```
'''
Instruction: 6CCS3PRJ
             Data Visualisation of Migration Data

Code Author: Yanpu Huang,
             1725298,
             K1763861,
             Kings College London,
             Computer Science,
             3rd year student

supervisor: Prof.Dr.Rita Borgo

User guide: Install python 2.7
            Install bokeh(set the path to python 2.7 site-packages)
            Install pandas(set the path to python 2.7 site-packages)
            Install numpy(set the path to python 2.7 site-packages)
            Install openpyxl(set the path to python 2.7 site-packages)
            open terminal, run: python -m bokeh serve --show web.py, wait for the respond from browser
'''

import openpyxl
import numpy as np
import pandas as pd
from bokeh import events
from bokeh.plotting import figure, curdoc, output_file, show
from bokeh.models import ColumnDataSource, HoverTool, CustomJS
from bokeh.layouts import row, column, gridplot, widgetbox, layout
from bokeh.models.widgets import Button, RadioButtonGroup, Select, Slider, Dropdown, Toggle,Tabs, Panel, CheckboxGroup
from bokeh.transform import linear_cmap, factor_cmap, dodge
from bokeh.io import export_png
import warnings
from bokeh.core.properties import value

output_file("LondonDataStoreDataVisualisation.html",title="Migration Data Visualisation") #output file

# Make line chart of long term migration(London vs UK) by years
def plot_long_term_migration():
    wb = openpyxl.load_workbook('data/Long term international migration.xlsx') # Import datasets
    ws = wb['Data']

    x=[]
    y1=[]
    y2=[]
    y3=[]
    y4=[]
    for row in range(32,72,4):
        x.append(ws.cell(row = row,column = 1).value[:4]) # append dates by years
        y1.append(ws.cell(row = row,column = 8).value) # append number of people migrated into London
        y2.append(ws.cell(row = row, column = 2).value) # append numebr of people migrated into UK
        y3.append(ws.cell(row = row, column = 11).value) # append numebr of people migrated out London
```

```python
        y4.append(ws.cell(row = row, column = 4).value) # append numebr of people migrated out UK

    #print(x)
    #print(y1)
    #print(y2)
    #print(y3)
    #print(y4)

    source1 = ColumnDataSource(data = dict(dates = x, values = y1))
    source2 = ColumnDataSource(data = dict(dates = x, values = y2))
    source3 = ColumnDataSource(data = dict(dates = x, values = y3))
    source4 = ColumnDataSource(data = dict(dates = x, values = y4))


    p = figure(plot_width = 600, plot_height = 300,x_axis_label = "dates",y_range = (0,1000),
            y_axis_label = "Migration-input population",tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom",title="Long-term migration(London vs UK)") # Create a new figure

    p.line(x = "dates", y = "values", line_width = 2, source = source1, color = "black", legend = "London-in") # draw a line chart
    p.line(x = "dates", y = "values", line_width = 2, source = source2, color = "teal", legend = "UK-in") # draw a line chart
    p.line(x = "dates", y = "values", line_width = 2, source = source3, color = "chocolate", legend = "London-out") # draw a line chart
    p.line(x = "dates", y = "values", line_width = 2, source = source4, color = "darkred", legend = "UK-out") # draw a line chart
    p.legend.location = "top_left"
    p.legend.orientation = "horizontal"
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source1) # point of line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source2) # point of line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source3) # point of line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source4) # point of line chart

    return p


# Make line chart of employment (London vs UK) by years
def plot_employment_population():
    wb = openpyxl.load_workbook('data/underemployment.xlsx') # Import datasets
    ws = wb.get_sheet_by_name('Data')

    x = []
    y1 = []
    y2 = []
    for row in range(8,19):
        x.append(ws.cell(row = row,column = 1).value)
        y1.append(ws.cell(row = row,column = 2).value)
        y2.append(ws.cell(row = row,column = 7).value)

    #print(x)
    #print(y1)
    #print(y2)

    source1 = ColumnDataSource(data = dict(dates = x, values = y1))
    source2 = ColumnDataSource(data = dict(dates = x, values = y2))

