Link Github: https://github.com/paulbboone/DataMining_ThucHanh

HOMEWORK LAB 05

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

questions_per_library = pd.read_csv('stackoverflow.zip', parse_dates = True, ind
                                    ).loc[:,'pandas':'bokeh'].resample('1M').sum()
                                      pd.date_range('2008-08', '2021-10', freq = '
                                      ).fillna(0)
questions_per_library.tail()
```

Out[2]:

	pandas	matplotlib	numpy	seaborn	geopandas	geoviews	altair	yellowbrick	vega	hol
2021-05-31	200734.0	57853.0	89812.0	6855.0	1456.0	57.0	716.0	46.0	532.0	
2021-06-30	205065.0	58602.0	91026.0	7021.0	1522.0	57.0	760.0	48.0	557.0	
2021-07-31	209235.0	59428.0	92254.0	7174.0	1579.0	62.0	781.0	50.0	572.0	
2021-08-31	213410.0	60250.0	93349.0	7344.0	1631.0	62.0	797.0	52.0	589.0	
2021-09-30	214919.0	60554.0	93797.0	7414.0	1652.0	63.0	804.0	54.0	598.0	

In [3]:
```
from matplotlib.animation import FuncAnimation
```

In [4]:
```
import matplotlib.pyplot as plt
from matplotlib import ticker

def bar_plot(data):
    fig, ax  =plt.subplots(figsize=(8, 6))
    sort_order = data.last('1M').squeeze().sort_values().index
    bars= [
        bar.set_label(label) for label, bar in
        zip(sort_order, ax.barh(sort_order, [0] * data.shape[1]))
    ]
    ax.set_xlabel('total questions', fontweight='bold')
    ax.set_xlim(0, 250_000)
    ax.xaxis.set_major_formatter (ticker. EngFormatter())
    ax.xaxis.set_tick_params (labelsize=12)
    ax.yaxis.set_tick_params (labelsize=12)

    for spine in ['top','right']:
            ax.spines[spine].set_visible(False)
            fig.tight_layout()
            return fig, ax
```

In [5]:
```
%config InlineBackend.figure_formats = ['svg']
%matplotlib inline
bar_plot(questions_per_library)
```

Out[5]:
```
(<Figure size 800x600 with 1 Axes>, <AxesSubplot:xlabel='total questions'>)

<Figure size 800x600 with 1 Axes>
```

In [6]:
```
def generate_plot_text(ax):
    annotations = [
        ax.annotate(
            '', xy=(0, bar.get_y() + bar.get_height()/2),
            ha = 'left', va = 'center'
        ) for bar in ax.patches
    ]

    time_text = ax.text(
        0.9, 0.1, '', transform = ax.transAxes,
        fontsize = 15, ha = 'center', va = 'center'
    )
    return annotations, time_text
```

In [7]:
```
def update (frame, * ax, df, annotations, time_text):
    data = df.loc[frame, :]
    #update bars
    for rect, text in zip(ax.patches, annotations):
        col = rect.get_label()
        if data[col]:
            rect.set_width (data[col])
            text.set_x(data[col])
            text.set_text(f' (data[col]:,.0f)')
        #update time
        time_text.set_text(frame.strftime('%b\n%Y'))
```

In [8]:
```
from functools import partial
def bar_plot_init(questions_per_library):
    fig, ax = bar_plot(questions_per_library)
    annotations, time_text = generate_plot_text(ax)

    bar_plot_update = partial (
        update, ax=ax, df =questions_per_library,
        annotations= annotations, time_text=time_text
    )
    return fig, bar_plot_update
```

In [9]:
```
pip install Pillow
```

```
Requirement already satisfied: Pillow in c:\anaconda3\lib\site-packages (9.2.0)
Note: you may need to restart the kernel to use updated packages.
```

In [10]:
```
pip install ffmpeg-python
```

