Visualization With Matplotlib and Pandas

Recipes

- Getting started with matplotlib
- Plotting basics with pandas
- Visualizing the flights dataset

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

(1) Getting Started With matplotlib

Getting Ready: Hierarchy of Matplotlib Objects

- "Matplotlib uses a hierarchy of objects to display all of its plotting items in the output. This hierarchy is key to understanding everything about matplotlib. The Figure and Axes objects are the two main components of the hierarchy."
- "The Figure object is at the top of the hierarchy. It is the container for everything that will be plotted."
- "Contained within the Figure is one or more Axes object(s). The Axes is the primary object that you will interact with when using matplotlib and can be more commonly thought of as the actual plotting surface. The Axes contains the x/y axis, points, lines, markers, labels, legends, and any other useful item that is plotted."

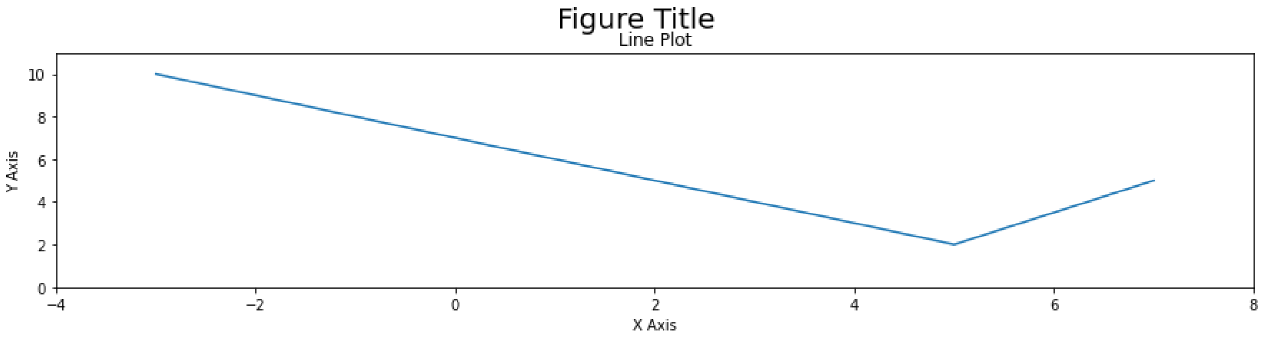
MATLAB-Like Stateful Interface

In [2]:

```
x = [-3, 5, 7]
y = [10, 2, 5]

plt.figure(figsize = (15, 3))
plt.plot(x, y)
plt.xlim(-4, 8)
plt.ylim(0, 11)
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.title('Line Plot')
plt.suptitle('Figure Title', size = 20, y = 1.03)
```

Out[2]:

```
Text(0.5, 1.03, 'Figure Title')
```



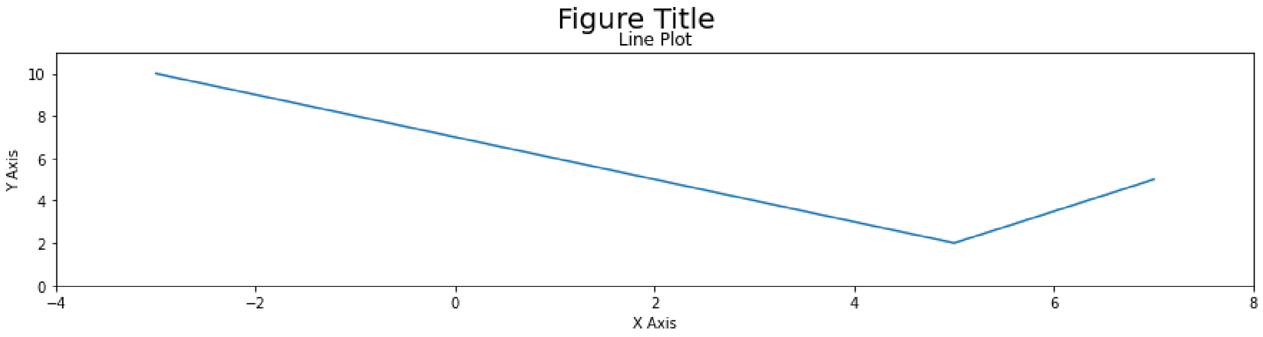
Object-Oriented Interface

In [3]:

```
x = [-3, 5, 7]
y = [10, 2, 5]

fig, ax = plt.subplots(figsize = (15, 3))
ax.plot(x, y)
ax.set_xlim(-4, 8)
ax.set_ylim(0, 11)
ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')
ax.set_title('Line Plot')
fig.suptitle('Figure Title', size = 20, y = 1.03)
```