    p = figure(plot_width = 600, plot_height = 300, x_axis_label = "dates",
                y_axis_label = "Employees number", tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom", title="Total employees/ self-employed(16+) number(London vs UK)") # Create a new figure


    p.line(x = "dates", y = "values", line_width = 2,source = source1, color = "black",legend = "London") # draw a line chart
    p.line(x = "dates", y = "values", line_width = 2,source = source2, color = "teal",legend = "UK") # draw a line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source1) # draw point of line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source2) # draw point of line chart

    p.legend.location = "top_left"
    p.legend.orientation = "horizontal"
    p.y_range.range_padding = 1

    return p

# Make line chart of underemployment rate(London vs UK) by years
def plot_underemployment_rate():
    wb = openpyxl.load_workbook('data/underemployment.xlsx') # Import datasets
    ws = wb.get_sheet_by_name('Data')

    x = []
    y1 = []
    y2 = []
    for row in range(8,19):
        x.append(ws.cell(row = row,column = 1).value)
        y1.append(ws.cell(row = row,column = 5).value)
        y2.append(ws.cell(row = row,column = 10).value)

    #print(x)
    #print(y1)
    #print(y2)

    source1 = ColumnDataSource(data = dict(dates = x, values = y1))
    source2 = ColumnDataSource(data = dict(dates = x, values = y2))

    p = figure(plot_width = 600, plot_height = 300, x_axis_label = "dates",
                y_axis_label = "Percentage underemployed", tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom", title="Underemployment rate(London vs UK)") #Create a new figure


    p.line(x = "dates", y = "values", line_width = 2,source = source1, color  = "black",legend = "London") #draw a line chart
    p.line(x = "dates", y = "values", line_width = 2,source = source2, color = "teal",legend = "UK") #draw a line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source1) #point of line chart
    p.circle(x = "dates", y = "values", fill_color = 'white',size = 3,source = source2) #point of line chart

    p.legend.location = "top_left"
    p.legend.orientation = "horizontal"
    p.y_range.range_padding = 1

    return p

# Make line chart fo short-term migration(London areas) by years
def plot_linechart_areas_short_term_migration():
    wb = openpyxl.load_workbook('data/Short term migration.xlsx') # Import datasets
    ws = wb.get_sheet_by_name('Data')
    x = []
    areas = []
    y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y20,y21,y22,y23,y24,y25,y26,y27,y28,y29,y30,y31,y32,y33,y34 = [],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]
    #Rearrange data
    for column in range (2,12):
        areas.append(ws.cell(row = row,column = 1).value)
        x.append(ws.cell(row = 5,column = column).value)
        y1.append(ws.cell(row = 6,column = column).value)
        y2.append(ws.cell(row = 7,column = column).value)
        y3.append(ws.cell(row = 8,column = column).value)
        y4.append(ws.cell(row = 9,column = column).value)
        y5.append(ws.cell(row = 10,column = column).value)
        y6.append(ws.cell(row = 11,column = column).value)
        y7.append(ws.cell(row = 12,column = column).value)
        y8.append(ws.cell(row = 13,column = column).value)
        y9.append(ws.cell(row = 14,column = column).value)
        y10.append(ws.cell(row = 15,column = column).value)
        y11.append(ws.cell(row = 16,column = column).value)
        y12.append(ws.cell(row = 17,column = column).value)
        y13.append(ws.cell(row = 18,column = column).value)
        y14.append(ws.cell(row = 19,column = column).value)
        y15.append(ws.cell(row = 20,column = column).value)
        y16.append(ws.cell(row = 21,column = column).value)
        y17.append(ws.cell(row = 22,column = column).value)
        y18.append(ws.cell(row = 23,column = column).value)
        y19.append(ws.cell(row = 24,column = column).value)
        y20.append(ws.cell(row = 25,column = column).value)
        y21.append(ws.cell(row = 26,column = column).value)
        y22.append(ws.cell(row = 27,column = column).value)
        y23.append(ws.cell(row = 28,column = column).value)
        y24.append(ws.cell(row = 29,column = column).value)