```
Collecting ffmpeg-python
  Downloading ffmpeg_python-0.2.0-py3-none-any.whl (25 kB)
Requirement already satisfied: future in c:\anaconda3\lib\site-packages (from f
fmpeg-python) (0.18.2)
Installing collected packages: ffmpeg-python
Successfully installed ffmpeg-python-0.2.0
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
import matplotlib.animation as animation
from IPython.display import HTML
```

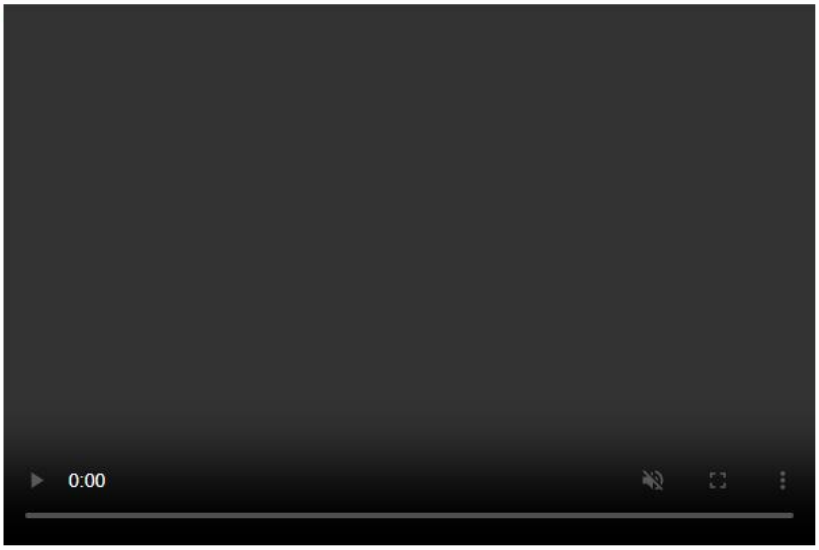
In [12]:
```
fig, update_func=bar_plot_init(questions_per_library)

ani=FuncAnimation(
    fig, update_func, frames=questions_per_library.index, repeat=False
)
ani.save(
    'stackoverflow_questions.gif',
    writer='ffmpeg',fps=30,bitrate=100,dpi=300
)
plt.close()
```

```
MovieWriter ffmpeg unavailable; using Pillow instead.
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\anaconda3\lib\site-packages\matplotlib\animation.py in saving(self, fig, outfile, dpi, *args, **kw
args)
    233         try:
--> 234             yield self
    235         finally:

C:\anaconda3\lib\site-packages\matplotlib\animation.py in save(self, filename, writer, fps, dpi, code
c, bitrate, extra_args, metadata, extra_anim, savefig_kwargs, progress_callback)
   1075             for anim in all_anim:
-> 1076                 anim._init_draw()  # Clear the initial frame
   1077             frame_number = 0

C:\anaconda3\lib\site-packages\matplotlib\animation.py in _init_draw(self)
   1695                 return
-> 1696             self._draw_frame(frame_data)
   1697         else:

C:\anaconda3\lib\site-packages\matplotlib\animation.py in _draw_frame(self, framedata)
   1717         # func needs to return a sequence of any artists that were modified.
-> 1718         self._drawn_artists = self._func(framedata, *self._args)
   1719

TypeError: update() got an unexpected keyword argument 'ax'

During handling of the above exception, another exception occurred:

~\AppData\Local\Temp\ipykernel_25860\3091944754.py in <module>
      4     fig, update_func, frames=questions_per_library.index, repeat=False
      5 )
----> 6 ani.save(
      7     'stackoverflow_questions.gif',
      8     writer='ffmpeg',fps=30,bitrate=100,dpi=300

C:\anaconda3\lib\site-packages\matplotlib\animation.py in save(self, filename, writer, fps, dpi, code
c, bitrate, extra_args, metadata, extra_anim, savefig_kwargs, progress_callback)
   1091                         progress_callback(frame_number, total_frames)
   1092                         frame_number += 1
-> 1093                 writer.grab_frame(**savefig_kwargs)
   1094
   1095     def _step(self, *args):

C:\anaconda3\lib\contextlib.py in __exit__(self, typ, value, traceback)
    135                 value = typ()
    136             try:
--> 137                 self.gen.throw(typ, value, traceback)
    138             except StopIteration as exc:
    139                 # Suppress StopIteration *unless* it's the same exception that

C:\anaconda3\lib\site-packages\matplotlib\animation.py in saving(self, fig, outfile, dpi, *args, **kw
args)
    234             yield self
    235         finally:
--> 236             self.finish()
    237
    238

