
Index(['year', 'month', 'carrier', 'carrier_name', 'airport', 'airport_name',
       'arr_flights', 'arr_del15', 'carrier_ct', 'weather_ct', 'nas_ct',
       'security_ct', 'late_aircraft_ct', 'arr_cancelled', 'arr_diverted',
       'arr_delay', 'carrier_delay', 'weather_delay', 'nas_delay',
       'security_delay', 'late_aircraft_delay'],
      dtype='object')
```

What is the structure of your dataset?

> There are 159335 flights information in the dataset with each entry lists the number of flights, the number of flights delayed, the number of flights canceled and diverted, the minutes of delay due to (carrier-weather-national air system-security) and finally the sum of total delay minutes.

What is/are the main feature(s) of interest in your dataset?

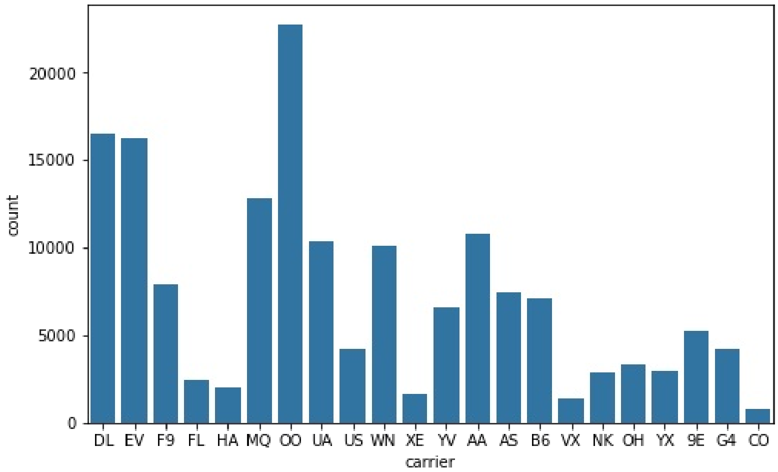
> I'm most interested in figuring out what features are best for predicting the flight being delayed, canceled or diverted.

What features in the dataset do you think will help support your investigation into your feature(s) of interest?

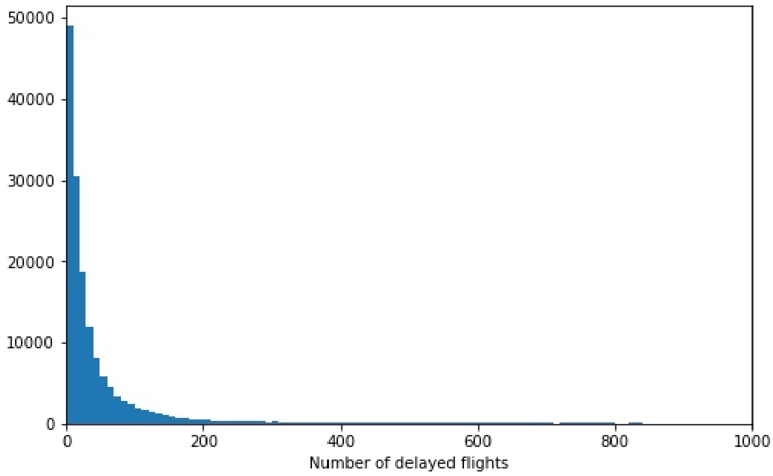
> I expect that carrier will have the strongest effect on the numbers of flights. I also think that the other time data will have effects on the numbers.

Univariate Exploration

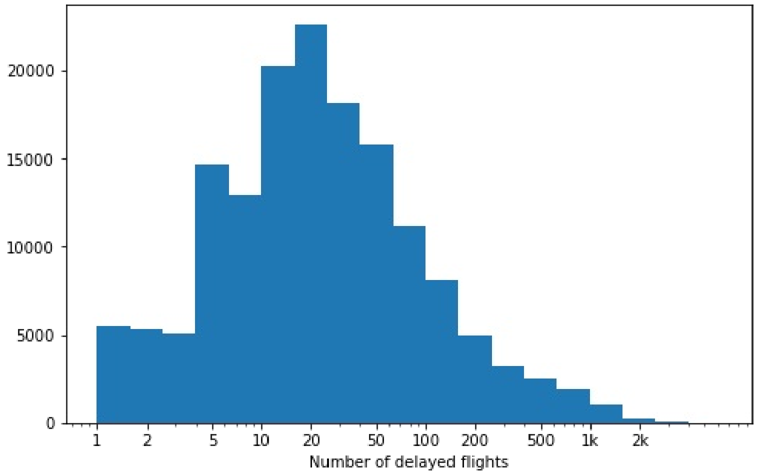
In [10]:
```python
# plot the casrrier qualitative variable to get an idea of its distribution.
default_color = sb.color_palette()[0]
plt.figure(figsize=[8, 5])
sb.countplot(data = df, x = 'carrier', color = default_color);
```



In [11]:
```python
# start with a standard-scaled plot for the number of delayed flights
binsize = 10
bins = np.arange(0, df.arr_del15.max()+binsize, binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_del15, bins = bins)
plt.xlim(0,1000)
plt.xlabel('Number of delayed flights');
```

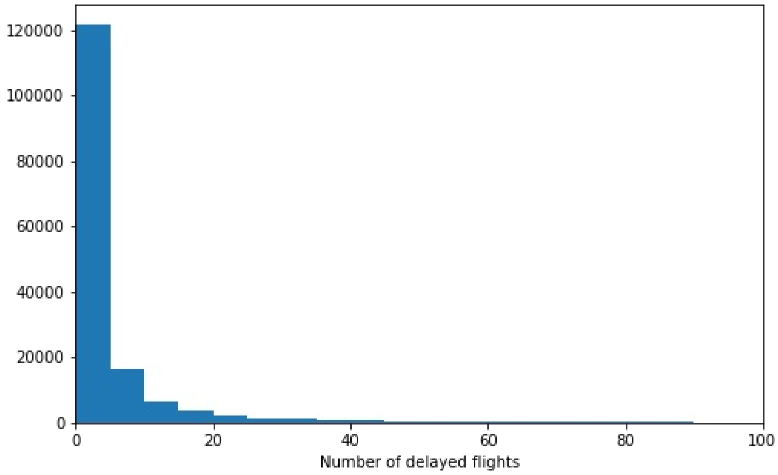


In [12]:
```python
# let's put it on a log scale instead
log_binsize = 0.2
bins = 10 ** np.arange(0, np.log10(df.arr_del15.max())+log_binsize, log_binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_del15, bins = bins)
plt.xscale('log')
plt.xticks([1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000], [1, 2, 5, 10, 20, 50, 100, 200, 500, '1k', '2k'])
plt.xlabel('Number of delayed flights');
```

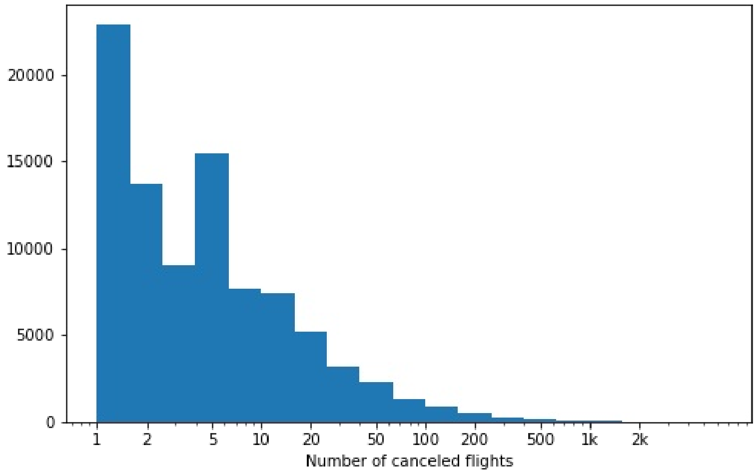


number of delayed flights has a long-tailed distribution, with a lot of numbers on the low section end, and few on the high section end. When plotted on a log-scale, the distribution looks roughly unimodal, with one peak a little above 20.

In [13]:
```
# start with a standard-scaled plot for the number of canceled flights
binsize = 5
bins = np.arange(0, df.arr_cancelled.max()+binsize, binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_cancelled, bins = bins)
plt.xlim(0,100)
plt.xlabel('Number of delayed flights');
```

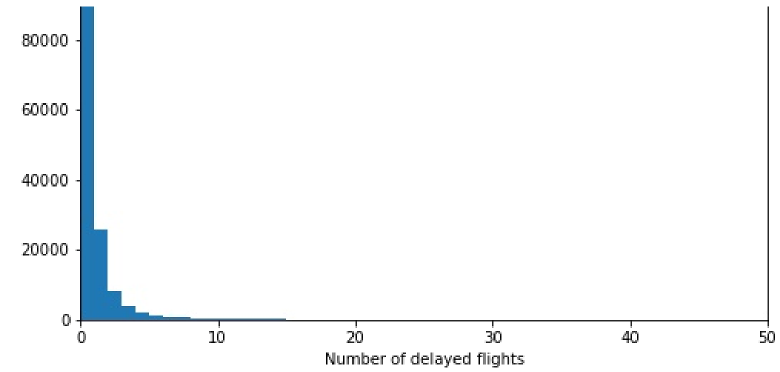


In [14]:
```
# let's try the same approach for the number of canceled flights
log_binsize = 0.2
bins = 10 ** np.arange(0, np.log10(df.arr_cancelled.max())+log_binsize, log_binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_cancelled, bins = bins)
plt.xscale('log')
plt.xticks([1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000], [1, 2, 5, 10, 20, 50, 100, 200, 500, '1k', '2k'])
plt.xlabel('Number of canceled flights');
```

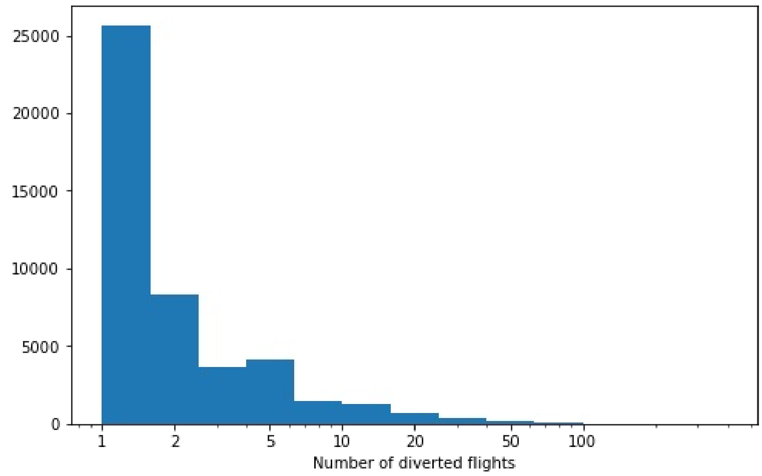


number of canceled flights has a long-tailed distribution, with a lot of numbers on the low section end, and few on the high section end the same as the number of delayed flights. When plotted on a log-scale, the distribution looks roughly bimodal, right-skewed with one peak a little above 1 and another above 5.

In [15]:
```
# start with a standard-scaled plot for the number of diverted flights
binsize = 1
bins = np.arange(0, df.arr_diverted.max()+binsize, binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_diverted, bins = bins)
plt.xlim(0,50)
plt.xlabel('Number of delayed flights');
```

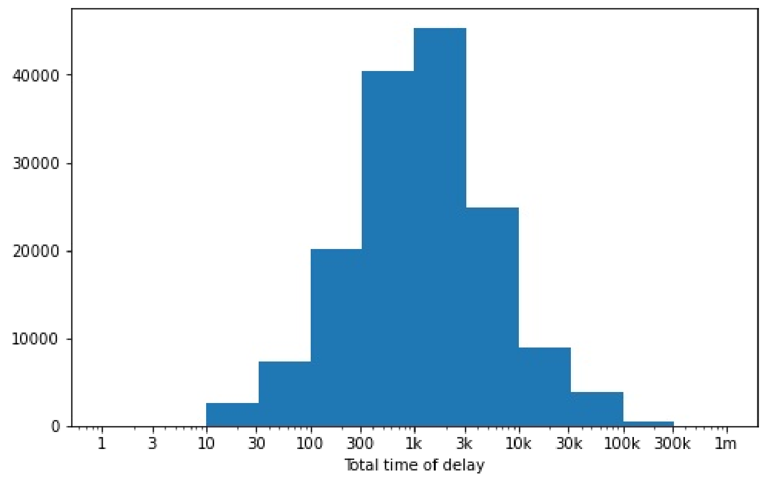


In [16]:
```
# again with the same approach for the number of diverted flights
log_binsize = 0.2
bins = 10 ** np.arange(0, np.log10(df.arr_diverted.max())+log_binsize, log_binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_diverted, bins = bins)
plt.xscale('log')
plt.xticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100])
plt.xlabel('Number of diverted flights');
```



the same as before but the number of diverted flights distribution on the log scale is still roughly skewed to the right

In [17]:
```
# now let's see the total delay distribution
log_binsize = 0.5
bins = 10 ** np.arange(0, np.log10(df.arr_delay.max())+log_binsize, log_binsize)

plt.figure(figsize=[8, 5])
plt.hist(df.arr_delay, bins = bins)
plt.xscale('log')
plt.xticks([1, 3, 10, 30, 100, 300, 1000, 3e3, 1e4, 3e4, 1e5, 3e5, 1e6], [1, 3, 10, 30, 100, 300, '1k', '3k', '1
plt.xlabel('Total time of delay');
```



Interestingly the distribution of the total time of delay on the log scale has a normal distribution shape with a mean around 1k.

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

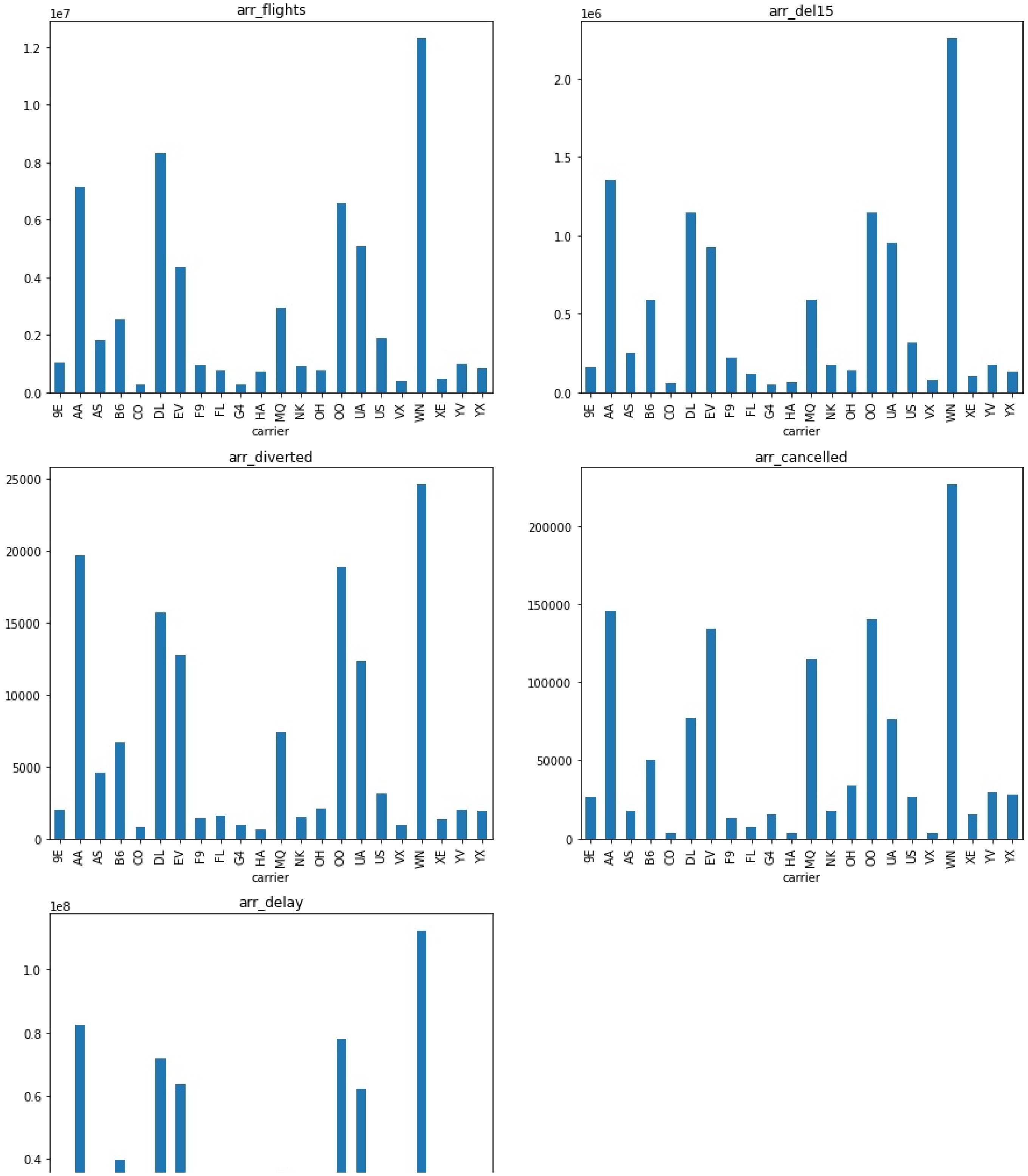
> I needed to preforme a log transformation to see the distribution more clearly.

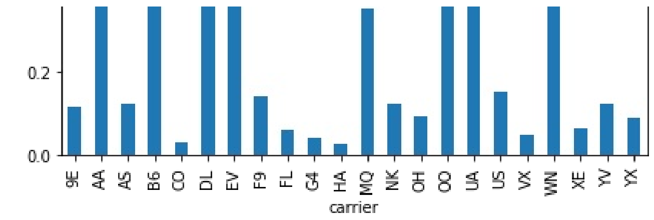
Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

> no.

Bivariate Exploration

In [18]:
```
# Let's see the distributions again but after grouping by the carrier
count_columns = ['arr_flights','arr_del15','arr_diverted', 'arr_cancelled', 'arr_delay']
fig, ax = plt.subplots(ncols = 2, nrows = 3 , figsize = [15,20])
axe = ax.ravel()
axe[5].axis('off')

for idx, column in enumerate(count_columns):
    df.groupby('carrier')[column].sum().plot(kind='bar', title=column, ax=axe[idx]);
```

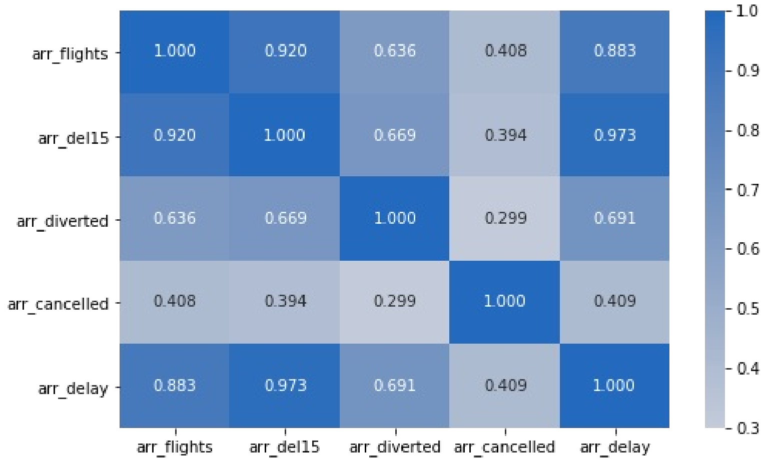




It seems that Southwest Airlines Co. carrier (WN) really stands out in every case which requires feature engineering to really see the distribution.

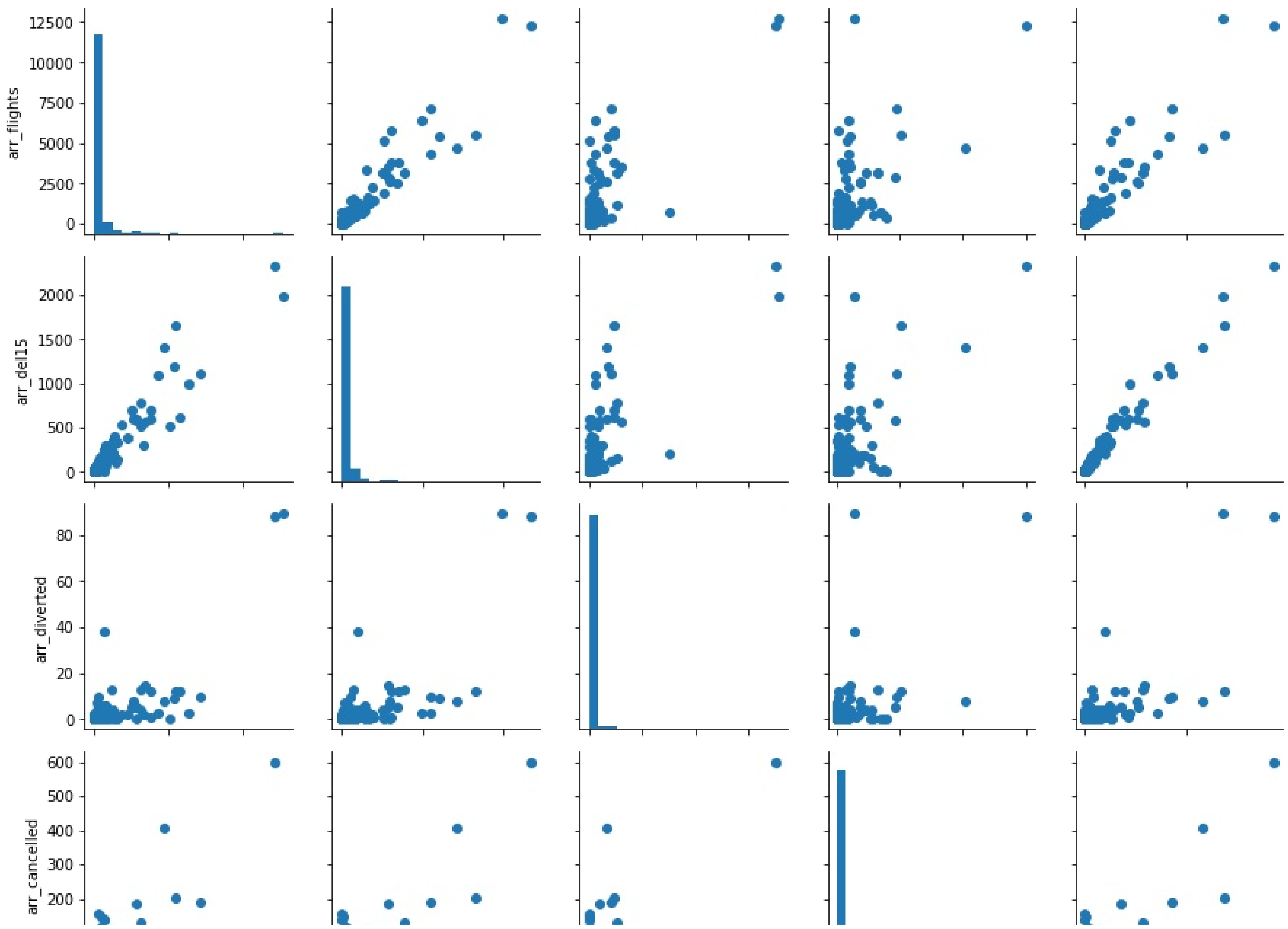
In [19]:

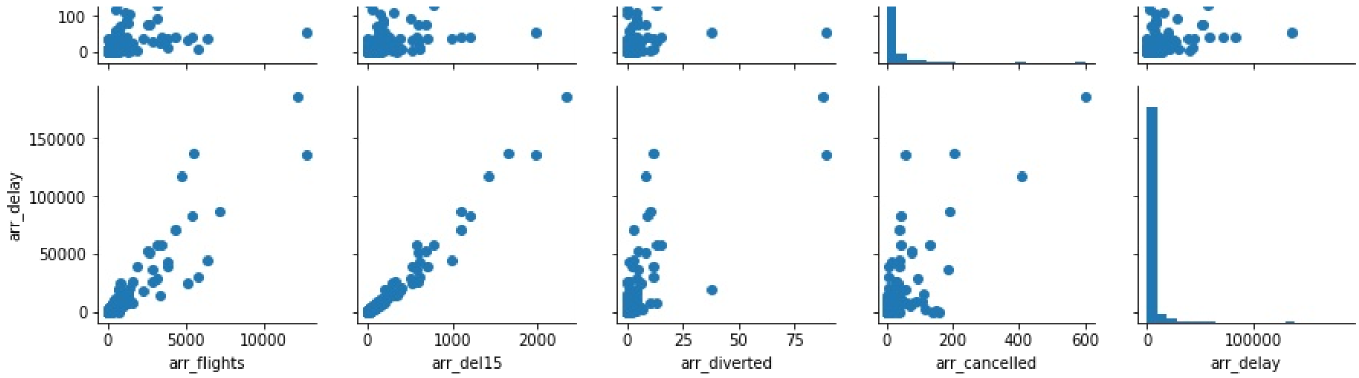
```
# let's now see the correlation between the different numbers of flights and the total delay time
plt.figure(figsize = [8, 5])
sb.heatmap(df[count_columns].corr(), annot = True, fmt = '.3f', cmap = 'vlag_r', center = 0);
```



In [20]:

```
# plot matrix: sample 500 flights
samples = np.random.choice(df.shape[0], 500, replace = False)
df_samp = df.loc[samples,:]

g = sb.PairGrid(data = df_samp, vars = count_columns)
g = g.map_diag(plt.hist, bins = 20);
g.map_offdiag(plt.scatter);
```

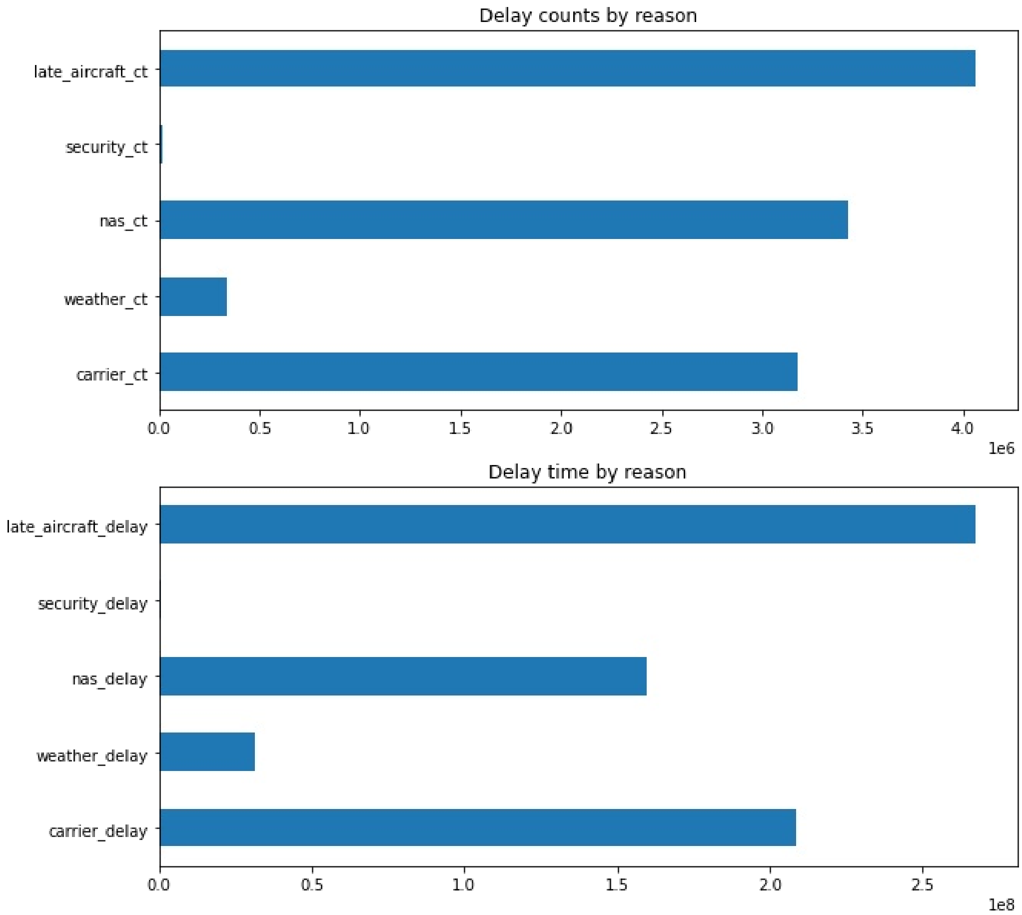




As expected, there are two strong correlations stands out between the total time of delay and the number of delayed flights also between the the number delayed flights and the number of arrived flights.

Next I'm gonna look deeper in the reasons of delay:

In [21]:
```
delay_counts = ['carrier_ct', 'weather_ct', 'nas_ct', 'security_ct', 'late_aircraft_ct']
delay_time = ['carrier_delay', 'weather_delay', 'nas_delay', 'security_delay', 'late_aircraft_delay']
fig, ax = plt.subplots(nrows = 2 , figsize = [10,10])

(df[delay_counts].sum()).plot(kind='barh', title='Delay counts by reason', ax=ax[0])
(df[delay_time].sum()).plot(kind='barh', title='Delay time by reason', ax=ax[1]);
```



It's clear to see that the late aircraft reason has the hightest numbers while the security reason has the lowest in both counts and time terms.

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

> All the features distributions with the carrier seem to have the same shape while the delay reasons have a clear distribution.

Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

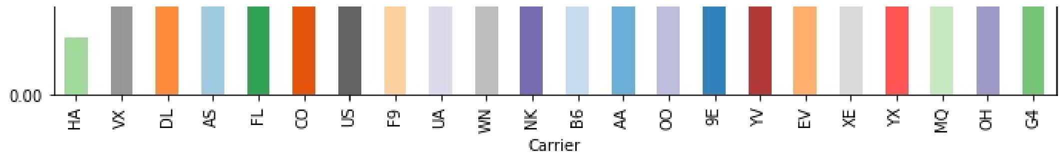
> The features are related to the total number of flights for each carrier not just the carrier itself which can be seen better with feature engineering also for the delay reasons.

Multivariate Exploration

In [22]:
```python
# make a color palette for the carriers
carriers = list(df.carrier.unique())
carriers.sort()
color = sb.color_palette('tab20c')
color.extend([to_rgb('#b33939'), to_rgb('#ff5252')])
colours = {carriers[i]: color[i] for i in range(len(carriers))}
```

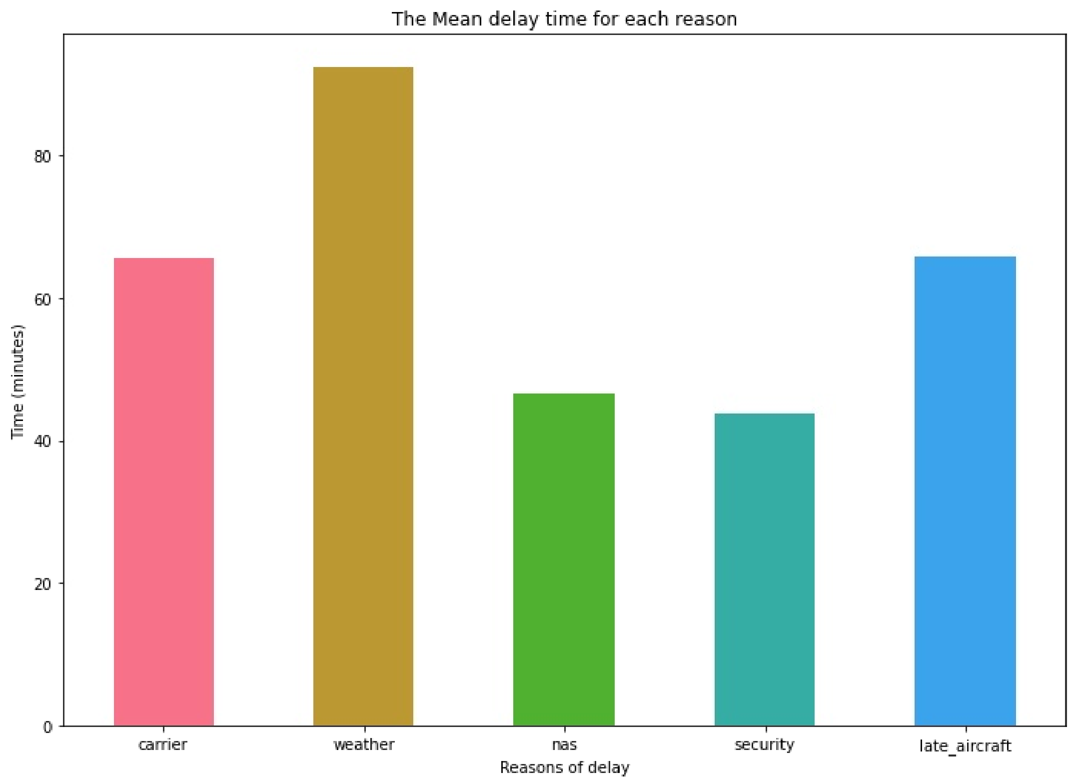
In [23]:
```python
# The percentage of the number of delayed and diverted flights to the total number of arrived flights for each ca
plt.figure(figsize=[11.69, 8.27])
s1 = ((df.groupby('carrier').arr_del15.sum() + df.groupby('carrier').arr_diverted.sum()) / df.groupby('carrier').
s1.plot(kind='bar', color=s1.index.map(colours))
plt.title('The percentage of the number of delayed and diverted flights to the total number of arrived flights fo
plt.xlabel('Carrier')
plt.ylabel('Proportion (%)');
```

In [24]:
```python
# Proportion between the number of canceled flights to the total number of arrived flights for each carrier visua
plt.figure(figsize=[11.69, 8.27])
s2 = (df.groupby('carrier').arr_cancelled.sum() / df.groupby('carrier').arr_flights.sum()).sort_values()
s2.plot(kind='bar', color=s2.index.map(colours))
plt.title('Proportion between the number of canceled flights to the total number of arrived flights for each carr
plt.xlabel('Carrier')
plt.ylabel('Proportion (%)');
```



At the end the result shows that the JetBlue Airways carrier (B6) has the highest proportion of delayed and diverted flights, the Allegiant Air carrier (G4) has the hightest proportions in canceled flights and Hawaiian Airlines Inc. carrier has the lowest proportions in all.

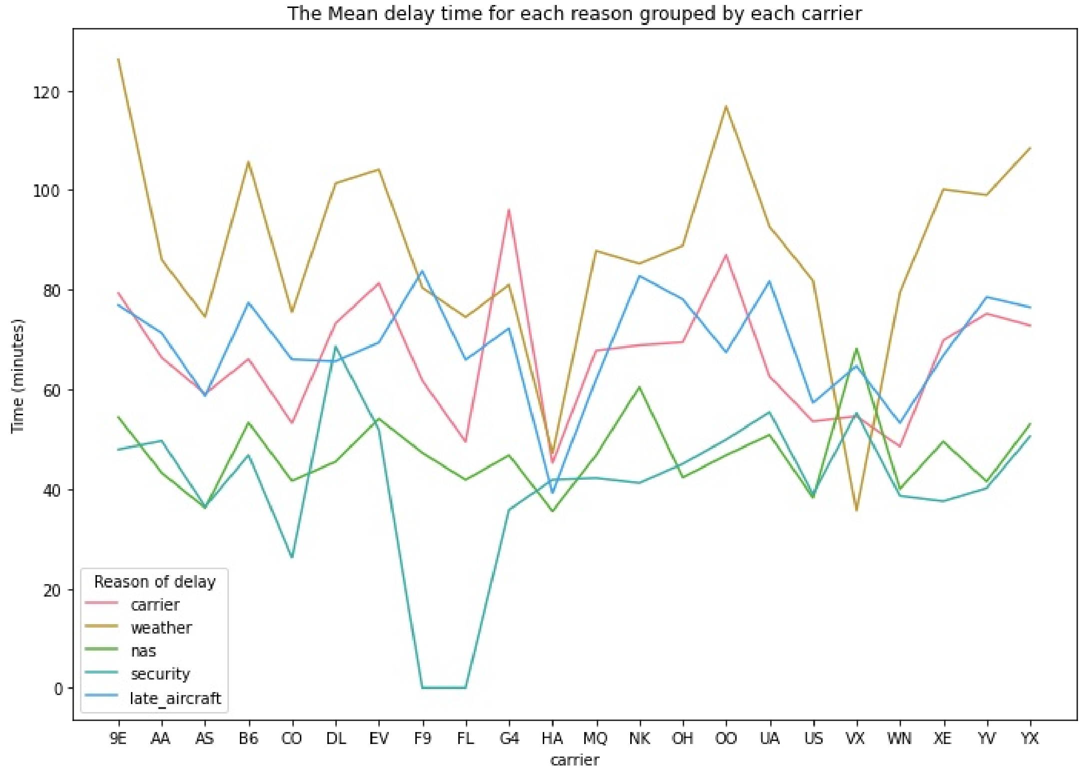
In [25]:
```
# Proportion between the total time of delay to the counts of delayed flights for each reason
reasons_names = ['carrier', 'weather', 'nas', 'security', 'late_aircraft']
# change the two series indexes name to match
a = df[delay_time].sum()
reasons = {a.index[i]: reasons_names[i] for i in range(len(reasons_names))}
a.rename(index=reasons, inplace=True)

b = df[delay_counts].sum()
reasons = {b.index[i]: reasons_names[i] for i in range(len(reasons_names))}
b.rename(index=reasons, inplace=True)
# plot the visual
plt.figure(figsize=[11.69, 8.27])
(np.divide(a, b)).plot(kind='bar', color=sb.color_palette("husl"), rot=0)
plt.title('The Mean delay time for each reason')
plt.xlabel('Reasons of delay')
plt.ylabel('Time (minutes)');
```



Interestingly, the weather has the highest mean not the late aircraft like we expected from the visuals before.

In [26]:
```
# Proportion between the total time of delay to the counts of delayed flights for each reason grouped by each car
reasons_names = ['carrier', 'weather', 'nas', 'security', 'late_aircraft']
# change the two dataframes columns name to match
a = df.groupby('carrier')[delay_time].sum()
reasons = {a.columns[i]: reasons_names[i] for i in range(len(reasons_names))}
a.rename(columns=reasons, inplace=True)

b = df.groupby('carrier')[delay_counts].sum()
reasons = {b.columns[i]: reasons_names[i] for i in range(len(reasons_names))}
b.rename(columns=reasons, inplace=True)
# plot the visual
ax = (a.div(b).fillna(0)).plot(figsize=[11.69, 8.27], color=sb.color_palette("husl"))
ax.set_xticks(np.arange(0, 22, 1))
ax.set_xticklabels(a.index)
plt.legend(title='Reason of delay')
plt.ylabel('Time (minutes)')
plt.title('The Mean delay time for each reason grouped by each carrier');
```

```
C:\Users\kasr1\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:1235: UserWarning: FixedFormatter
should only be used together with FixedLocator
  ax.set_xticklabels(xticklabels)
```



Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

> Indeed the number of arrived flights for each carrier changed the pattern seen before so did the comapring the mean delay time for each reason.

Were there any interesting or surprising interactions between features?

> Southwest Airlines Co. carrier (WN) doesn't stand out anymore and late aircraft doesn't have the hightest mean of delay time.