        y25.append(ws.cell(row = 30,column = column).value)
        y26.append(ws.cell(row = 31,column = column).value)
        y27.append(ws.cell(row = 32,column = column).value)
        y28.append(ws.cell(row = 33,column = column).value)
        y29.append(ws.cell(row = 34,column = column).value)
        y30.append(ws.cell(row = 35,column = column).value)
        y31.append(ws.cell(row = 36,column = column).value)
        y32.append(ws.cell(row = 37,column = column).value)
        y33.append(ws.cell(row = 38,column = column).value)
        y34.append(ws.cell(row = 39,column = column).value)


    p_areas = figure(plot_width = 1000, plot_height = 500, x_axis_label = "dates",
                     y_axis_label = "Migration population",y_range = (0,7000),
                    tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom", title="Areas of London short-term migration") #Create a new figure
    p2 = p_areas.line(x = x, y = y2, line_width = 1, color = "#000003",legend = "City of London") #draw a line chart
    p3 = p_areas.line(x = x, y = y3, line_width = 1, color = "#140D35",legend = "Barking and Dagenham") #draw a line chart
    p4 = p_areas.line(x = x, y = y4, line_width = 1, color = "#3B0F6F",legend = "Barnet") #draw a line chart
    p5 = p_areas.line(x = x, y = y5, line_width = 1, color = "#63197F",legend = "Bexley") #draw a line chart
    p6 = p_areas.line(x = x, y = y6, line_width = 1, color = "#8C2980",legend = "Brent") #draw a line chart
    p7 = p_areas.line(x = x, y = y7, line_width = 1, color = "#B53679",legend = "Bromley") #draw a line chart
    p8 = p_areas.line(x = x, y = y8, line_width = 1, color = "#DD4968",legend = "Camden") #draw a line chart
    p9 = p_areas.line(x = x, y = y9, line_width = 1, color = "#F66E5B",legend = "Croydon") #draw a line chart
    p10 = p_areas.line(x = x, y = y10, line_width = 1, color = "#FD9F6C",legend = "Ealing") #draw a line chart
    p11 = p_areas.line(x = x, y = y11, line_width = 1, color = "#FDCD90",legend = "Enfield") #draw a line chart
    p12 = p_areas.line(x = x, y = y12, line_width = 1, color = "#FBFCBF",legend = "Greenwich") #draw a line chart
    p13 = p_areas.line(x = x, y = y13, line_width = 1, color = "#a6cee3",legend = "Hackney") #draw a line chart
    p14 = p_areas.line(x = x, y = y14, line_width = 1, color = "#1f78b4",legend = "Hammersmith and Fulham") #draw a line chart
    p15 = p_areas.line(x = x, y = y15, line_width = 1, color = "#b2df8a",legend = "Haringey") #draw a line chart
    p16 = p_areas.line(x = x, y = y16, line_width = 1, color = "#33a02c",legend = "Harrow") #draw a line chart
    p17 = p_areas.line(x = x, y = y17, line_width = 1, color = "#fb9a99",legend = "Havering") #draw a line chart
    p18 = p_areas.line(x = x, y = y18, line_width = 1, color = "#e31a1c",legend = "Hillingdon") #draw a line chart
    p19 = p_areas.line(x = x, y = y19, line_width = 1, color = "#fdbf6f",legend = "Hounslow") #draw a line chart
    p20 = p_areas.line(x = x, y = y20, line_width = 1, color = "#ff7f00",legend = "Islington") #draw a line chart
    p21 = p_areas.line(x = x, y = y21, line_width = 1, color = "#cab2d6",legend = "Kensington and Chelsea") #draw a line chart
    p22 = p_areas.line(x = x, y = y22, line_width = 1, color = "#6a3d9a",legend = "Kingston upon Thames") #draw a line chart
    p23 = p_areas.line(x = x, y = y23, line_width = 1, color = "#ffff99",legend = "Lambeth") #draw a line chart
    p24 = p_areas.line(x = x, y = y24, line_width = 1, color = "#b15928",legend = "Lewisham") #draw a line chart
    p25 = p_areas.line(x = x, y = y25, line_width = 1, color = "#e41a1c",legend = "Merton") #draw a line chart
    p26 = p_areas.line(x = x, y = y26, line_width = 1, color = "#377eb8",legend = "Newham") #draw a line chart
    p27 = p_areas.line(x = x, y = y27, line_width = 1, color = "#4daf4a",legend = "Redbridge") #draw a line chart
    p28 = p_areas.line(x = x, y = y28, line_width = 1, color = "#984ea3",legend = "Richmond upon Thames") #draw a line chart
    p29 = p_areas.line(x = x, y = y29, line_width = 1, color = "#ff7f00",legend = "Southwark") #draw a line chart
    p30 = p_areas.line(x = x, y = y30, line_width = 1, color = "#ffff33",legend = "Sutton") #draw a line chart
    p31 = p_areas.line(x = x, y = y31, line_width = 1, color =