C:\anaconda3\lib\site-packages\matplotlib\animation.py in finish(self)
    510
    511     def finish(self):
--> 512         self._frames[0].save(
    513             self.outfile, save_all=True, append_images=self._frames[1:],
    514             duration=int(1000 / self.fps), loop=0)

IndexError: list index out of range
```

In [13]:
```
from IPython import display

display.Video(
    'stackoverflow_questions.mp4', width=600, height=400,
    embed=True, html_attributes = 'controls muted autoplay'
)
```

Out[13]:



In [15]:
```
subway = pd.read_csv(
    'NYC_subway_daily.csv', parse_dates=['Datetime'],
    index_col=[ 'Borough', 'Datetime']
)
subway_daily=subway.unstack (0)
subway_daily.head()
```

Out[15]:

	Entries				Exits			
Borough	Bk	Bx	M	Q	Bk	Bx	M	Q
Datetime								
2017-02-04	617650.0	247539.0	1390496.0	408736.0	417449.0	148237.0	1225689.0	279699.0
2017-02-05	542667.0	199078.0	1232537.0	339716.0	405607.0	139856.0	1033610.0	268626.0
2017-02-06	1184916.0	472846.0	2774016.0	787206.0	761166.0	267991.0	2240027.0	537780.0
2017-02-07	1192638.0	470573.0	2892462.0	790557.0	763653.0	270007.0	2325024.0	544828.0
2017-02-08	1243658.0	497412.0	2998897.0	825679.0	788356.0	275695.0	2389534.0	559639.0

In [16]:
```
manhattan_entries = subway_daily['Entries']['M']
```

In [17]:
```
import numpy as np

count_per_bin, bin_ranges = np.histogram(manhattan_entries, bins =30)
```

In [18]:
```
def subway_histogram(data, bins, date_range):
    _, bin_ranges = np.histogram(data, bins=bins)
    weekday_mask= data. index.weekday < 5
    configs = [
        {'label': 'Weekend', 'mask': ~weekday_mask, 'ymax': 60},
        {'label' : 'Weekday', 'mask': weekday_mask, 'ymax': 120}
    ]

    fig, axes = plt.subplots(1, 2, figsize=(8, 4), sharex=True)
    for ax, config in zip (axes, configs):
        _, _, config['hist'] =ax.hist (
            data[config['mask']].loc[date_range], bin_ranges, ec='black'
        )
        ax.xaxis.set_major_formatter (ticker. EngFormatter())
        ax.set(
            xlim =(0, None), ylim=(0, config['ymax']),
             xlabel=f'{config["label"]} Entries'
        )
        for spine in ['top', 'right']:
            ax.spines [spine].set_visible (False)



        axes[0].set_ylabel('Frequency')
        fig.suptitle('Histogram of Daily Subway Entries in Manhattan ')
        fig.tight_layout()
        return fig, axes, bin_ranges, configs
```

In [19]:
```
_ = subway_histogram(manhattan_entries, bins=30, date_range = '2017')
```

```
<Figure size 800x400 with 2 Axes>
```

In [20]:
```
def add_time_text(ax):
    time_text = ax.text(
        0.15, 0.9, '', transform=ax.transAxes,
        fontsize=15, ha='center', va='center'
    )
    return time_text
```

[21]:
```
def update(frame, *, data, configs, time_next, bin_ranges):
    artists = []
    time = frame.strftime('%b\n%Y')
    if time != time_text.get_text():
        time_next.set_text(time)
        artist.append(time_next)

    for config in configs:
        time_frame_mask = \
            (data.index > frame - pd.Timedelta(days=365)) & (data.index <= frame)
        counts, _ = np.histogram(
            data[time_frame_mask & config['mask']],
            bin_ranges
        )
        for count, rect in zip(counts, config['hist'].patches):
            if count != rect.get_height():
                rect.set_height(count)
                artists.append(rect)

    return artists
```

[22]:
```
def histogram_init(data, bins, initial_data_range):
    fig, axes, bin_ranges, configs = subway_histogram(data, bins, initial_date_range)

    update_func = partial(
        update, data=data, configs=configs,
        time_next=add_time_text(axes[0]),
        bin_ranges=bin_ranges
    )

    return fig, update_func
```

[23]:
```
fig, update_func = histogram_init (
    manhattan_entries, bins=30, initial_date_range=slice('2017', '2019-07')
)

ani = FuncAnimation(
    fig, update_func, frames=manhattan_entries['2019-08':'2021'].index,
    repeat=False, blit=True
)
ani.save(
    'subway_entries_subplots.gif',
    writer='ffmpeg', fps=30, bitrate=500, dpi=300
)

plt.close()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25860\4035103259.py in <module>
----> 1 fig, update_func = histogram_init (
      2     manhattan_entries, bins=30, initial_date_range=slice('2017', '2019-07')
      3 )
      4
      5 ani = FuncAnimation(