Out[3]:

```
Text(0.5, 1.03, 'Figure Title')
```



In [4]:

```
type(fig)
```

Out[4]:

```
matplotlib.figure.Figure
```

In [5]:

```
type(ax)
```

Out[5]:

```
matplotlib.axes._subplots.AxesSubplot
```

In [6]:

```
dir(fig)
```

Out[6]:

```
['__class__',
 '__delattr__',
 '__dict__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getstate__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__module__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__setstate__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 '__weakref__',
 '_add_axes_internal',
 '_agg_filter',
 '_align_xlabel_grp',
 '_align_ylabel_grp',
 '_alpha',
 '_animated',
 '_axobservers',
 '_axstack',
 '_cachedRenderer',
 '_clipon',
 '_clippath',
 '_constrained',
 '_constrained_layout_pads',
 '_contains',
 '_default_contains',
 '_dpi',
 '_gci',
 '_get_clipping_extent_bbox',
 '_get_dpi',
 '_gid',
 '_gridspecs',
 '_in_layout',
 '_label',
 '_layoutbox',
 '_make_key',
 '_mouseover',
 '_normalize_grid_string',
 '_oid',
 '_path_effects',
 '_picker',
 '_process_projection_requirements',
 '_propobservers',
 '_rasterized',
 '_remove_method',
 '_repr_html_',
 '_set_artist_props',
 '_set_dpi',
 '_set_gc_clip',
 '_sketch',
 '_snap',
 '_stale',
 '_sticky_edges',
 '_suptitle',
 '_tight',
 '_tight_parameters',
 '_transform',
 '_transformSet',
 '_url',
 '_visible',
 'add_artist',
 'add_axes',
 'add_axobserver',
 'add_callback',
 'add_gridspec',
 'add_subplot',
 'align_labels',
 'align_xlabels',
 'align_ylabels',
 'artists',
 'autofmt_xdate',
 'axes',
 'bbox',
 'bbox_inches',
 'callbacks',
 'canvas',
 'clear',
 'clf',
 'clipbox',
 'colorbar',
 'contains',
 'convert_xunits',
 'convert_yunits',
 'delaxes',
 'dpi',
 'dpi_scale_trans',
 'draw',
 'draw_artist',
 'eventson',
 'execute_constrained_layout',
 'figimage',
 'figure',
 'findobj',
 'format_cursor_data',
 'frameon',
 'gca',
 'get_agg_filter',
 'get_alpha',
 'get_animated',
 'get_axes',
 'get_children',
 'get_clip_box',
 'get_clip_on',
 'get_clip_path',
 'get_constrained_layout',
 'get_constrained_layout_pads',
 'get_contains',
 'get_cursor_data',
 'get_default_bbox_extra_artists',
 'get_dpi',
 'get_edgecolor',
 'get_facecolor',
 'get_figheight',
 'get_figure',
 'get_figwidth',
 'get_frameon',
 'get_gid',
 'get_in_layout',
 'get_label',
 'get_path_effects',
 'get_picker',
 'get_rasterized',
 'get_size_inches',
 'get_sketch_params',
 'get_snap',
 'get_tight_layout',
 'get_tightbbox',
 'get_transform',
 'get_transformed_clip_path_and_affine',
 'get_url',
 'get_visible',
 'get_window_extent',
 'get_zorder',
 'ginput',
 'have_units',
 'images',
 'init_layoutbox',
 'is_transform_set',
 'legend',
 'legends',
 'lines',
 'mouseover',
 'number',
 'patch',
 'patches',
 'pchanged',
 'pick',
 'pickable',
 'properties',
 'remove',
 'remove_callback',
 'savefig',
 'sca',
 'set',
 'set_agg_filter',
 'set_alpha',
 'set_animated',
 'set_canvas',
 'set_clip_box',
 'set_clip_on',
 'set_clip_path',
 'set_constrained_layout',
 'set_constrained_layout_pads',
 'set_contains',
 'set_dpi',
 'set_edgecolor',
 'set_facecolor',
 'set_figheight',
 'set_figure',
 'set_figwidth',
 'set_frameon',
 'set_gid',
 'set_in_layout',
 'set_label',
 'set_path_effects',
 'set_picker',
 'set_rasterized',
 'set_size_inches',
 'set_sketch_params',
 'set_snap',
 'set_tight_layout',
 'set_transform',
 'set_url',
 'set_visible',
 'set_zorder',
 'show',
 'stale',
 'stale_callback',
 'sticky_edges',
 'subplot_mosaic',
 'subplotpars',
 'subplots',
 'subplots_adjust',
 'suppressComposite',
 'suptitle',
 'text',
 'texts',
 'tight_layout',
 'transFigure',
 'update',
 'update_from',
 'waitforbuttonpress',
 'zorder']
```

In [7]:

```
fig.axes
```

Out[7]:

```
[<AxesSubplot:title={'center':'Line Plot'}, xlabel='X Axis', ylabel='Y Axis'>]
```

In [8]:

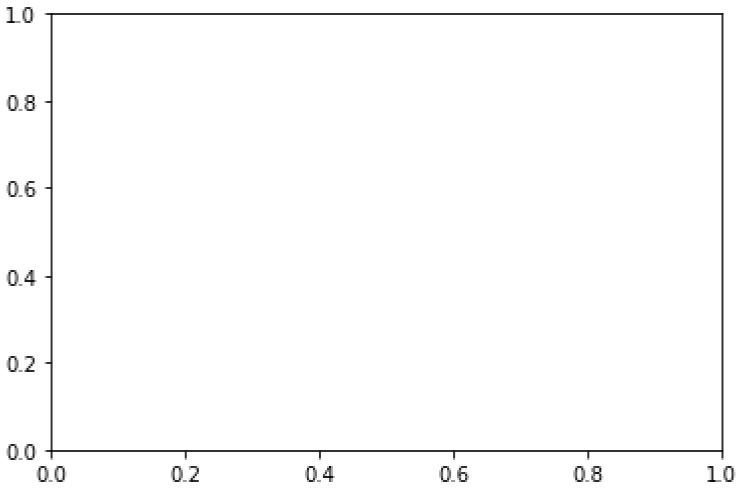
```
fig.axes[0] is ax
```

Out[8]:

```
True
```

In [9]:

```
plot_objects = plt.subplots()
```



In [10]:

```
type(plot_objects)
```

Out[10]:

```
tuple
```

In [11]:

```
len(plot_objects)
```

Out[11]:

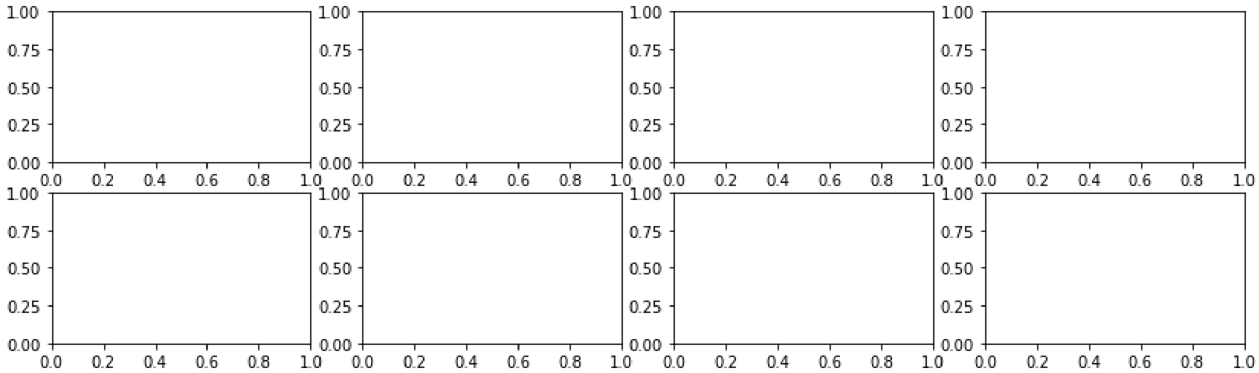
```
2
```

In [12]:

```
fig = plot_objects[0]
ax = plot_objects[1]
```

In [13]:

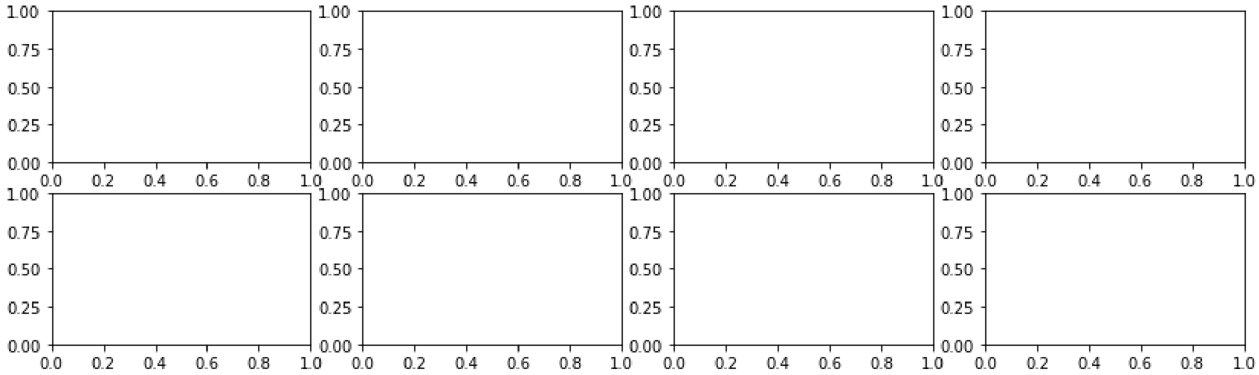
```
plot_objects = plt.subplots(2, 4, figsize = (14, 4))
```



In [14]:

```
plot_objects[0]
```

Out[14]:



In [15]:

```
plot_objects[1]
```

Out[15]:

```
array([[<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>],
       [<AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]],
      dtype=object)
```

In [16]:

```
plot_objects[1][0][1]
```

Out[16]:

```
<AxesSubplot:>
```

(2) Plotting Basics With Pandas

In [ ]:

```
from IPython.display import HTML
HTML('<iframe src=https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Dat
```

In [17]:

```
df = pd.DataFrame({'Apples': [20, 10, 40, 20, 50], 'Oranges': [35, 40, 25, 19, 33]}, in
df
```

Out[17]:

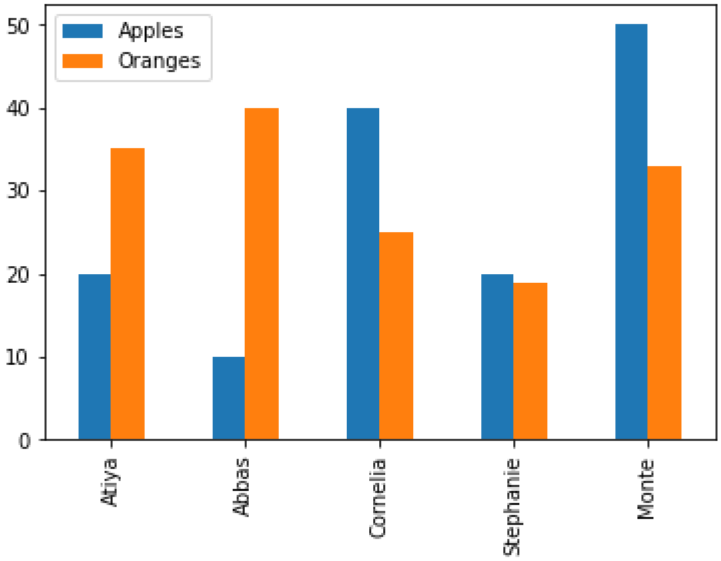
	Apples	Oranges
Atiya	20	35
Abbas	10	40
Cornelia	40	25
Stephanie	20	19
Monte	50	33

In [18]:

```
df.plot(kind = 'bar')
```

Out[18]:

```
<AxesSubplot:>
```

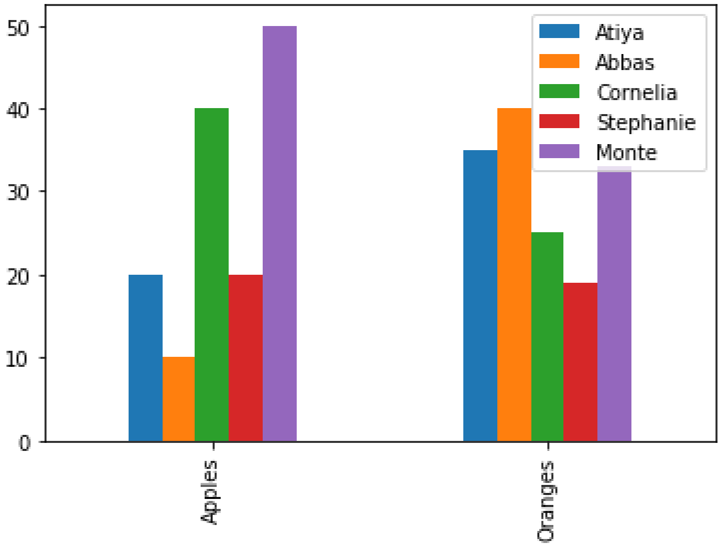


In [19]:

```
df.transpose().plot(kind = 'bar')
```

Out[19]:

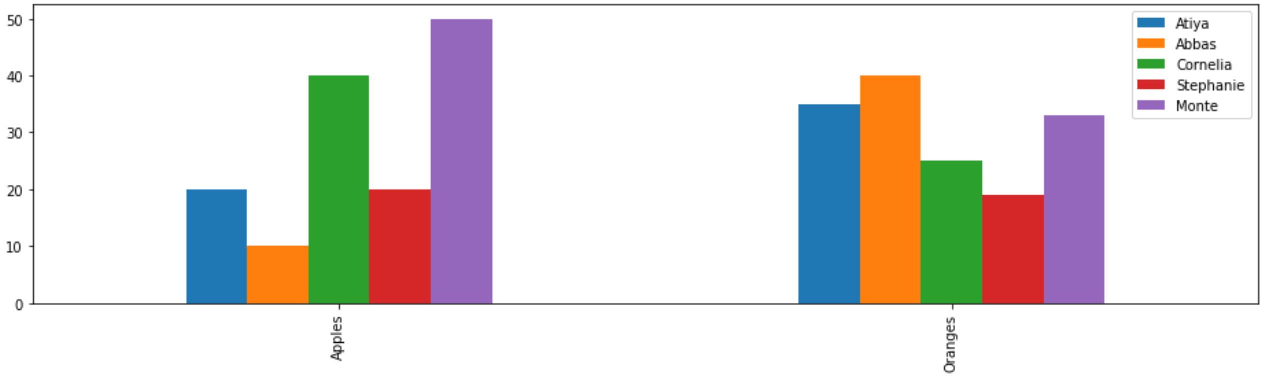
```
<AxesSubplot:>
```



In [20]:

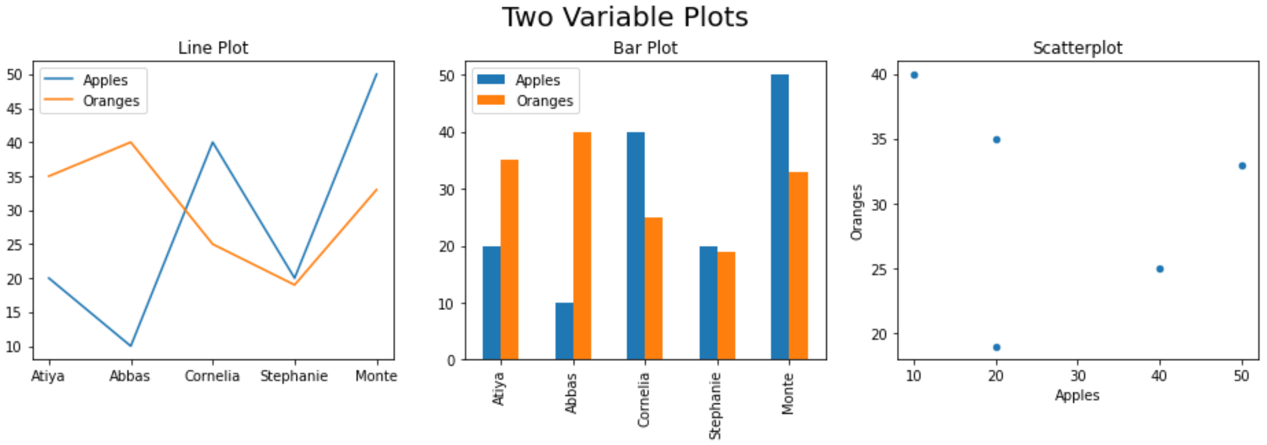
```
df.transpose().plot(kind = 'bar', figsize = (16, 4))
```

```
Out[20]: <AxesSubplot:>
```



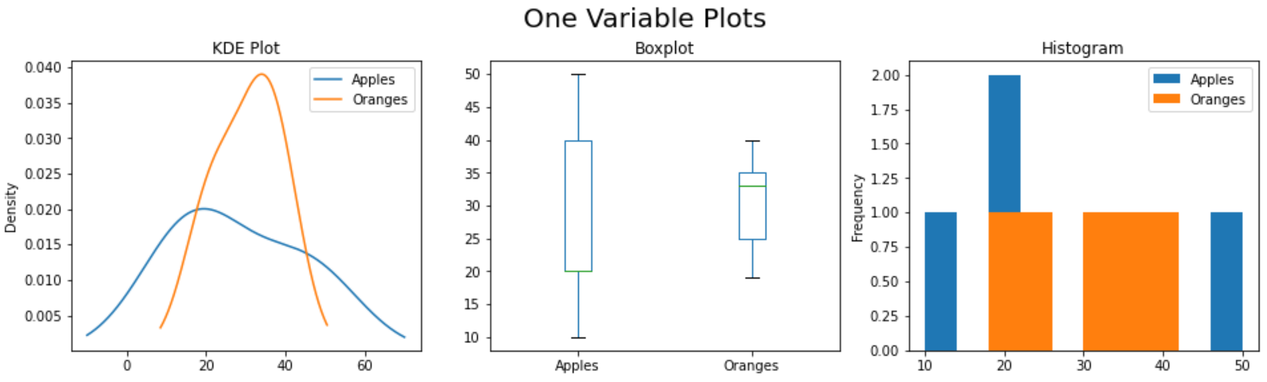
```
In [21]: fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize = (16, 4))
         fig.suptitle('Two Variable Plots', size = 20, y = 1.02)
         df.plot(kind = 'line', ax = ax1, title = 'Line Plot')
         df.plot(kind = 'bar', ax = ax2, title = 'Bar Plot')
         df.plot(x = 'Apples', y = 'Oranges', kind = 'scatter', ax = ax3, title = 'Scatterplot')
```

```
Out[21]: <AxesSubplot:title={'center':'Scatterplot'}, xlabel='Apples', ylabel='Oranges'>
```



```
In [22]: fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize = (16, 4))
         fig.suptitle('One Variable Plots', size = 20, y = 1.02)
         df.plot(kind = 'kde', ax = ax1, title = 'KDE Plot')
         df.plot(kind = 'box', ax = ax2, title = 'Boxplot')
         df.plot(kind = 'hist', ax = ax3, title = 'Histogram')
```

```
Out[22]: <AxesSubplot:title={'center':'Histogram'}, ylabel='Frequency'>
```

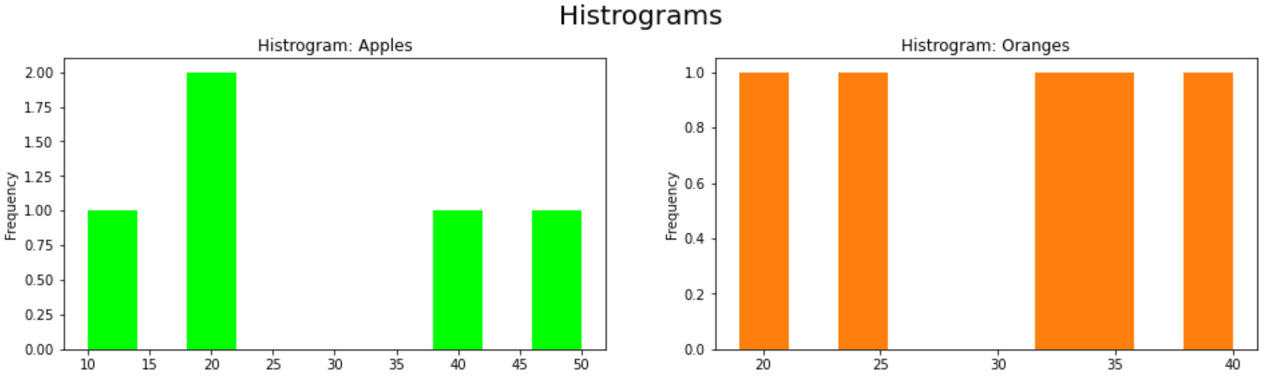


In [23]:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize = (16, 4))
fig.suptitle('Histrograms', size = 20, y = 1.02)
df.loc[:, 'Apples'].plot(kind = 'hist', ax = ax1, color = 'lime', title = 'Histrogram: 
df.loc[:, 'Oranges'].plot(kind = 'hist', ax = ax2, color = 'tab:orange', title = 'Histr
```

Out[23]:

```
<AxesSubplot:title={'center':'Histrogram: Oranges'}, ylabel='Frequency'>
```



(3) Visualizing the Flights Dataset

In [24]:

```
flights = pd.read_csv('flights.csv')
flights
```

Out[24]:

	MONTH	DAY	WEEKDAY	AIRLINE	ORG_AIR	DEST_AIR	SCHED_DEP	DEP_DELAY	AIR_TIME	DI
0	1	1	4	WN	LAX	SLC	1625	58.0	94.0	5
1	1	1	4	UA	DEN	IAD	823	7.0	154.0	14
2	1	1	4	MQ	DFW	VPS	1305	36.0	85.0	6
3	1	1	4	AA	DFW	DCA	1555	7.0	126.0	11
4	1	1	4	WN	LAX	MCI	1720	48.0	166.0	13
...	...	...	...	...	...	...	...	...	...	
58487	12	31	4	AA	SFO	DFW	515	5.0	166.0	14
58488	12	31	4	F9	LAS	SFO	1910	13.0	71.0	4
58489	12	31	4	OO	SFO	SBA	1846	-6.0	46.0	2
58490	12	31	4	WN	MSP	ATL	525	39.0	124.0	9
58491	12	31	4	OO	SFO	BOI	859	5.0	73.0	5

58492 rows × 14 columns

In [25]:

```
ac = flights['AIRLINE'].value_counts()
ac
```

Out[25]:

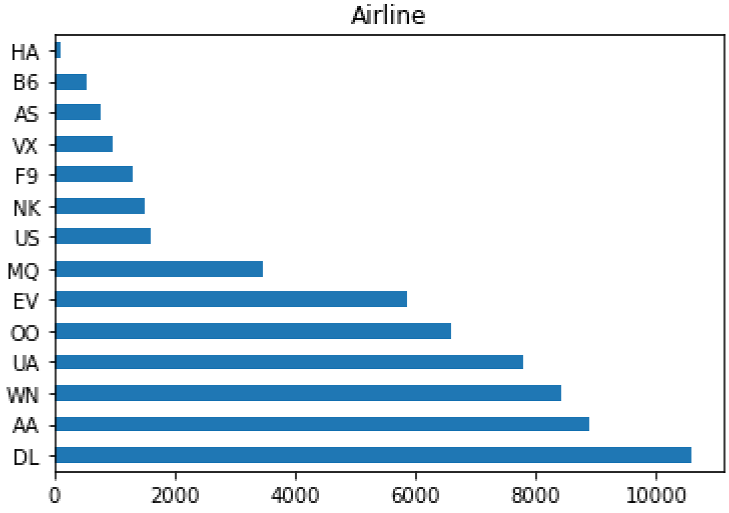
```
DL    10601
AA     8900
WN     8418
UA     7792
OO     6588
EV     5858
MQ     3471
US     1615
NK     1516
F9     1317
VX      993
AS      768
B6      543
HA      112
Name: AIRLINE, dtype: int64
```

In [26]:

```
ac.plot(kind = 'barh', title = 'Airline')
```

Out[26]:

```
<AxesSubplot:title={'center':'Airline'}>
```



In [27]:

```
oc = flights['ORG_AIR'].value_counts()
oc
```

Out[27]:

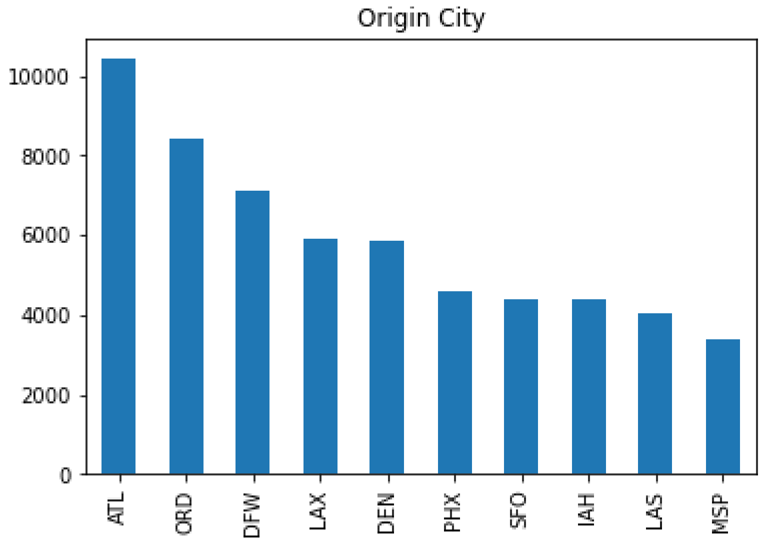
```
ATL    10413
ORD     8394
DFW     7121
LAX     5889
DEN     5857
PHX     4603
SFO     4402
IAH     4384
LAS     4019
MSP     3410
Name: ORG_AIR, dtype: int64
```

In [28]:

```
oc.plot(kind = 'bar', title = 'Origin City')
```

Out[28]:

```
<AxesSubplot:title={'center':'Origin City'}>
```

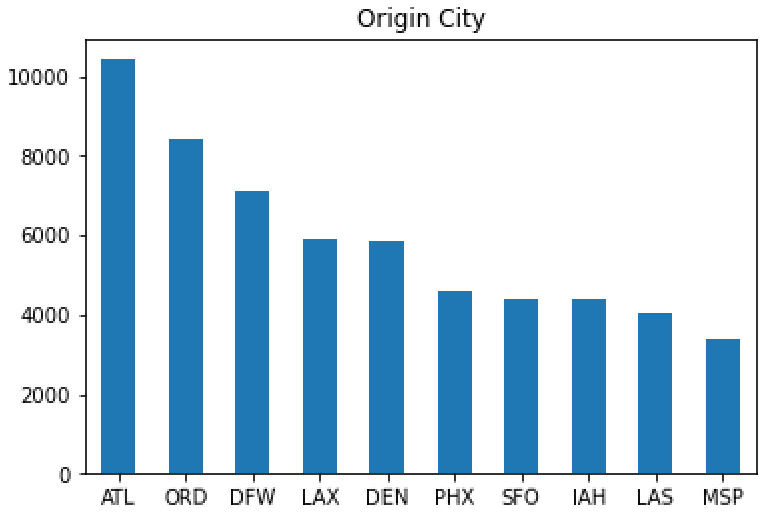


In [29]:

```
oc.plot(kind = 'bar', rot = True, title = 'Origin City')
```

Out[29]:

```
<AxesSubplot:title={'center':'Origin City'}>
```



In [30]:

```
dc = flights['DEST_AIR'].value_counts()
dc
```

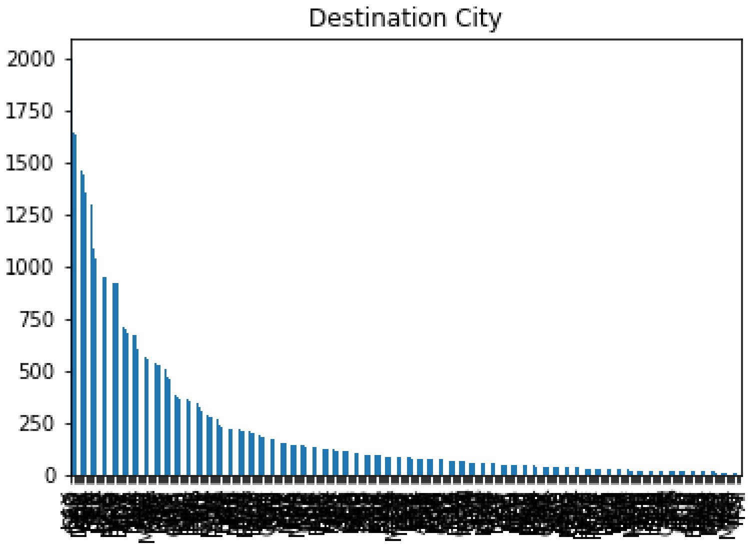
Out[30]:

```
LAX    1991
SFO    1637
ORD    1634
DEN    1581
DFW    1454
       ... 
MMH       4
ITO       2
CEC       2
IMT       2
FAI       1
Name: DEST_AIR, Length: 271, dtype: int64
```

In [31]:

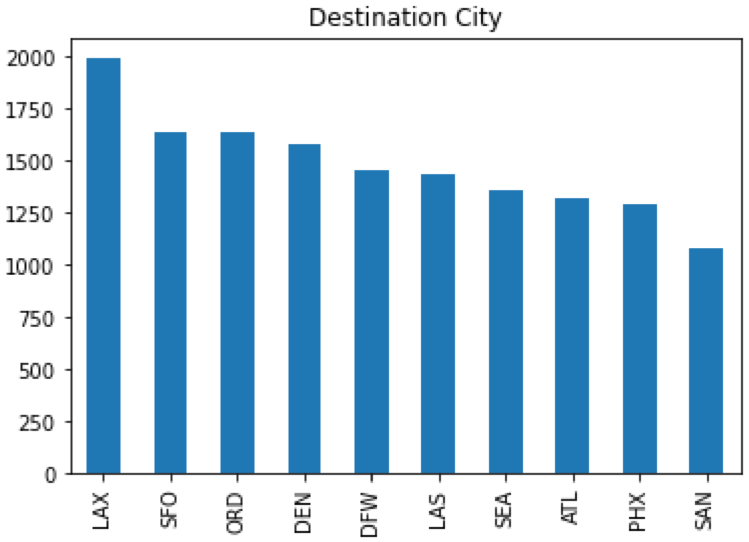
```
dc.plot(kind = 'bar', title = 'Destination City')
```

```
Out[31]: <AxesSubplot:title={'center':'Destination City'}>
```



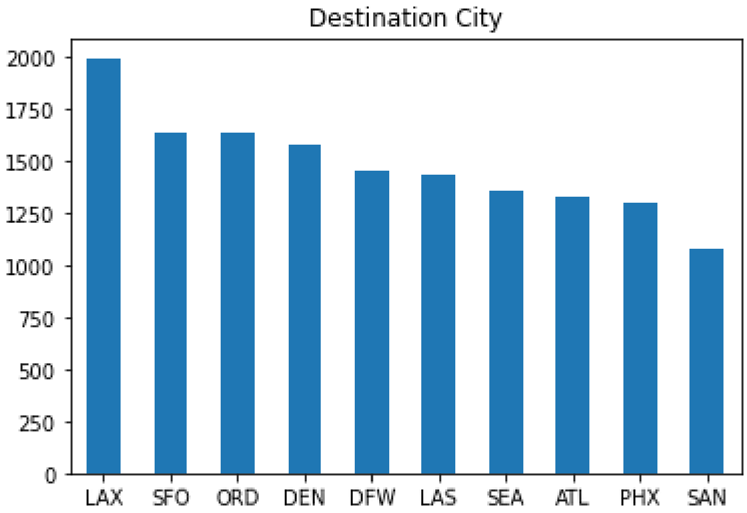
```
In [32]: dc = flights['DEST_AIR'].value_counts().head(10)
         dc.plot(kind = 'bar', title = 'Destination City')
```

```
Out[32]: <AxesSubplot:title={'center':'Destination City'}>
```



```
In [33]: dc.plot(kind = 'bar', rot = True, title = 'Destination City')
```

```
Out[33]: <AxesSubplot:title={'center':'Destination City'}>
```



In [34]:

```
flights
```

Out[34]:

	MONTH	DAY	WEEKDAY	AIRLINE	ORG_AIR	DEST_AIR	SCHED_DEP	DEP_DELAY	AIR_TIME	DI
0	1	1	4	WN	LAX	SLC	1625	58.0	94.0	5
1	1	1	4	UA	DEN	IAD	823	7.0	154.0	14
2	1	1	4	MQ	DFW	VPS	1305	36.0	85.0	6
3	1	1	4	AA	DFW	DCA	1555	7.0	126.0	11
4	1	1	4	WN	LAX	MCI	1720	48.0	166.0	13
...	...	...	...	...	...	...	...	...	...	
58487	12	31	4	AA	SFO	DFW	515	5.0	166.0	14
58488	12	31	4	F9	LAS	SFO	1910	13.0	71.0	4
58489	12	31	4	OO	SFO	SBA	1846	-6.0	46.0	2
58490	12	31	4	WN	MSP	ATL	525	39.0	124.0	9
58491	12	31	4	OO	SFO	BOI	859	5.0	73.0	5.

58492 rows × 14 columns

In [35]:

```
flights.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58492 entries, 0 to 58491
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   MONTH      58492 non-null  int64
 1   DAY        58492 non-null  int64
 2   WEEKDAY    58492 non-null  int64
 3   AIRLINE    58492 non-null  object
 4   ORG_AIR    58492 non-null  object
 5   DEST_AIR   58492 non-null  object
 6   SCHED_DEP  58492 non-null  int64
 7   DEP_DELAY  57659 non-null  float64
 8   AIR_TIME   57474 non-null  float64
 9   DIST       58492 non-null  int64
 10  SCHED_ARR  58492 non-null  int64
 11  ARR_DELAY  57474 non-null  float64
 12  DIVERTED   58492 non-null  int64
 13  CANCELLED  58492 non-null  int64
dtypes: float64(3), int64(8), object(3)
memory usage: 6.2+ MB
```

In [36]:

```
flights['DELAYED'] = flights['ARR_DELAY'].ge(15).astype('int64')
```

In [37]:

```
flights.columns
```

Out[37]:

```
Index(['MONTH', 'DAY', 'WEEKDAY', 'AIRLINE', 'ORG_AIR', 'DEST_AIR',
       'SCHED_DEP', 'DEP_DELAY', 'AIR_TIME', 'DIST', 'SCHED_ARR', 'ARR_DELAY',
       'DIVERTED', 'CANCELLED', 'DELAYED'],
      dtype='object')
```

In [38]:

```
flights[['DIVERTED', 'CANCELLED', 'DELAYED']]
```

Out[38]:

	DIVERTED	CANCELLED	DELAYED
0	0	0	1
1	0	0	0
2	0	0	1
3	0	0	0
4	0	0	1
...	...	...	...
58487	0	0	0
58488	0	0	0
58489	0	0	0
58490	0	0	1
58491	0	0	0

58492 rows × 3 columns

In [39]:

```
flights[['DIVERTED', 'CANCELLED', 'DELAYED']].any()
```

Out[39]:

```
DIVERTED     True
CANCELLED    True
DELAYED      True
dtype: bool
```

In [40]:

```
flights[['DIVERTED', 'CANCELLED', 'DELAYED']].any(axis = 'columns')
```

Out[40]:

```
0         True
1        False
```

```
2         True
3        False
4         True
         ...
58487    False
58488    False
58489    False
58490     True
58491    False
Length: 58492, dtype: bool
```

In [41]:

```
1 - flights[['DIVERTED', 'CANCELLED', 'DELAYED']].any(axis = 'columns')
```

Out[41]:

```
0        0
1        1
2        0
3        1
4        0
        ..
58487    1
58488    1
58489    1
58490    0
58491    1
Length: 58492, dtype: int32
```

In [42]:

```
flights['ON_TIME'] = 1 - flights[['DIVERTED', 'CANCELLED', 'DELAYED']].any(axis = 1)
```

In [43]:

```
flights
```

Out[43]:

	MONTH	DAY	WEEKDAY	AIRLINE	ORG_AIR	DEST_AIR	SCHED_DEP	DEP_DELAY	AIR_TIME	DI
0	1	1	4	WN	LAX	SLC	1625	58.0	94.0	5
1	1	1	4	UA	DEN	IAD	823	7.0	154.0	14
2	1	1	4	MQ	DFW	VPS	1305	36.0	85.0	6
3	1	1	4	AA	DFW	DCA	1555	7.0	126.0	11
4	1	1	4	WN	LAX	MCI	1720	48.0	166.0	13
...	...	...	...	...	...	...	...	...	...	
58487	12	31	4	AA	SFO	DFW	515	5.0	166.0	14
58488	12	31	4	F9	LAS	SFO	1910	13.0	71.0	4
58489	12	31	4	OO	SFO	SBA	1846	-6.0	46.0	2
58490	12	31	4	WN	MSP	ATL	525	39.0	124.0	9
58491	12	31	4	OO	SFO	BOI	859	5.0	73.0	5

58492 rows × 16 columns

In [44]:

```
flights[['DIVERTED', 'CANCELLED', 'DELAYED', 'ON_TIME']]
```

Out[44]:

	DIVERTED	CANCELLED	DELAYED	ON_TIME
0	0	0	1	0
1	0	0	0	1
2	0	0	1	0
3	0	0	0	1
4	0	0	1	0
...	...	...	...	...
58487	0	0	0	1
58488	0	0	0	1
58489	0	0	0	1
58490	0	0	1	0
58491	0	0	0	1

58492 rows × 4 columns

In [45]:

```
flights[['DIVERTED', 'CANCELLED', 'DELAYED', 'ON_TIME']].sum()
```

Out[45]:

```
DIVERTED       137
CANCELLED      881
DELAYED      11685
ON_TIME      45789
dtype: int64
```

In [46]:

```
status = flights[['DIVERTED', 'CANCELLED', 'DELAYED', 'ON_TIME']].sum()
status
```

Out[46]:

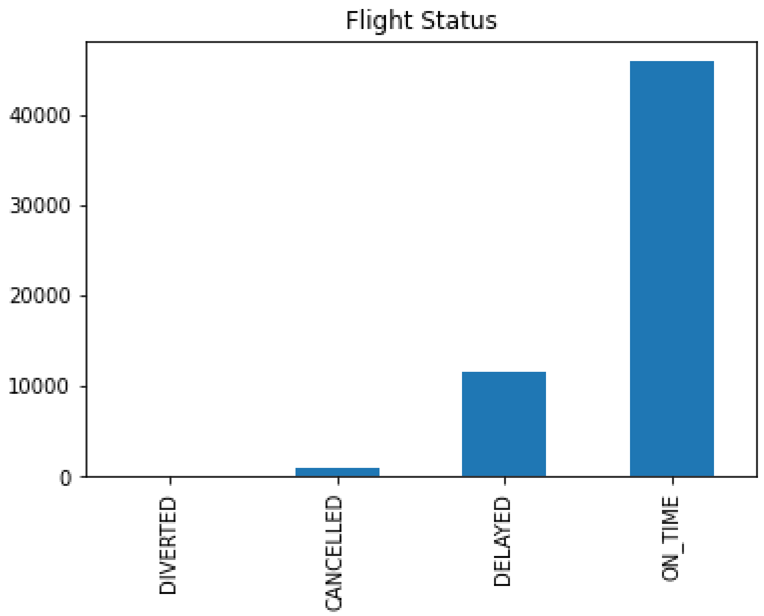
```
DIVERTED       137
CANCELLED      881
DELAYED      11685
ON_TIME      45789
dtype: int64
```

In [47]:

```
status.plot(kind = 'bar', title = 'Flight Status')
```

Out[47]:

```
<AxesSubplot:title={'center':'Flight Status'}>
```

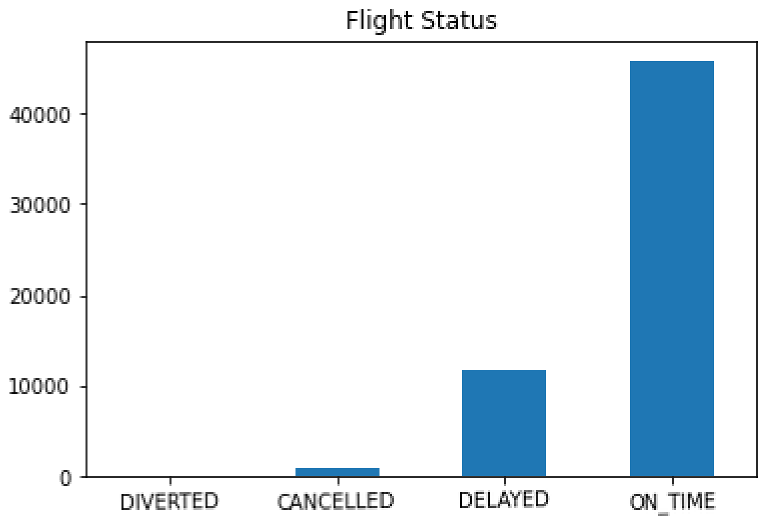


In [48]:

```
status.plot(kind = 'bar', rot = True, title = 'Flight Status')
```

Out[48]:

```
<AxesSubplot:title={'center':'Flight Status'}>
```



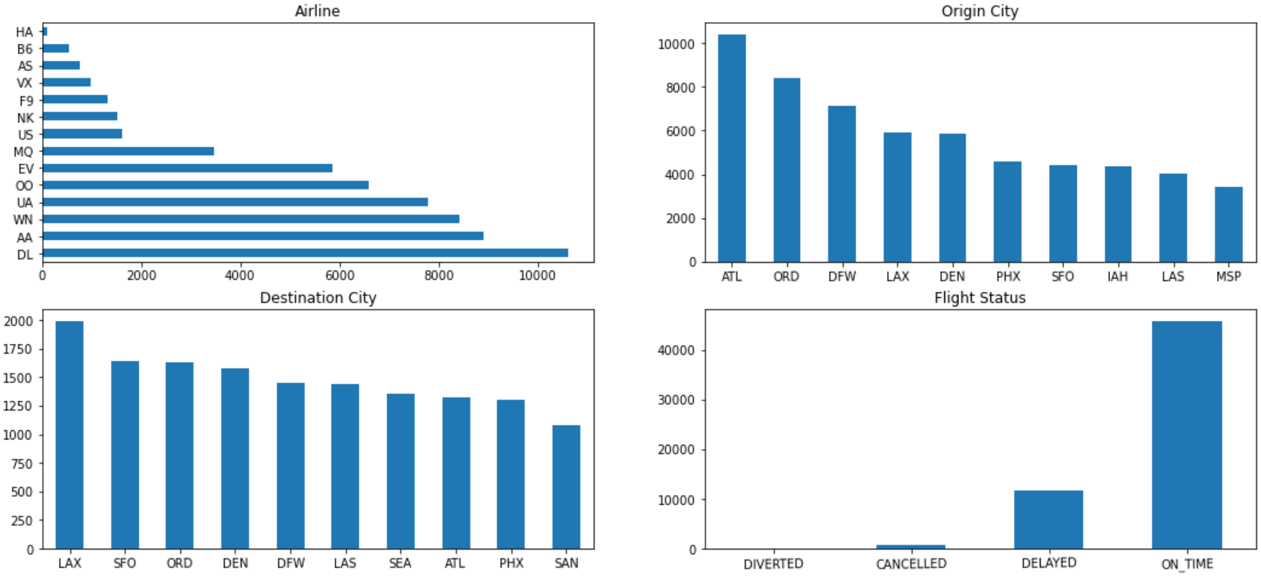
In [49]:

```
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize = (18, 8))

fig.suptitle('US Flights: Univariate Summary', size = 20)

ac = flights['AIRLINE'].value_counts()
ac.plot(kind = 'barh', ax = ax1, rot = True, title = 'Airline')

oc = flights['ORG_AIR'].value_counts()
oc.plot(kind = 'bar', ax = ax2, rot = True, title = 'Origin City')

dc = flights['DEST_AIR'].value_counts().head(10)
dc.plot(kind = 'bar', ax = ax3, rot = True, title = 'Destination City')

status.plot(kind = 'bar', ax = ax4, rot = True, title = 'Flight Status')
```

Out[49]:

```
<AxesSubplot:title={'center':'Flight Status'}>
```

US Flights: Univariate Summary



Airline

Origin City

Destination City

Flight Status