"#a65628",legend = "Tower Hamlets") #draw a line chart
    p32 = p_areas.line(x = x, y = y32, line_width = 1, color =
"#f781bf",legend = "Waltham Forest") #draw a line chart
    p33 = p_areas.line(x = x, y = y33, line_width = 1, color =
"#410967",legend = "Wandsworth") #draw a line chart
    p34 = p_areas.line(x = x, y = y34, line_width = 1, color =
"#6A176E",legend = "Westminster") #draw a line chart

    #Set legend location adn legemd orientation
    p_areas.legend.location = "top_left"
    p_areas.legend.orientation = "horizontal"
    #Set callback function after click checkboxes
    display_event = CustomJS(code="""
                                p2.visible = false;
                                p3.visible = false;
                                p4.visible = false;
                                p5.visible = false;
                                p6.visible = false;
                                p7.visible = false;
                                p8.visible = false;
                                p9.visible = false;
                                p10.visible = false;
                                p11.visible = false;
                                p12.visible = false;
                                p13.visible = false;
                                p14.visible = false;
                                p15.visible = false;
                                p16.visible = false;
                                p17.visible = false;
                                p18.visible = false;
                                p19.visible = false;
                                p20.visible = false;
                                p21.visible = false;
                                p22.visible = false;
                                p23.visible = false;
                                p24.visible = false;
                                p25.visible = false;
                                p26.visible = false;
                                p27.visible = false;
                                p28.visible = false;
                                p29.visible = false;
                                p30.visible = false;
                                p31.visible = false;
                                p32.visible = false;
                                p33.visible = false;
                                p34.visible = false;

                                if(cb_obj.active.includes(0)){
                                    p2.visible = true;
                                }
                                if (cb_obj.active.includes(1)){
                                    p3.visible = true;
                                }
                                if (cb_obj.active.includes(2)){
                                    p4.visible = true;
                                }
                                if (cb_obj.active.includes(3)){
                                    p5.visible = true;
                                }
                                if (cb_obj.active.includes(4)){
```

```
                                p6.visible = true;
                            }
                             if (cb_obj.active.includes(5)){
                                p7.visible = true;
                            }
                            if (cb_obj.active.includes(6)){
                                p8.visible = true;
                            }
                            if (cb_obj.active.includes(7)){
                                p9.visible = true;
                            }
                            if (cb_obj.active.includes(8)){
                                p10.visible = true;
                            }
                            if (cb_obj.active.includes(9)){
                                p11.visible = true;
                            }
                            if (cb_obj.active.includes(10)){
                                p12.visible = true;
                            }
                            if (cb_obj.active.includes(11)){
                                p13.visible = true;
                            }
                            if (cb_obj.active.includes(12)){
                                p14.visible = true;
                            }
                            if (cb_obj.active.includes(13)){
                                p15.visible = true;
                            }
                            if (cb_obj.active.includes(14)){
                                p16.visible = true;
                            }
                            if (cb_obj.active.includes(15)){
                                p17.visible = true;
                            }
                            if (cb_obj.active.includes(16)){
                                p18.visible = true;
                            }
                            if (cb_obj.active.includes(17)){
                                p19.visible = true;
                            }
                            if (cb_obj.active.includes(18)){
                                p20.visible = true;
                            }
                            if (cb_obj.active.includes(19)){
                                p21.visible = true;
                            }
                            if (cb_obj.active.includes(20)){
                                p22.visible = true;
                            }
                            if (cb_obj.active.includes(21)){
                                p23.visible = true;
                            }
                            if (cb_obj.active.includes(22)){
                                p24.visible = true;
                            }
                            if (cb_obj.active.includes(23)){
                                p25.visible = true;
                            }
                            if (cb_obj.active.includes(24)){
                                p26.visible = true;
                            }
                            if (cb_obj.active.includes(25)){
                                p27.visible = true;
                            }
                            if (cb_obj.active.includes(26)){
                                p28.visible = true;
                            }
                            if (cb_obj.active.includes(27)){
                                p29.visible = true;
                            }
                            if (cb_obj.active.includes(28)){
                                p30.visible = true;
                            }
                            if (cb_obj.active.includes(29)){
                                p31.visible = true;
                            }
                            if (cb_obj.active.includes(30)){
                                p32.visible = true;
                            }
                            if (cb_obj.active.includes(31)){
                                p33.visible = true;
                            }
                            if (cb_obj.active.includes(32)){
                                p34.visible = true;
                            }
                            """,args={'p2': p2, 'p3': p3, 'p4': p4,
                            'p5': p5,'p6': p6,'p7': p7,'p8': p8,'p9':
p9,'p10': p10,
                            'p11': p11,'p12': p12,'p13': p13,'p14':
p14,'p15': p15,'p16': p16,
                            'p17': p17,'p18': p18,'p19': p19,'p20':
p20,'p21': p21,'p22': p22,
                            'p23': p23,'p24': p24,'p25': p25,'p26':
p26,'p27': p27,'p28': p28,
                            'p29': p29,'p30': p30,'p31': p31,'p32':
p32,'p33': p33,'p34': p34
                            })
    #Set widgets checkboxes
    selection_box = CheckboxGroup(labels= [
    "City of London",
    "Barking and Dagenham",
    "Barnet",
    "Bexley",
    "Brent",
    "Bromley",
    "Camden",
    "Croydon",
    "Ealing",
    "Enfield",
    "Greenwich",
    "Hackney",
    "Hammersmith and Fulham",
    "Haringey",
    "Harrow",
    "Havering",
    "Hillingdon",
    "Hounslow",
    "Islington",
    "Kensington and Chelsea",
    "Kingston upon Thames",
    "Lambeth",
```

```python
    "Lewisham",
    "Merton",
    "Newham",
    "Redbridge",
    "Richmond upon Thames",
    "Southwark",
    "Sutton",
    "Tower Hamlets",
    "Waltham Forest",
    "Wandsworth",
    "Westminster"],active = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33])

    selection_box.js_on_click(display_event)
    row_1 = [p_areas, selection_box] #Make selection boxes located besides line chart
    return row_1


def plot_linechart_London_short_term_migration():
    wb = openpyxl.load_workbook('data/Short term migration.xlsx') # Import datasets
    ws = wb.get_sheet_by_name('Data')
    x = []
    y = []
    for column in range (2,12):
        x.append(ws.cell(row = 5,column = column).value)
        y.append(ws.cell(row = 6,column = column).value)

    p_London = figure(plot_width = 600, plot_height = 300, x_axis_label = "dates",
                y_axis_label = "Migration population",
                tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom", title="London short-term migration") #Create a new figure

    p_London.line(x = x, y = y, line_width = 2, color = "black", legend = "London")
    p_London.circle(x = x, y = y, fill_color = 'white',size = 3)
    p_London.legend.location = "top_left"
    p_London.legend.orientation = "horizontal"
    return p_London


def plot_shortterm_vbar():
    wb = openpyxl.load_workbook('data/Short term migration.xlsx')  # Import datasets
    ws = wb.get_sheet_by_name('Data')

    #Rearrange data, append them into new lists
    year2data = {}
    areas_list = []
    year_list = []
    for column in range (2,12):
        year = str(ws.cell(row = 5,column = column).value)
        year_list.append(year)
        data_dict = {}
        for row in range(7, 40):
            areas = ws.cell(row = row,column = 1).value
            if areas not in areas_list:
                areas_list.append(areas)
            data_dict[areas] = ws.cell(row=row, column=column).value

        year2data[year] = data_dict

    defult_year = year_list[0]
    counts_list = []
    display_data_dict = year2data[defult_year]
    for areas in areas_list:
        counts_list.append(display_data_dict[areas])

    source = ColumnDataSource(data=dict(areas_list=areas_list,
counts_list=counts_list))

    #plot new figure
    p_year = figure(plot_width=1000, plot_height=500,
               y_axis_label="Migration population", x_range=areas_list,
               y_range=(0, 6000),
tools="hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom",
               title="Areas of London short-term migration")
    #set colour
    colors = ['#000003', '#140D35', '#3B0F6F', '#63197F', '#8C2980',
              '#B53679', '#DD4968', '#F66E5B', '#FD9F6C', '#FDCD90',
              '#FBFCBF', '#a6cee3', '#1f78b4', '#b2df8a', '#33a02c',
              '#fb9a99', '#e31a1c', '#fdbf6f', '#ff7f00', '#cab2d6',
              '#6a3d9a', '#ffff99', '#b15928', '#e41a1c', '#377eb8',
              '#4daf4a', '#984ea3', '#ff7f00', '#ffff33', '#a65628',
              '#f781bf', '#410967', '#6A176E']
    #implement vertical bar chart
    p_vbar = p_year.vbar(x='areas_list', top='counts_list', source=source,
width=0.5, alpha=0.8, color=factor_cmap('areas_list', palette=colors,
factors=areas_list))
    p_year.xaxis.major_label_orientation = 1.2
    p_year.x_range.range_padding = 0.05
    p_year.legend.location = "top_left"
    p_year.legend.orientation = "horizontal"

    #set select menu
    select = Select(title="choose year", value=year_list[0],
options=year_list)

    #set callback funciton when click select menu
    def callback_select(attr, old, new):
        year = select.value
        counts_list = []
        display_data_dict = year2data[year]
        for areas in areas_list:
            counts_list.append(display_data_dict[areas])

        p_vbar.data_source.data['counts_list'] = counts_list



    select.on_change('value', callback_select)
    select.width = 100

    return p_year, select



def plot_multi_stackvbar():
    wb = openpyxl.load_workbook('data/LTIM reason (1).xlsx')  # Import datasets
    ws = wb.get_sheet_by_name('Data')

    #append data into lists
    year2data = {}
    years = []
    index = ['work definite job', 'work looking for a job', 'accompany', 'formal study', 'other']
    for row in range(4, 66):
        if ws.cell(row=row, column=2).value:
            year = str(ws.cell(row = row,column = 1).value)[:4]
            if year not in years:
                years.append(year)
            data_dict = year2data.get(year, {'In': [0,0,0,0,0], 'Out': [0,0,0,0,0], 'Net': [0,0,0,0,0]})
            data_in = data_dict['In']
            data_out = data_dict['Out']
            data_net = data_dict['Net']
            ws_in = [ws.cell(row = row,column = 2).value, ws.cell(row = row,column = 6).value,
                       ws.cell(row=row, column=10).value, ws.cell(row = row,column = 14).value,
                       ws.cell(row=row, column=18).value]
            ws_out = [ws.cell(row=row, column=3).value, ws.cell(row=row, column=7).value,
                     ws.cell(row=row, column=11).value, ws.cell(row=row, column=15).value,
                     ws.cell(row=row, column=19).value]
            ws_net = [ws_in[0]-ws_out[0], ws_in[1]-ws_out[1], ws_in[2]-ws_out[2], ws_in[3]-ws_out[3], ws_in[4]-ws_out[4]]

            total_in = [data_in[0]+ws_in[0], data_in[1]+ws_in[1], data_in[2]+ws_in[2], data_in[3]+ws_in[3], data_in[4]+ws_in[4]]
            total_out = [data_out[0]+ws_out[0], data_out[1]+ws_out[1], data_out[2]+ws_out[2], data_out[3]+ws_out[3], data_out[4]+ws_out[4]]
            total_net = [data_net[0]+ws_net[0], data_net[1]+ws_net[1], data_net[2]+ws_net[2], data_net[3]+ws_net[3], data_net[4]+ws_net[4]]

            data_dict['In'] = total_in
            data_dict['Out'] = total_out
            data_dict['Net'] = total_net
            year2data[year] = data_dict

            #arrange data into new form

    defaul_year = years[0]
    data_dict = year2data[defaul_year]
    df = pd.DataFrame(data_dict, index=index)
    x_index = df.index.tolist()
    type_work = df.columns.tolist()

    data = {'index': x_index}
    for type in type_work:
        data[type] = df[type].tolist()
    print(data)

    source = ColumnDataSource(data=data)

    #Plot new figure
```

```python
    p = figure(plot_width=1000, plot_height=500, y_axis_label="Migration population", x_range=x_index,
                y_range=(0, 1000), tools="hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom", title="reason for migration")

    p_vbar_in = p.vbar(x=dodge('index', -0.25, range=p.x_range), top='In', width=0.2, source=source, color="#ffff99",legend=value("In"))
    p_vbar_out = p.vbar(x=dodge('index', 0.0, range=p.x_range), top='Out', width=0.2, source=source, color="#b15928",legend=value("Out"))
    p_vbar_net = p.vbar(x=dodge('index', 0.25, range=p.x_range), top='Net', width=0.2, source=source, color="#e41a1c",legend=value("Net"))

    p.legend.location = "top_left"
    p.legend.orientation = "horizontal"

    menu = years
    dropdown = Dropdown(label=defaul_year, menu=menu) #Set new dropdown menu
    def callback_dropdown(attr, old, new): #set call back function of dropdown menu
        year = dropdown.value
        dropdown.label = year
        display_data_dict = year2data[year]
        p_vbar_in.data_source.data['In'] = display_data_dict['In']
        p_vbar_out.data_source.data['Out'] = display_data_dict['Out']
        p_vbar_net.data_source.data['Net'] = display_data_dict['Net']

        dropdown.label = year

    dropdown.on_change('value', callback_dropdown)
    dropdown.width = 100
    dropdown.height = 30

    return p, dropdown


#set callback functions of interfaces switch buttons
def callback1():
    layout_1.visible = True
    layout_2.visible = False
    layout_3.visible = False

def callback2():
    layout_1.visible = False
    layout_2.visible = True
    layout_3.visible = False

def callback3():
    layout_1.visible = False
    layout_2.visible = False
    layout_3.visible = True

#set interfaces switch buttons
button_1 = Button(label="short-term migration(London areas)")
button_2 = Button(label="migration vs employment(UK & London)")
button_3 = Button(label="reason for migration(UK)")
button_1.on_click(callback1)
button_2.on_click(callback2)
button_3.on_click(callback3)

p_year, select = plot_shortterm_vbar()
```

```
bt_row = row(button_1, button_2, button_3)
row_year = row(p_year, select)
row_1 = 
row(plot_linechart_London_short_term_migration(),plot_long_term_migration())
row_2 = row(plot_underemployment_rate(),plot_employment_population())

p_reason_mig, dropdown = plot_multi_stackvbar()
row_reason_mig = row(p_reason_mig, dropdown)

#set layout
bt_layout = layout(bt_row)
layout_1 = layout(row_year, plot_linechart_areas_short_term_migration())
layout_2 = layout(row_1,row_2)
layout_3 = layout(row_reason_mig)
layouts = layout(bt_layout, layout_1, layout_2, layout_3)

#show(layouts)
curdoc().add_root(layouts)


# python -m bokeh serve --show web.py
# By python 2.7


```