TypeError: histogram_init() got an unexpected keyword argument 'initial_date_range'
```

[28]:
```
pip install geopandas
```

```
Requirement already satisfied: geopandas in c:\anaconda3\lib\site-packages (0.13.0)
Requirement already satisfied: pandas>=1.1.0 in c:\anaconda3\lib\site-packages (from geopanc
4)
Requirement already satisfied: shapely>=1.7.1 in c:\anaconda3\lib\site-packages (from geopar
0.1)
Requirement already satisfied: packaging in c:\anaconda3\lib\site-packages (from geopandas)
Requirement already satisfied: fiona>=1.8.19 in c:\anaconda3\lib\site-packages (from geopanc
4.post1)
Requirement already satisfied: pyproj>=3.0.1 in c:\anaconda3\lib\site-packages (from geopanc
0)
Requirement already satisfied: click-plugins>=1.0 in c:\anaconda3\lib\site-packages (from fi
19->geopandas) (1.1.1)
Requirement already satisfied: certifi in c:\anaconda3\lib\site-packages (from fiona>=1.8.19
as) (2022.9.14)
Requirement already satisfied: importlib-metadata in c:\anaconda3\lib\site-packages (from fi
19->geopandas) (4.11.3)
Requirement already satisfied: attrs>=19.2.0 in c:\anaconda3\lib\site-packages (from fiona>=
eopandas) (21.4.0)
Requirement already satisfied: click~=8.0 in c:\anaconda3\lib\site-packages (from fiona>=1.8
andas) (8.0.4)
Requirement already satisfied: cligj>=0.5 in c:\anaconda3\lib\site-packages (from fiona>=1.8
andas) (0.7.2)
Requirement already satisfied: six in c:\anaconda3\lib\site-packages (from fiona>=1.8.19->ge
(1.16.0)
Requirement already satisfied: pytz>=2020.1 in c:\anaconda3\lib\site-packages (from pandas>=
opandas) (2023.3)
Requirement already satisfied: numpy>=1.18.5 in c:\anaconda3\lib\site-packages (from pandas>
eopandas) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\anaconda3\lib\site-packages (fro
=1.1.0->geopandas) (2.8.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\anaconda3\lib\site-packages (t
ging->geopandas) (3.0.9)
Requirement already satisfied: colorama in c:\anaconda3\lib\site-packages (from click~=8.0->
8.19->geopandas) (0.4.5)
Requirement already satisfied: zipp>=0.5 in c:\anaconda3\lib\site-packages (from importlib-m
fiona>=1.8.19->geopandas) (3.8.0)
Note: you may need to restart the kernel to use updated packages.
```

[30]:
```
import geopandas as gpd

earthquakes = gpd.read_file('earthquakes.geojson').assign(
    time=lambda x: pd.to_datetime(x.time, unit='ms'),
    month=lambda x: x.time.dt.month
)[['geometry', 'mag', 'time', 'month']]

earthquakes.shape
```

[30]:
```
(188527, 4)
```

[31]:
```
earthquakes.head()
```

[31]:

	geometry	mag	time	month
0	POINT Z (-67.12750 19.21750 12.00000)	2.75	2020-01-01 00:01:56.590	1
1	POINT Z (-67.09010 19.07660 6.00000)	2.55	2020-01-01 00:03:38.210	1
2	POINT Z (-66.85410 17.87050 6.00000)	1.81	2020-01-01 00:05:09.440	1
3	POINT Z (-66.86360 17.89930 8.00000)	1.84	2020-01-01 00:05:36.930	1
4	POINT Z (-66.86850 17.90660 8.00000)	1.64	2020-01-01 00:09:20.060	1

[37]:
```
pip install geoviews
```

```
Collecting geoviews
  Using cached geoviews-1.10.0-py2.py3-none-any.whl (504 kB)
Requirement already satisfied: packaging in c:\anaconda3\lib\site-packages (from geo
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages (from geoviev
Requirement already satisfied: shapely in c:\anaconda3\lib\site-packages (from geovi
Collecting holoviews>=1.16.0
  Using cached holoviews-1.16.0-py2.py3-none-any.whl (4.3 MB)
Collecting panel>=1.0.0
  Using cached panel-1.0.3-py2.py3-none-any.whl (19.9 MB)
Collecting bokeh<3.2.0,>=3.1.0
  Using cached bokeh-3.1.1-py3-none-any.whl (8.3 MB)
Requirement already satisfied: param in c:\anaconda3\lib\site-packages (from geoviev
Requirement already satisfied: pyproj in c:\anaconda3\lib\site-packages (from geovie
Requirement already satisfied: cartopy>=0.18.0 in c:\anaconda3\lib\site-packages (fr
(0.21.1)
Collecting contourpy>=1
  Using cached contourpy-1.0.7-cp39-cp39-win_amd64.whl (160 kB)
Collecting xyzservices>=2021.09.1
  Using cached xyzservices-2023.5.0-py3-none-any.whl (56 kB)
```

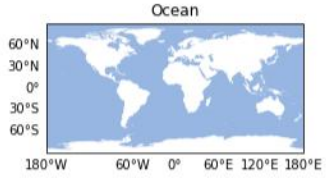
[33]:
```
!pip install shapely --no-binary shapely
```

```
Requirement already satisfied: shapely in c:\anaconda3\lib\site-packages (2.0.1)
Requirement already satisfied: numpy>=1.14 in c:\anaconda3\lib\site-packages (from shapely) (1.21.5)
```

[38]:
```
import geoviews as gv
import geoviews.feature as gf
import holoviews as hv
gv.extension('matplotlib')
gf.ocean
```

```
C:\anaconda3\lib\site-packages\cartopy\io\__init__.py:241: DownloadWarning: Downloading: https://natu
ralearth.s3.amazonaws.com/110m_physical/ne_110m_ocean.zip
  warnings.warn(f'Downloading: {url}', DownloadWarning)
```

[38]:



[4]:
```
import geopandas as gpd
import pandas as pd

earthquakes = gpd.read_file('earthquakes.geojson').assign(
    time = lambda x: pd.to_datetime(x.time, unit='ms'),
    month = lambda x: x.time.dt.month
).dropna()

earthquakes.head()
```

t[4]:

	mag	place	time	tsunami	magType	geometry	month
0	2.75	80 km N of Isabela, Puerto Rico	2020-01-01 00:01:56.590	0	md	POINT Z (-67.12750 19.21750 12.00000)	1
1	2.55	64 km N of Isabela, Puerto Rico	2020-01-01 00:03:38.210	0	md	POINT Z (-67.09010 19.07660 6.00000)	1
2	1.81	12 km SSE of Maria Antonia, Puerto Rico	2020-01-01 00:05:09.440	0	md	POINT Z (-66.85410 17.87050 6.00000)	1
3	1.84	9 km SSE of Maria Antonia, Puerto Rico	2020-01-01 00:05:36.930	0	md	POINT Z (-66.86360 17.89930 8.00000)	1
4	1.64	8 km SSE of Maria Antonia, Puerto Rico	2020-01-01 00:09:20.060	0	md	POINT Z (-66.86850 17.90660 8.00000)	1

[5]:
```
january_earthquakes = earthquakes.query('month == 1').assign(
    longitude=lambda x: x.geometry.x,
    latitude=lambda x: x.geometry.y
).drop(columns = ['month', 'geometry'])
```

n [6]:
```
import hvplot.pandas
```

n [7]:
```
geo = january_earthquakes.hvplot(
    x = 'longitude', y='latitude', kind = 'points',
    color = 'mag', cmap='fire_r', clim = (-2, 10),
    tiles = 'CartoLight', geo = True, global_extent = True,
    xlabel = 'Longitude', ylabel = 'Latitude', title = 'January 2020 Earthquakes',
    frame_height=450
)
```

n [9]:
```
import numpy as np
flights_stats = pd.read_csv(
    'T100_MARKET_ALL_CARRIER.zip',
    usecols=[
        'CLASS', 'REGION', 'UNIQUE_CARRIER_NAME', 'ORIGIN_CITY_NAME', 'ORIGIN',
        'DEST_CITY_NAME', 'DEST', 'PASSENGERS', 'FREIGHT', 'MAIL'
    ]
).rename(lambda x: x.lower(), axis=1).assign(
    region=lambda x: x.region.replace({
        'D': 'Domestic', 'I': 'International', 'A': 'Atlantic',
        'L': 'Latin America', 'P': 'Pacific', 'S': 'System'
    }),
    route = lambda x: np.where(
        x.origin < x.dest,
        x.origin + '-' + x.dest,
        x.dest + '-' + x.origin
    )
)
```

n [10]:
```
flights_stats.head()
```

ut[10]:

	passengers	freight	mail	unique_carrier_name	region	origin	origin_city_name	dest	dest_city_name	class	route
0	0.0	53185.0	0.0	Emirates	International	DXB	Dubai, United Arab Emirates	IAH	Houston, TX	G	DXB-IAH
1	0.0	9002.0	0.0	Emirates	International	DXB	Dubai, United Arab Emirates	JFK	New York, NY	G	DXB-JFK
2	0.0	2220750.0	0.0	Emirates	International	DXB	Dubai, United Arab Emirates	ORD	Chicago, IL	G	DXB-ORD
3	0.0	1201490.0	0.0	Emirates	International	IAH	Houston, TX	DXB	Dubai, United Arab Emirates	G	DXB-IAH
4	0.0	248642.0	0.0	Emirates	International	JFK	New York, NY	DXB	Dubai, United Arab Emirates	G	DXB-JFK

n [11]:
```
cities = [
    'Atlanta, GA', 'Chicago, IL', 'New York, NY', 'Los Angeles, CA',
    'Dallas/Fort Worth, TX', 'Denver, CO', 'Houston, TX',
    'San Francisco, CA', 'Seattle, WA', 'Orlando, FL'
]

top_airlines = [
    'American airlines Inc.', 'Delta Air Lines Inc.', 'JetBlue Airways',
    'Southwest Airlines Co.', 'United Air Lines Inc.'
]
```

n [12]:
```
total_flight_stats = flights_stats.query(
    f'`class` == "F" and origin_city_name != dest_city_name'
    f' and origin_city_name.isin({cities}) and dest_city_name.isin({cities})'
).groupby([
    'origin', 'origin_city_name', 'dest', 'dest_city_name'
]) [['passengers', 'freight', 'mail']].sum().reset_index().query('passengers > 0')
```

n [13]:
```
total_flight_stats.sample(10, random_state=1)
```

ut[13]:

	origin	origin_city_name	dest	dest_city_name	passengers	freight	mail
78	LGA	New York, NY	DEN	Denver, CO	589190.0	506023.0	293108.0
117	ORD	Chicago, IL	SEA	Seattle, WA	810594.0	1063463.0	2627325.0
31	DFW	Dallas/Fort Worth, TX	MCO	Orlando, FL	683700.0	187672.0	95570.0
5	ATL	Atlanta, GA	LAX	Los Angeles, CA	1121378.0	8707125.0	3267077.0
126	SEA	Seattle, WA	LGA	New York, NY	24.0	0.0	0.0
45	IAH	Houston, TX	ATL	Atlanta, GA	566369.0	367543.0	726670.0
14	DEN	Denver, CO	HOU	Houston, TX	305193.0	363119.0	0.0
44	HOU	Houston, TX	SFO	San Francisco, CA	1843.0	5523.0	0.0
73	LAX	Los Angeles, CA	MDW	Chicago, IL	277226.0	2022416.0	0.0
89	MCO	Orlando, FL	DEN	Denver, CO	594878.0	368516.0	138811.0

[14]:
```
import holoviews as hv

chord = hv.Chord(
    total_flight_stats,
    kdims = ['origin', 'dest'],
    vdims = ['passengers', 'origin_city_name', 'dest_city_name', 'mail', 'freight']
)
```

[15]:
```
from bokeh.models import HoverTool

tooltips = {
    'Source': '@origin_city_name (@origin)',
    'Target': '@dest_city_name (@dest)',
    'Passengers': '@passengers{0,.}',
    'Mail': '@mail{0,.} lbs.',
    'Freight': '@freight{0,.} lbs.',
}
hover = HoverTool(tooltips=tooltips)
```

[16]:
```
top_cities = cities[:5]

domestic_passenger_travel = flights_stats.query(
    'region == "Domestic" and `class` == "F" and origin_city_name != dest_city_name '
    f'and origin_city_name.isin({top_cities}) and dest_city_name.isin({top_cities})'
).groupby([
    'region', 'unique_carrier_name', 'route',
    'origin_city_name', 'dest_city_name'
]).passengers.sum().reset_index()

domestic_passenger_travel.head()
```

t[16]:

	region	unique_carrier_name	route	origin_city_name	dest_city_name	passengers
0	Domestic	Air Wisconsin Airlines Corp	ATL-ORD	Atlanta, GA	Chicago, IL	915.0
1	Domestic	Air Wisconsin Airlines Corp	ATL-ORD	Chicago, IL	Atlanta, GA	556.0
2	Domestic	Alaska Airlines Inc.	JFK-LAX	Los Angeles, CA	New York, NY	265307.0
3	Domestic	Alaska Airlines Inc.	JFK-LAX	New York, NY	Los Angeles, CA	257685.0
4	Domestic	Alaska Airlines Inc.	LAX-ORD	Chicago, IL	Los Angeles, CA	48269.0

[17]:
```
domestic_passenger_travel.unique_carrier_name.replace(
    '^(?!' + '|'.join(top_airlines) + ').*$',
    'Other Airlines',
    regex = True, inplace=True
)
```

[18]:
```
domestic_passenger_travel.groupby('unique_carrier_name').passengers.sum().div(
    domestic_passenger_travel.passengers.sum()
)
```

t[18]:
```
unique_carrier_name
Delta Air Lines Inc.      0.312187
JetBlue Airways           0.049500
Other Airlines            0.457730
Southwest Airlines Co.    0.079074
United Air Lines Inc.     0.101509
Name: passengers, dtype: float64
```

[19]:
```
def get_edges(data, *, source_col, target_col):
    aggregated = data.groupby([source_col, target_col]).passengers.sum()
    return aggregated.reset_index().rename(
        columns={source_col: 'source', target_col: 'target'}
    ).query('passengers > 0')
```

[20]:
```
carrier_edges = get_edges(
    domestic_passenger_travel,
    source_col='region',
    target_col = 'unique_carrier_name'
).replace('Domestic', 'Top Routes')

carrier_edges
```

t[20]:

	source	target	passengers
0	Top Routes	Delta Air Lines Inc.	8727210.0
1	Top Routes	JetBlue Airways	1383776.0
2	Top Routes	Other Airlines	12795875.0
3	Top Routes	Southwest Airlines Co.	2210533.0
4	Top Routes	United Air Lines Inc.	2837682.0

[21]:
```
carrier_to_route_edges = get_edges(
    domestic_passenger_travel,
    source_col='unique_carrier_name',
    target_col = 'route'
)

carrier_to_route_edges.sample(10, random_state=1)
```

t[21]:

	source	target	passengers
24	Other Airlines	DFW-JFK	373159.0
22	Other Airlines	ATL-MDW	1201.0
39	United Air Lines Inc.	ATL-DFW	4.0
35	Southwest Airlines Co.	ATL-MDW	498481.0
2	Delta Air Lines Inc.	ATL-LAX	1539875.0
3	Delta Air Lines Inc.	ATL-LGA	1649627.0
29	Other Airlines	JFK-ORD	382713.0
32	Other Airlines	LGA-ORD	1593364.0
47	United Air Lines Inc.	LAX-ORD	1297377.0
26	Other Airlines	DFW-LGA	1372795.0

[22]:
```
all_edges = pd.concat([carrier_edges, carrier_to_route_edges]).assign(
    passengers = lambda x: x.passengers / 1e6
)
```

[23]:
```
sankey = hv.Sankey(
    all_edges,
    kdims = ['source', 'target'],
    vdims = hv.Dimension('passengers', unit = 'M')
).opts(
    labels='index', label_position = 'right', cmap='Set1',
    edge_color= 'lightgray',
    width = 750, height = 600,
    title='Travele Between the Top 5 Cities in 2019'
)
```

[25]:
```
sankey
```

t[25]: