6CCS3PRJ
Data Visualisation for Migration Data

Final Project Report

Author: Yanpu Huang

Supervisor: Prof. Dr. Rita Borgo

Student ID: 1725298

April 21, 2020

Abstract

Human migration has formed the culture of each continent. In today's society, the formation of each country can not do without the keyword of immigration. The immigration data in London can show the total number of long-term and short-term immigrants in London, and the total number of immigrants in various regions of London. We can find out what other factors influence London and the UK by the trend of migration from these data, to find the connection between immigration and other thematic data. London Data Store is an official database website in London, which stores data statistics of various themes in London. The purpose of this project is to carry out data visualization processing for immigration data of London data store and some data of other topics and to provide a visualization website for users to do data analysis and interaction for the city of London.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Yanpu Huang
April 21, 2020

Acknowledgements

I would like to express my sincere thanks to my supervisor Professor. Dr. Rita Borgo. At the beginning of the project theme selection and project resources, she gave me a lot of help. And in the process of the project, she helped me to improve many points to think about, which helped me to expand my thinking. Thanks to Department of information staff and it services at King's College London. Thanks to the public data resources from the London Data Store official website.

Contents

1 Introduction **3**

1.1 Project Background . . . 3

1.2 Project Aim . . . 4

1.3 Report Structure . . . 5

2 Background **6**

2.1 Migration and Population . . . 6

2.2 Migration and Labours . . . 7

2.3 Migration and Education Level . . . 8

2.4 Short-Term Migration . . . 9

2.5 Long-Term Migration . . . 9

2.6 Reasons For Migration . . . 10

3 Working Process **12**

3.1 First Impression . . . 12

3.2 Working Plan . . . 13

3.3 Reality . . . 14

4 Design & Specification **15**

4.1 Design . . . 15

4.2 Specification . . . 19

5 Requirement & Implementation **20**

5.1 Requirement . . . 20

5.2 Implementation . . . 21

6 Legal, Social, Ethical and Professional Issues **30**

7 Results/Evaluation **31**

7.1 Software Testing . . . 31

7.2 Applied Analysis . . . 32

8 Conclusion and Future Work **38**

8.1 Unsolved Requirements . . . 38

8.2 Program Review/If I start from beginning . . . 38

8.3 Learn from work . . . 39

8.4 Future Development/Challenges . . . 41

A Appendix **42**

A.1 Definition . . . 42

A.2 Reading Resources/Data sets resources . . . 42

Bibliography . . . 42

B User Guide **44**

B.1 Instructions . . . 44

C Source Code **45**

C.1 Source Code . . . 45

Chapter 1

Introduction

1.1 Project Background

This project is about data visualisation from the London Data Store website. I download and extract data from this website, import it into my project, and visually process the data to get the display effect that I want. The goal of this project is to complete software to meet the needs of users for data analysis. Users can analyse the data in a more deeply way through visualisation and can interact with the software at the same time.

The theme I chose was immigration data in London and the UK. Immigration takes place in every country. As a developed country in the world, Britain has attracted numerous overseas immigrants. According to statistics, thousands of immigrants go into the UK every year, which has amounts of impact on the economy, employment, housing and other fields of the UK. Therefore, immigration is a field that can not be ignored for a country, and its importance is undeniable. So what do these immigrants want? Has their immigration had a good or bad impact on all areas of the UK? What factors affect migration? What factors do immigrants affect? These conclusions can be obtained through visual analysis of immigration-related data in the London Data Store.

When my tutor told me to go to the London Data Store website and search for a theme from this website to complete the visualisation project, I was attracted by the urban planning at the beginning. My idea at the start was to explore more potential areas of London by looking for housing, population, income and other data. However, in the process of looking for data, many datasets are not enough for me to complete this topic, so I began to search for other themes. In this process, I looked at the data of income, unemployment, employment and education. Finally, I found the complete data that I wanted to support the analysis in an immigration dataset called ”Migration indicators”. I was amazed by this, so I located the theme of my project and reported it to my tutor.

I chose to implement this project with Python and bokeh libraries and make it a web page. Because Python is easy to learn, it’s straightforward for me to get started for novices. At the same time, Python’s functions for web development are also potent. The reason I chose bokeh is that bokeh has powerful visualisation function. Its library has many visualisation options and widgets. The same thing is that it also has the characteristics of being easy to use. After comparing the evaluation of Plotly, Matplotlib and Seaborn on the Internet, I found that compared with bokeh, these visualisation development libraries are more complex. Although some of them can develop very complex visualisation projects, for my single-personal project, I don’t need to use such complex and powerful functions. At the same time, bokeh is a data visualisation library for web development, which is just in line with my project objectives. So I finally chose Python and bokeh.

1.2 Project Aim

The goal of my project is to complete a data visualisation website that supports user analysis. In this website, users can browse many visual forms of immigration data, and there are other aspects of data such as unemployment rate, employment data to compare, they can get the conclusion of the desired research direction through analysis.

The site needs to include accurate charts to represent the data, including histogram, line chart, stack chart, multiple column chart and various line charts. They must correspond with the data on the datasets accurately so that users can analyse the real and practical data. At the same time, these visualisation charts must be complete so that users can observe the trend in a sufficient range of data. On the web page, enough parts to support the user’s operation are also necessary. They will allow the user to choose the direction of observation and analysis.

1.3 Report Structure

This report covers multiple contents, including introductionbackground, development process specifications requirements design implementation, legal, social, ethical and professional issues evaluation analysis conclusion reference appendix.

Each content has its own topics.

Background contains the context by motivating the subject matter and relating it to existing published work. The background will include a critical evaluation of the existing literature in the migration and data visualisation area. I will also introduce the where the data sets come from. Sections of Background are Migration and Population, Migration and Labours, Migration and Education Level, Short-Term Migration, Long-term Migration.

Working process contains the planning of my whole work. Includes my first impressions, time arrangement(it will show by Gannt Chart), working plan, and the final attempt.

Design & Specification are split by design and specification. The design part contains design inspiration, visualisation design, interaction design. The specification part contains visualisation sketch, interaction on adobe.

Requirement & implementation are split by requirement and implementation. Requirement contains user requirement, functional requirement, non-functional requirement. Implementation contains programming language, visualisation tool, work flow, requirement solutions, forepart version, final version.

Legal, social, ethical and professional issues include the four topics as the section title shows.

Evaluation analysis includes software testing, applied analysis of three topics, they are short-term migration, migration employment, reason for migration.

The conclusion and future work contains unsolved requirements, program review(if I start from beginning), learning from work, Challenge, future development.

Chapter 2

Background

2.1 Migration and Population

Migrants for tens of thousands of years had been on the vanguard of the advancement of civilisations. The first era of globalisation in the second half of the nineteenth century was associated with the first mass voluntary movement of people, as millions of people migrated internationally in search of greater security and opportunity. The advent of steamships made long-distance travel more affordable, safer and quicker, facilitating voyage to the Americas, Southern Africa, and Australia. This ‘age of mass migration’ from around 1840 to the First World War in 1914, increased the working population of North America and Australia by at least a third, and Argentina by half, with the number of Europeans migrating rising from around 300,000 per year in the 1850s to over 3 million migrants per year in the early 20th century. The peak of the industrial revolution was the main period of British and German migration to North America, and between 1800 and 1860 two-thirds of the migrants to the U.S. were from Britain and 20 per cent from Germany, as displaced workers sought opportunity elsewhere. From 1860 to 1920 most of the 30 million immigrants to the U.S. came from Scandinavia, Ireland, Italy, Spain, and Eastern Europe. With many of the Irish and Eastern European migrants going first to Britain, the relative share of migrants in Britain and the United States was higher than today. (Lead author Professor Ian Goldin, Professor of Globalisation and Development at the University of Oxford”Migration and the Economy: Economic Realities, Social Impacts and

Political Choices”19).

The above historical information describes the impact of these immigrants on the population of different regions of the world. This graduation project opportunity allowed me to explore the migration datasets in London, from which I can analyse the relationship between population and migration population to realise the inspiration I got from this paper.

To realise this visual analysis, I downloaded the dataset of the population by borough 1939 to 2039.xlsx from the London data store. This dataset contains an excel workbook showing through population estimates and projects for the period 1939-2039 and a summary of population change in the capital. In 2015 London’s population survived its previous peak of 8.6 million people, it was closely related to the immigration situation in London at that time.

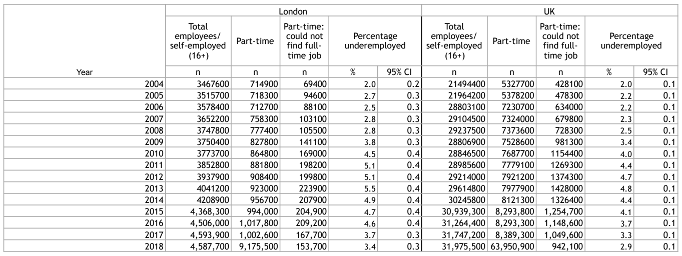
2.2 Migration and Labours

World War Two led to the death of well over 50 million people in combat and concentration camps and the displacement of over 30 million people. As the European economy began to recover in the 1950s, it experienced a growing labour shortage and the demand for migrants and for displaced people to settle increased. (Lead author Professor Ian Goldin, Professor of Globalisation and Development at the University of Oxford”Migration and the Economy: Economic Realities, Social Impacts and Political Choices”21).

As we can see from the reference, the author discussed about the labours market demand and the demand for migration. I began to think about what the migration alternation and the transformation of labor force brought to these two places if a group of people moved from one place to another

Therefore, I confirmed the relationship between migration and labours of a country are closely related, they also provide high volume for researching. As the population increases, the migration number is positive cross bonding to it, so the migration number will increase as well. As the migration number increases, the labours demand will increases as well. The internal reason that make this happened is worth for discussing.

So the above ideas inspired me, it could be work in a city as well, why don’t I look for the connection between immigrants and labour force in London? Labour means the number of people working in the market, which also means the chances of people finding jobs are closely related to the situation of immigrants. Immigration may have a positive or negative impact on employment and unemployment in London. So I downloaded underemployment.xlsx from the London data store, which records the total employees/self employed (16 +; part-time employees; part-time: could not find full-time job number; percentage underemployed from 2004 to 2018 in London and the UK. To a certain extent, it can give me enough data support to help me complete a visual chart of the underemployment rate and employment number.



	London					UK				
	Total employees/ self-employed (16+)	Part-time	Part-time: could not find full-time job	Percentage underemployed		Total employees/ self-employed (16+)	Part-time	Part-time: could not find full-time job	Percentage underemployed	
Year	n	n	n	%	95% CI	n	n	n	%	95% CI
2004	3467600	714900	69400	2.0	0.2	21494400	5327700	428100	2.0	0.1
2005	3515700	718300	94600	2.7	0.3	21964200	5378200	478300	2.2	0.1
2006	3578400	712700	88100	2.5	0.3	28803100	7230700	634000	2.2	0.1
2007	3652200	758300	103100	2.8	0.3	29104500	7324000	679800	2.3	0.1
2008	3747800	777400	105500	2.8	0.3	29237500	7373600	728300	2.5	0.1
2009	3750400	827800	141100	3.8	0.3	28806900	7528600	981300	3.4	0.1
2010	3773700	864800	169000	4.5	0.4	28846500	7687700	1154400	4.0	0.1
2011	3852800	881800	198200	5.1	0.4	28985600	7779100	1269300	4.4	0.1
2012	3937900	908400	199800	5.1	0.4	29214000	7921200	1374300	4.7	0.1
2013	4041200	923000	223900	5.5	0.4	29614800	7977900	1428000	4.8	0.1
2014	4208900	956700	207900	4.9	0.4	30245800	8121300	1326400	4.4	0.1
2015	4,368,300	994,000	204,900	4.7	0.4	30,939,300	8,293,800	1,254,700	4.1	0.1
2016	4,506,000	1,017,800	209,200	4.6	0.4	31,264,400	8,293,300	1,148,600	3.7	0.1
2017	4,593,900	1,002,600	167,700	3.7	0.3	31,747,200	8,389,300	1,049,600	3.3	0.1
2018	4,587,700	9,175,500	153,700	3.4	0.3	31,975,500	63,950,900	942,100	2.9	0.1

1

Figure 2.1: Underemployment.xlsx

2.3 Migration and Education Level

The gap in labor market outcomes between migrants and natives is closely linked to education level. to be specific, the lower the education level, the more favorable the labor market outcomes of migrants in comparison to natives. (Lead author Professor Ian Goldin, Professor of Globalisation and Development at the University of Oxford"Migration and the Economy: Economic Realities, Social Impacts and Political Choices"71).

As what we mentioned in section 2.2, the labours market is highly related to the migration. Then we can get the point that the education level is related to the migration too. The discussion of immigration and education level can be beneficial to the development of social civilization level. For a country, the level of education may vary in different regions due to different immigration situations.

Therefore, the research of migration and education level is potential and beneficial for the whole society.

2.4 Short-Term Migration

that of economic migrants from poor nations to advanced market economy countries. Most economic analyses agree that economic migration”as distinct from refugee flows. (”The Management Of International Migration: Short-Term”,author:Jeannette Money,1998)

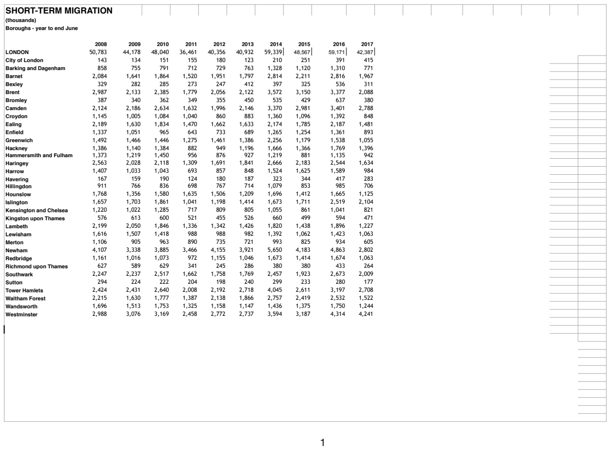
As far as I know, a large part of Chinese students’ families have short-term migration. During the period of studying in the United Kingdom or the United States and other countries, they will apply for a student visa. When they have enough time to study abroad and meet the criteria for local residence qualification, they will choose to apply for a local residence permit to achieve the purpose of short-term immigration.

There are many purposes for short-term migration, such as work, residence and investment. Short term migration brings economic benefits and labour force to a city or even a country, and the group of overseas students is the talent transmission channel of a town or a country. At the same time, the living needs of short-term immigrants have also brought income increase to all walks of life of the country.

From British news reports, we can often hear stories of immigrants, a large part of which are short-term immigrants. This drove me to understand the current situation of short-term migration. I downloaded the data set of ”short term migration.xlsx” from the London data store, which contains the number of short-term immigrants from 2008 to 2017 in various regions of London. These data can support my visual analysis on the time and area of short-term migration, and also enable users to understand the current situation of short-term migration in London.

2.5 Long-Term Migration

Long term migration refers to people who live in the country of immigration for a long time and enjoy similar benefits and obligations as the citizens of the country. Many of their aims are



SHORT-TERM MIGRATION
(thousands)
Boroughs - year to end June

	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
LONDON	50,783	44,178	48,040	36,461	40,356	40,932	59,339	48,567	59,171	42,387
City of London	143	134	151	155	180	123	210	251	391	415
Barking and Dagenham	858	755	791	712	729	763	1,328	1,120	1,310	771
Barnet	2,084	1,641	1,864	1,520	1,951	1,797	2,814	2,211	2,816	1,967
Bexley	329	282	285	273	247	412	397	325	536	311
Brent	2,987	2,133	2,385	1,779	2,056	2,122	3,572	3,150	3,377	2,088
Bromley	387	340	362	349	355	450	535	429	637	380
Camden	2,124	2,186	2,634	1,632	1,996	2,146	3,370	2,981	3,401	2,788
Croydon	1,145	1,005	1,084	1,040	860	883	1,360	1,096	1,392	848
Ealing	2,189	1,630	1,834	1,470	1,662	1,633	2,174	1,785	2,187	1,481
Enfield	1,337	1,051	965	643	733	689	1,265	1,254	1,361	893
Greenwich	1,492	1,466	1,446	1,275	1,461	1,386	2,256	1,179	1,538	1,055
Hackney	1,386	1,140	1,384	882	949	1,196	1,666	1,366	1,769	1,396
Hammersmith and Fulham	1,373	1,219	1,450	956	876	927	1,219	881	1,135	942
Haringey	2,563	2,028	2,118	1,309	1,691	1,841	2,666	2,183	2,544	1,634
Harrow	1,407	1,033	1,043	693	857	848	1,524	1,625	1,589	984
Havering	167	159	190	124	180	187	323	344	417	283
Hillingdon	911	766	836	698	767	714	1,079	853	985	706
Hounslow	1,768	1,356	1,580	1,635	1,506	1,209	1,696	1,412	1,665	1,125
Islington	1,657	1,703	1,861	1,041	1,198	1,414	1,673	1,711	2,519	2,104
Kensington and Chelsea	1,220	1,022	1,285	717	809	805	1,055	861	1,041	821
Kingston upon Thames	576	613	600	521	455	526	660	499	594	471
Lambeth	2,199	2,050	1,846	1,336	1,342	1,426	1,820	1,438	1,896	1,227
Lewisham	1,616	1,507	1,418	988	988	982	1,392	1,062	1,423	1,063
Merton	1,106	905	963	890	735	721	993	825	934	605
Newham	4,107	3,338	3,885	3,466	4,155	3,921	5,650	4,183	4,863	2,802
Redbridge	1,161	1,016	1,073	972	1,155	1,046	1,673	1,414	1,674	1,063
Richmond upon Thames	627	589	629	341	245	286	380	380	433	264
Southwark	2,247	2,237	2,517	1,662	1,758	1,769	2,457	1,923	2,673	2,009
Sutton	294	224	222	204	198	240	299	233	280	177
Tower Hamlets	2,424	2,431	2,640	2,008	2,192	2,718	4,045	2,611	3,197	2,708
Waltham Forest	2,215	1,630	1,777	1,387	2,138	1,866	2,757	2,419	2,532	1,522
Wandsworth	1,696	1,513	1,753	1,325	1,158	1,147	1,436	1,375	1,750	1,244
Westminster	2,988	3,076	3,169	2,458	2,772	2,737	3,594	3,187	4,314	4,241

Figure 2.2: Short-term migration.xlsx

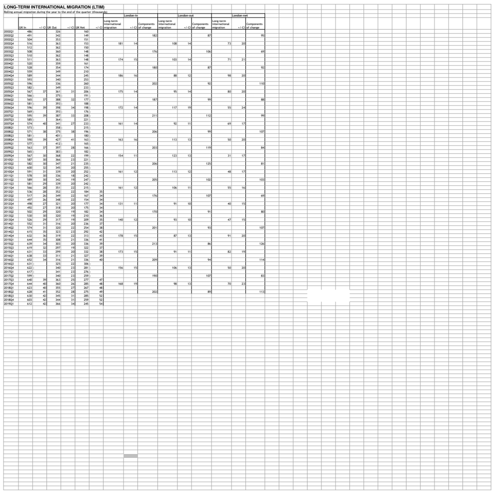
to be satisfied with the living environment of the country and want to live here for a long time. These people provide population input for the country or region, even marry and have children in the local area, bringing different cultures of their own country to a new country. Therefore, they bring not only economic benefits impact, resource impact, but also cultural impact.

Implementing the immigrant policy is a long process, so it required time to evaluate the effect of the policy. Two economists attempt a longer-term historical perspective in a new Centre for Economic Policy Research paper.

I believe that the research on long-term migration will give the government suggestions on the issue of long-term migration policy, and issue a reasonable and group benefits policy for the country or region. For the immigrant groups, they will see the immigration data of different years, observe the migration trend of a country, which makes the number of these immigrants change. They can also create their own decisions based on these studies. It will be a study that will benefit both the government and the migrants.

I downloaded the dataset Long term international migration.xlsx from London Data Store, it contains the data of long-term international migration from 2002-2018 to/from UK and London:

2.6 Reasons For Migration



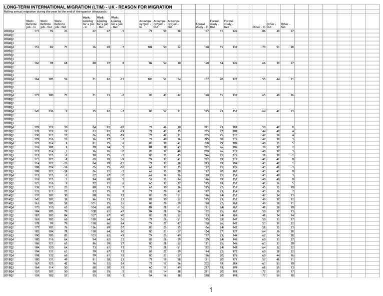
LONG-TERM INTERNATIONAL MIGRATION (LTIM)

Rolling annual migration during the year to the end of the quarter (thousands)

							London-in			London-out			London-net		
	UK In	+/-CI	UK Out	+/-CI	UK Net	+/-CI	Long-term international migration	+/-CI	Components of change	Long-term international migration	+/-CI	Components of change	Long-term international migration	+/-CI	Components of change
2002Q1	486		326		160										
2002Q2	491		342		149				182			87			95
2002Q3	504		353		151										
2002Q4	516		363		153		181	14		108	14		73	20	
2003Q1	512		362		150										
2003Q2	508		360		148				176			106			69
2003Q3	510		362		148										
2003Q4	511		363		148		174	15		103	14		71	21	
2004Q1	520		359		161										
2004Q2	528		354		174				180			87			92
2004Q3	559		349		210										
2004Q4	589		344		245		186	16		88	12		98	20	
2005Q1	593		340		253										
2005Q2	596		336		260				202			92			110
2005Q3	582	:	349		233	:									
2005Q4	567	37	361	31	206	:	175	14		95	14		80	20	
2006Q1	566	:	375	:	191	:									
2006Q2	565	37	388	32	177	:			187			99			88
2006Q3	581	:	393	:	188	:									
2006Q4	596	39	398	34	198	:	172	14		117	19		55	24	
2007Q1	569	:	393	:	176	:									
2007Q2	595	39	387	33	208	:			211			112			99
2007Q3	585	:	364	:	221	:									
2007Q4	574	40	341	27	233	:	161	14		92	11		69	17	
2008Q1	573	:	358	:	215	:									
2008Q2	571	38	375	38	196	:			206			99			107
2008Q3	581	:	401	:	180	:									
2008Q4	590	39	427	41	163	:	163	16		113	13		50	20	
2009Q1	577	:	412	:	165	:									
2009Q2	563	37	397	28	166	:			203			119			84
2009Q3	565	:	383	:	182	:									
2009Q4	567	30	368	22	199	:	154	11		123	13		31	17	
2010Q1	587	30	366	23	221	:									
2010Q2	582	30	347	21	235	:			206			125			81
2010Q3	600	32	345	20	255	:									
2010Q4	591	31	339	20	252	:	161	12		113	12		48	17	
2011Q1	578	30	336	18	242	:									
2011Q2	589	30	342	19	247	:			205			102			103
2011Q3	581	29	339	20	242	:									
2011Q4	566	28	351	22	215	:	161	12		106	11		55	16	
2012Q1	536	28	352	22	184	35									
2012Q2	517	26	349	22	167	34			176			107			69
2012Q3	497	26	348	22	154	34									
2012Q4	498	27	321	20	177	34	131	11		91	10		40	15	
2013Q1	492	27	318	20	175	34									
2013Q2	502	29	320	19	182	34			170			91			80
2013Q2	530	30	320	19	210	36									
2013Q4	526	29	317	19	209	35	140	12		93	10		47	15	
2014Q1	552	31	316	20	236	37									
2014Q2	574	31	320	22	254	38			201			93			107
2014Q3	615	35	323	23	292	42									
2014Q4	632	36	319	22	313	43	178	15		87	13		91	20	
2015Q1	644	35	308	21	336	41									
2015Q2	639	34	303	20	336	39			213			86			126
2015Q3	619	32	297	19	322	37									
2015Q4	631	33	299	20	332	38	173	15		91	11		82	19	
2016Q1	638	33	311	21	327	39									
2016Q2	652	34	316	21	336	40			209			94			114
2016Q3	631	:	325	22	306	:									
2016Q4	622	:	340	23	281	:	156	15		106	13		50	20	
2017Q1	617	:	341	23	276	:									
2017Q2	599	:	340	23	259	:			190			107			83
2017Q3	640	39	363	25	277	47									
2017Q4	644	40	360	26	285	48	168	19		98	13		70	23	
2018Q1	623	40	355	27	267	48									
2018Q2	628	41	352	28	275	49			202			89			113
2018Q3	630	42	345	31	285	52									
2018Q4	603	42	344	31	259	52									
2019Q1	612	42	366	34	245	54									

1

Figure 2.3: Long-term international migration.xlsx



LONG-TERM INTERNATIONAL MIGRATION (LTIM) - UK - REASON FOR MIGRATION

Rolling annual migration during the year to the end of the quarter (thousands)

	Work: Definite job - In	Work: Definite job - Out	Work: Definite job - Net	Work: Looking for a job - In	Work: Looking for a job - Out	Work: Looking for a job - Net	Accompany/join - In	Accompany/join - Out	Accompany/join - Net	Formal study - In	Formal study - Out	Formal study - Net	Other - In	Other - Out	Other - Net
2003Q4	115	92	23	62	67	-5	77	59	18	137	11	126	86	49	37
2004Q1															
2004Q2															
2004Q3															
2004Q4	153	82	71	76	69	7	102	50	52	148	15	133	79	51	28
2005Q1															
2005Q2															
2005Q3															
2005Q4	166	98	68	80	72	8	84	54	30	140	14	126	66	39	27
2006Q1															
2006Q2															
2006Q3															
2006Q4	164	105	59	71	82	-11	105	51	54	157	20	137	55	44	11
2007Q1															
2007Q2															
2007Q3															
2007Q4	171	100	71	71	73	-2	85	43	42	148	15	133	65	49	16
2008Q1															
2008Q2															
2008Q3															
2008Q4	145	136	9	75	82	-7	88	57	31	175	23	152	64	41	23
2009Q1															
2009Q2															
2009Q3															
2009Q4	129	119	10	64	92	-28	76	46	30	211	23	188	50	42	8
2010Q1	131	119	12	63	92	-29	78	43	35	235	27	208	44	40	4
2010Q2	130	113	17	66	85	-19	73	42	31	235	25	210	42	38	4
2010Q3	129	116	13	76	77	-1	76	40	36	245	30	215	42	39	3
2010Q4	122	114	8	81	75	6	80	39	41	238	29	209	40	35	5
2011Q1	116	108	8	79	74	5	81	38	43	232	26	206	39	37	2
2011Q2	117	114	3	76	76	0	85	37	48	239	26	213	40	37	3
2011Q3	113	115	-2	70	75	-5	80	35	45	246	21	225	38	39	-1
2011Q4	115	123	-8	69	78	-9	74	33	41	232	19	213	41	41	0
2012Q1	114	127	-13	64	79	-15	71	33	38	213	19	194	43	42	1
2012Q2	108	124	-16	65	75	-10	68	33	35	197	21	176	43	46	-3
2012Q3	109	127	-18	66	71	-5	63	35	28	187	20	167	43	43	0
2012Q4	113	115	-2	67	67	0	62	36	26	180	21	159	43	40	3
2013Q1	116	115	1	74	69	5	59	35	24	176	19	157	40	40	0
2013Q2	125	116	9	77	71	6	60	33	27	175	18	157	41	36	5
2013Q2	138	113	25	80	73	7	66	30	36	175	22	153	45	35	10
2013Q4	132	111	21	83	75	8	71	29	42	177	23	154	43	36	7
2014Q1	137	107	30	88	76	12	80	29	51	176	24	152	47	34	13
2014Q2	145	107	38	96	73	23	82	30	52	175	23	152	49	37	12
2014Q3	163	105	58	101	75	26	88	29	59	190	22	168	49	38	11
2014Q4	175	110	65	104	68	36	89	28	61	191	24	167	48	38	10
2015Q1	184	105	79	108	61	47	84	28	56	192	25	167	50	37	13
2015Q2	187	103	84	107	67	40	80	28	52	193	24	169	48	34	14
2015Q3	169	103	66	120	64	56	77	26	51	175	28	147	50	33	17
2015Q4	178	99	79	130	66	64	74	27	47	168	26	142	53	31	22
2016Q1	177	101	76	126	69	57	80	25	55	166	24	142	58	35	23
2016Q2	182	104	78	130	64	66	80	23	57	164	27	137	64	36	28
2016Q3	190	105	85	103	62	41	74	25	49	167	23	144	62	34	28
2016Q4	180	116	64	94	62	32	85	26	59	169	24	145	60	33	27
2017Q1	186	121	65	86	59	27	80	28	52	171	25	146	63	33	30
2017Q2	184	120	64	73	61	12	79	28	51	172	24	148	64	32	32
2017Q3	194	131	63	79	67	12	86	27	59	194	22	172	60	38	22
2017Q4	198	132	66	79	61	18	80	23	57	196	20	176	60	44	16
2018Q1	180	131	49	81	58	23	77	19	58	191	20	171	57	46	11
2018Q2	167	125	42	76	52	24	73	17	56	202	18	184	63	53	10
2018Q3	164	114	50	68	50	18	60	11	49	217	18	199	68	55	13
2018Q4	157	107	50	60	55	5	52	14	38	211	20	191	72	55	17
2019Q1	159	102	57	55	58	-3	54	16	38	218	20	198	77	59	18

1

Figure 2.4: Reason for migration

Chapter 3

Working Process

3.1 First Impression

When I received this project, my first impression was wondering what data visualisation was. My idea of the word data visualisation is that it often appears in advertisements on Web pages. I can't help but think of it, is it related to big data? What is visualisation? When the supervisor recommended my data visualisation books, and through my exploration of the keyword "data visualisation" in the network, and after enjoying many excellent visualisations works, I slowly understood the meaning of the word. Data visualisation is scientific and technological research on the visual representation of data. Among them, the visual representation of this data is defined as information extracted in some summary form, including various attributes and variables of corresponding information units. It is an evolving concept, and its boundaries are constantly expanding. It mainly refers to the more advanced technical methods, which allow the use of graphics, image processing, computer vision and user interface, through the expression, modelling and display of three-dimensional, surface, attribute and animation, to interpret the data visually. Compared with the specialised technical methods such as stereo modelling, data visualisation covers a lot of mechanical methods.

At the same time, this project is also significant for the exploration of London data store. I am very excited about the official database of London, an international metropolis. I

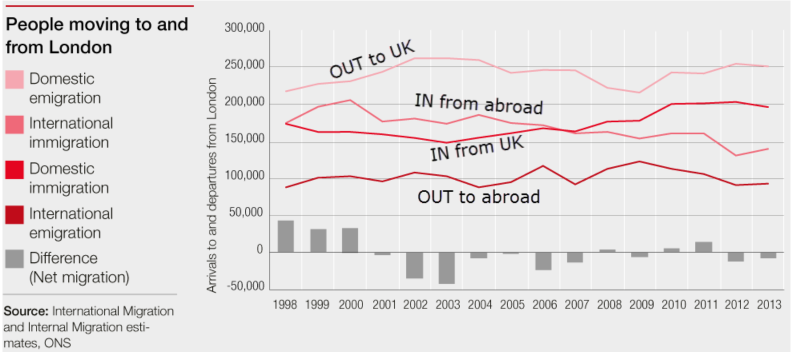


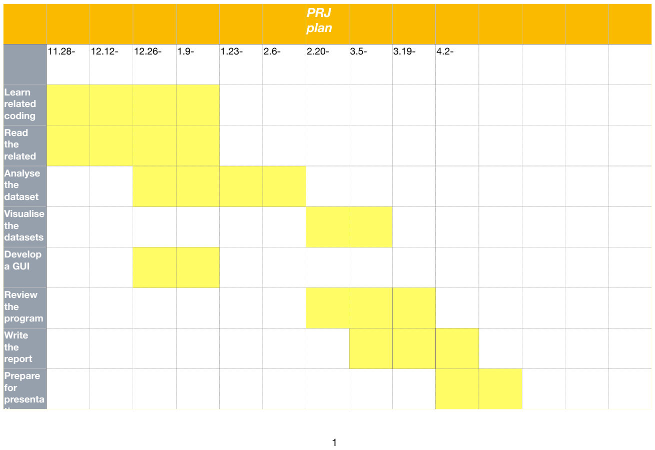
Figure 3.1: Online example of migration visualisation

began to think about what kind of achievement is valuable, what benefits it will bring to society and people, or what practical technical support it will bring to the government of London.

This project is my first step in data visualisation. It allows me to think about problems I never thought about and opens a new door for me.

3.2 Working Plan

After a meeting with the supervisor, she wanted us to use a Gantt chart to draw our plans. I divide my plan into learning-related coding, read the associated resource, analyse the dataset, visualise the datasets, develop a GUI, review the program, write the report, prepare for presentation.



							PRJ plan							
	11.28-	12.12-	12.26-	1.9-	1.23-	2.6-	2.20-	3.5-	3.19-	4.2-				
Learn related coding	🟨	🟨	🟨	🟨										
Read the related	🟨	🟨	🟨	🟨										
Analyse the dataset			🟨	🟨	🟨	🟨								
Visualise the datasets							🟨	🟨						
Develop a GUI			🟨	🟨										
Review the program							🟨	🟨	🟨					
Write the report								🟨	🟨	🟨				
Prepare for presenta										🟨	🟨			

Figure 3.2: Gantt chart

3.3 Reality

In reality, my actual completion is much longer than the plan. But fortunately, I took a faster time to complete the Python code learning, which did not affect my progress. When the tutor asked us to write BSPR, I didn't have a particular direction and idea. My idea was still vague, so I submitted a BSPR that was not very clear at that time. However, later, I accelerated the efficiency of drawing my initial London immigration data visualisation project and revised it several times in the later stage. There were three versions, which used two methods to realise the project, and my final version also met my requirements in operability and aesthetics.

Chapter 4

Design & Specification

4.1 Design

4.1.1 Design Inspiration

My design inspiration comes from the completed data visualisation project and data visualisation teaching website on the internet. Public projects on the internet always express beautiful and complete data. First of all, colour is a significant factor. Colour is the main impact on human's vision. It enables users to clearly distinguish the characteristics of different data when face to the visualisation charts. Second, choose the appropriate graph to represent different data. In the visual vocabulary website, the purpose of various charts is introduced in detail. Different types of chart within the category to form some initial ideas about what might work best. Third, it also needs careful design to put multiple data together for comparison. What kind of way they will produce accurate and beautiful visual effect combination, also can let users obtain data at the same time, and praise such design. In this book (Fundamentals of data visualisation, Claus O. Wilke), the author introduces many ways to combine data, such as visualising many distributions at once, multi-panel figures, etc., which can effectively compare data.

4.1.2 Visualisation Design

When I started designing visualisation charts, I found that I was just a beginner. In the beginning, I set myself the goal of simple line chart and histogram. A line chart can show the trend of time flow, data rising or falling. The histogram can explain the difference between different kinds of data. I intend to use these two kinds of images, even multiple histograms and multiple line chart. My choice of colour is bright and conspicuous so that users can distinguish data more clearly when browsing.

Short-Term Migration

When I designed the first dataset, short-term migration.xlsx, I chose the region as the x-axis variable and an annual number of immigrants as the y-axis variable. Therefore, the histogram will be the most appropriate representation chart. The histogram with different length and colour in each region will more clearly express the different total number of short-term immigrants in each district every year than the table of data. In the beginning, I had the idea of drawing a basic choropleth map. Still, the map could not sincerely show the specific value comparison of the number of immigrants in each region, so I gave up the idea and adopted the original Barchart design.

I also drew a multi-line chart to show the trend of short-term migration in each region as the year increased. In this multi-line chart, the x-axis becomes time. From 2008 to 2017, users will be able to observe the trend comparison of various regions.

Long-Term International Migration

The second dataset is long term international migration.xlsx. This dataset has the data of London input and London output, UK input and skout. They have different data every year, but the unit of the Y-axis is the same, so I decided to draw a multiple line chart to compare the four data. The four data will be used in different colours to achieve better contrast effect.

Employment And Underemployment

The third dataset is underemplacement.xlsx. This dataset has two data: the underemployment rate and total employees number. Because the two data units are different, I decided to draw two multiple line charts for them, and compare the unemployment rate in London and the UK, and the total employment in London and the UK.

Reason For Migration

The fourth data set is the LTIM reason.xlsx. This data set covers the causes of long-term international migration. We can find that each reason in this dataset contains three data: the UK in, the UK out and UK net. So we can draw three histograms for one reason, and group them into one group, thus forming multiple histograms. The x-axis will be the cause of each long-term international migration, and the y-axis will show the number of immigrants from each purpose.

4.1.3 Interaction Design

Interaction design is also essential for visualisation projects. It can increase the user's sense of experience and meet the user's various exploration purposes.

Interfaces Swich

In the design of the whole interface, I use three buttons to distinguish three topics of the panel. Each button corresponds to an interface layout. Click different buttons to display different interfaces. The first interface button is short-term migration(London areas). Click this button to view the histogram and multi-line chart of London's short term migration. The second interface button is migration vs employment(London UK). Click it to display the London short-term migration line chart, the London and UK long-term international migration input and output line chart, the London and UK employment line chart, and the London and UK unemployment rate line chart. The third interface button is the reason for migration(UK). Clicking this button will display the histogram of British immigration reasons.

Small Widgets

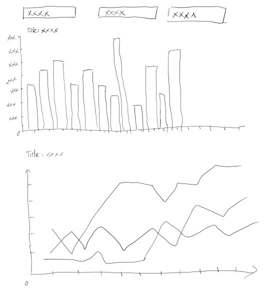
In terms of small widget design, I chose to select menu and checkboxes. In the widget design of the histogram of each region in the short term migration, I decide to use the select list. When the user clicks the menu, the years from 2008 to 2017 can be displayed. When he clicks one of the years, the histogram will jump to the corresponding year data. The user can choose the year he wants to browse for specific analysis. In the reason for migration icon, I also selected the select widget, which allows users to select different years to compare. On the short term migration multi-line map of each region, I choose to use the checkboxes widget, because, in this way, users can select multiple parts for comparative analysis, and selectively increase or decrease the number of regions they want to analyse.

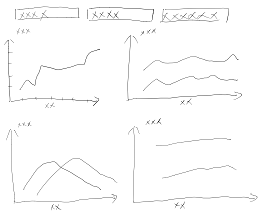
Passive Interaction

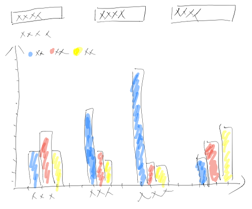
In the design of passive interaction, I added ”hover”, ”panbox”, ”zoom”, ”save” and other functions. ”Hover”'s function is that when the user's mouse moves to an absolute value of the chart, the chart will automatically display the corresponding accurate value next to the mouse. ”Pan” function is to move the chart, and users can drag the chart to the desired position. The function of the ”boxzoom” is to select and enlarge the chart. When the chart is chosen to the specified location, the chart in this part will be expanded and displayed. The function of ”save” is that when the user clicks, the chart will be downloaded from the browser to the user's computer.

4.2 Specification

4.2.1 Visualisation Sketch







Chapter 5

Requirement & Implementation

5.1 Requirement

5.1.1 User Requirements

- Users can analyse the migration data with other parts of data of London.(At least 5 topics)
- Users can select the immigration visualization theme they want to browse.
- Users can navigate the data of different years in the short term migration histogram.
- Users can select the immigration visualization theme they want to browse.
- Users can navigate the data of different years in the short term migration histogram.
- Users can browse different year data of multiple histograms of the reason for migration.
- Users can navigate the data from various regions in the short term migration line chart.
- Users can view the value of a point on the graph.
- Users can drag and drop graph.
- Users can enlarge the figure.
- Users can narrow the chart.

- Users can choose to expand the chart area.
- Users can save the chart.
- Users can reset the chart.

5.1.2 Functional Requirements

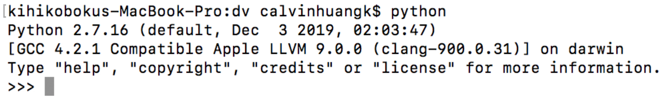
- The select menu of the short-term migration histogram can contain all the years 2008-2017.
- The checkboxes of the short term migration multiline chart can include all regions of London.
- The select menu of multiple histograms of the reason for migration can contain all years 2003-2019.
- All visualisation graphs will have the functions of selecting, zooming in, zooming out, displaying values, dragging, undo, reset and saving pictures.

5.2 Implementation

As I plan to design this visualization project as a website, which requires a tool that can not only complete the visualization requirements, but also construct the website. At the same time, I need to determine a programming language I want to use to write this program. In this section I will also show my workflow and achieved implementation of the requirement.

5.2.1 Programming Language

Python2.7



```
[kihikobokus-MacBook-Pro:dv calvinhuangk$ python
 Python 2.7.16 (default, Dec  3 2019, 02:03:47)
 [GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)] on darwin
 Type "help", "copyright", "credits" or "license" for more information.
 >>> 
```

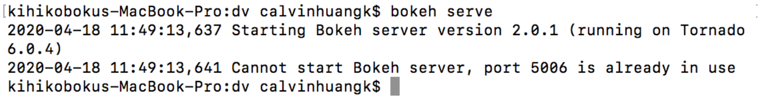
I started comparing Python, R, JavaScript, and then I chose Python as my programming language. In the process of machine learning at King's College London, the professor decided to use Python as our teaching programming language, so I have more experience in Python resource and syntax. At the same time, I compared the online comments on JavaScript, R and Python applied to visualization on the Internet. Python is the programming language that supports the most visual tool libraries, which enables users to make timely changes when selecting visualization tools, or to modify the design in the middle of the way. Not only that, but Python is also the easiest of the three for me, and its syntactic convenience stands out.

As we all know, Python is a cross-platform computer programming language. It is a high-level scripting language that combines interpretability, compilation, interactivity and object-oriented—originally designed to write automated scripts (shells), with the continuous update of versions and the addition of new language features, more and more are used for independent, large-scale project development. Because of the simplicity, readability and extensibility of Python language, it has been used by many users.

After I chose Python, I reported it to my project supervisor Rita. The use of Python is both unfamiliar and familiar to me because the use of Python visualization is still a new challenge to me. With the encouragement and advice from my tutor, I learned Python's syntax again and recorded all of Python's data import methods. Panda and openpyxl, which are commonly used by everyone, can read, write, extract unit data and self-made unit data. Import data is vital for my project because if we want to obtain all kinds of immigration data from the London data store; meanwhile, it is the first difficulty in importing and extracting its table data. At first, I thought about the method of import API from web pages. Still, later I found that there were only a few data providing API, some of which needed to be crawled by web crawlers, so I gave up the way of using JSON to enter the website API to extract data. I chose panda and openpyxl libraries, which can import and operate CSV files and xlsx files. They are speedy and convenient. Just download the form files provided on the website, and then import them into panda or openpyxl based on their format. The data extraction of the London Data Store has achieved the desired effect.

5.2.2 Visualisation Tool

bokeh1.4.0, Tornado 5.1.1



```
[kihikobokus-MacBook-Pro:dv calvinhuangk$ bokeh serve
 2020-04-18 11:49:13,637 Starting Bokeh server version 2.0.1 (running on Tornado
 6.0.4)
 2020-04-18 11:49:13,641 Cannot start Bokeh server, port 5006 is already in use
 kihikobokus-MacBook-Pro:dv calvinhuangk$ 
```

On the way of looking for visualization tools, I searched many materials online, many articles and blogs recorded the users' evaluation and experience records after using these visualization tools. To a certain extent, this makes me very confused, because as a novice in visualization, I have never been exposed to visualization tools at all. I don't know which tool is the most suitable one for me. I need to choose a visualization tool that can be combined with Python and is ideal for me to learn and website develop.

I emailed Rita, my tutor, to recommend some proper visualization tools. In a meeting with her, she approved plot, bokeh, Seaborn. I went to the Internet and focused on these three kinds of software.

Plotly is a common and powerful open source data visualization framework. It can display information by building interactive charts in the form of web-based on browser display and can create dozens of exquisite charts and maps.

Bokeh is a visualization tool developed for browsers. Like Matplotlib, bokeh has a series of API interfaces. For example, glyphs interface, which is very similar to the artists' interface in Matplotlib, is mainly used to draw ring graph, square graph and polygon diagram. Recently, bokeh has opened a new graphic interface, which is used primarily for process dictionary data or data frame data, and to draw can map.

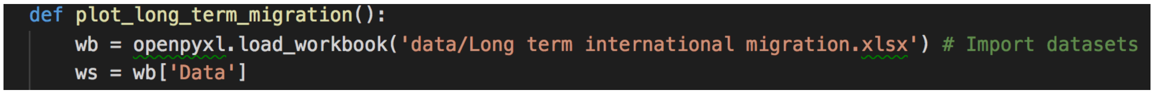
Seaborn is a more advanced API encapsulation based on Matplotlib, which makes drawing easier. In most cases, using Seaborn can make beautiful drawings, while using Matplotlib can make drawings with more characteristics. Seaborn should be seen as a complement to Matplotlib, not a substitute. At the same time, it can be highly compatible with NumPy and pandas data structures and statistical models such as SciPy and statsmodels.

For me, bokeh is an interactive visual class library based on Python language displayed in new browsers. Bokeh provides a fast and straightforward high-performance interactive, elegant and concise chart making based on big data and streaming data. These characteristics are just in line with my original intention. I am a personal project, which does not need too many intricate functions. At the same time, bokeh can display the project directly in the browser, which can achieve the purpose that I want to complete the data visualization web page display. Bokeh's data-based function is convenient and beautiful, which is my favourite display effect, so I chose it for decisively. Plotly used to be my first choice, but because of the related guides online are not that much as bokeh, so I gave up Plotly. And seaborn has a comprehensive list of its functions, but to develop web application, I don't need too many complicated functions. So in the end, bokeh wins.

5.2.3 Workflow

Import Data

When I started programming, importing data became my first job. I decided to draw a simple line graph from the data set of long-term international migration.xlsx, so I set up a function named ”plotLongTermMigration()”. I import data through the openpyxl library and use the function of openpyxl, loadWordbook function. When the data is imported successfully, I will form different lists with the cells I need in the table, so that the abscissa and ordinate data of the line chart can be obtained. Later, I used openpyxl for short-term migration.xlsx, LTIM reason.xlsx, underemplacement.xlsx to import and collect data successfully.

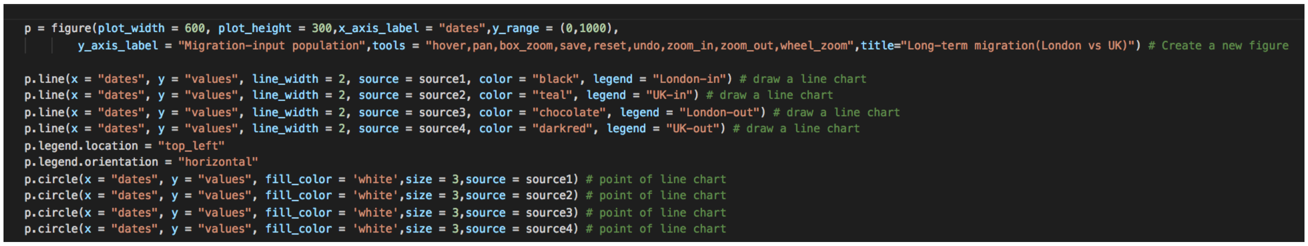


```
def plot_long_term_migration():
    wb = openpyxl.load_workbook('data/Long term international migration.xlsx') # Import datasets
    ws = wb['Data']
```

Figure 5.1: Pieces of data inport codes

Build Charts

In the process of drawing, bokeh's simple grammar helped me a lot. First, I create a figure of the desired size, in which many attributes can be set, such as Y-axis label, tools bar, Y-axis and X-axis length, etc. Then I use the existing chart generation functions of bokeh, such as line (), vbar (), to generate various visualisation charts. I used histogram and multi-line chart in the theme of short-term immigration, and I used line chart, multi-line chart and dot chart in the topic of immigration and employment. I used multi histogram in the theme of immigration reasons. In the function of creating a graph, the setting up of each attribute is critical, such as colour, label, X-axis, Y-axis, and even a significant feature of bokeh: columndatasource, which can directly arrange the data into a column list form, so that the data can instantly generate the corresponding chart. The display between the chart and chart also needs the layout function to arrange. Row(), column(), layout(), gridplot(), these methods can arrange the chart to form an orderly arrangement in the web page for users to browse.



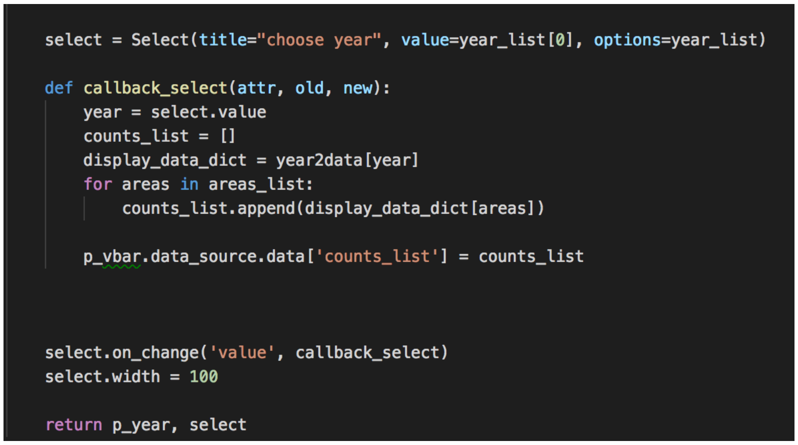
```
p = figure(plot_width = 600, plot_height = 300,x_axis_label = "dates",y_range = (0,1000),
        y_axis_label = "Migration-input population",tools = "hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom",title="Long-term migration(London vs UK)") # Create a new figure

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
```

Figure 5.2: Pieces of figure plotting code

Set Widgets

In the process of widget design, I used the select menu, dropdown menu and checkboxes. These three parts are respectively built in the function of plot charts, and then a callback (update) method is created in the function of setting up the image, to achieve the effect of the callback of widgets button function. Among them, in the bar charts of short term migration, I used the select menu, which will cover all the years of 2008-2017. Each time I click one of the years, I can call up the data of the selected year and show it in the chart. I choose to use checkboxes in the polyline chart of short term migration, and I use js code callback in its callback function, which refers to JavaScript code. In the image of the reason for migration, I choose to use the dropdown menu and then use the callback of the transformation data.

In the process of implementing the page switch button, I set up three buttons, which are connected by the onclick method and callback method, respectively. In the callback method, the visible method is used to display different layouts.

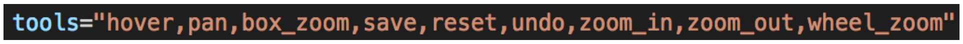


```
select = Select(title="choose year", value=year_list[0], options=year_list)

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
```

Figure 5.3: Pieces of widgets setting codes

5.2.4 Small Interaction

I used hover, pan, boxzoom, save, reset, undo, zoom in, zoom out, wheelzoom and other methods in the tools module to set them in the attribute of chart during the plot chart. The tools contain passive and active interaction.



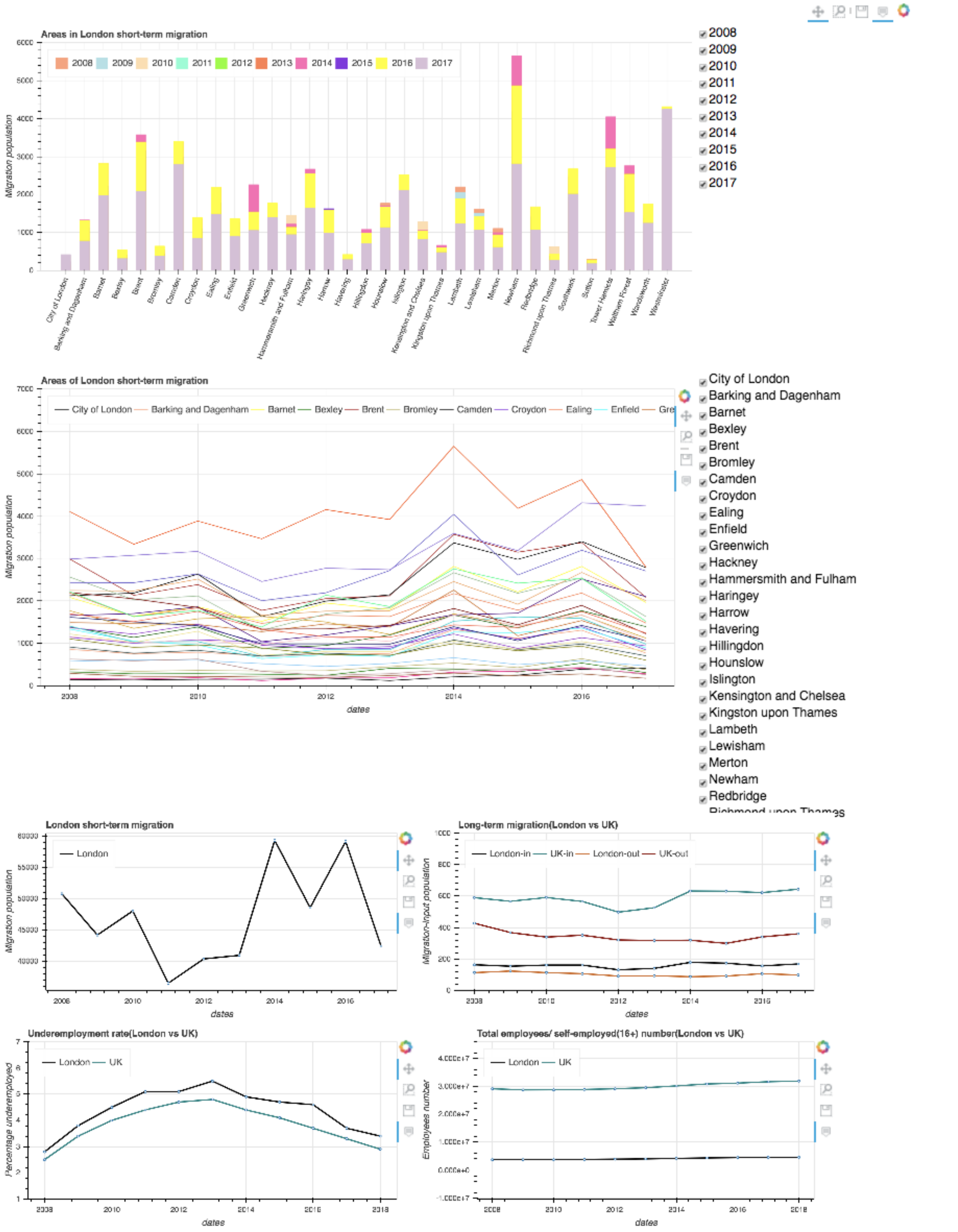
```
tools="hover,pan,box_zoom,save,reset,undo,zoom_in,zoom_out,wheel_zoom"
```

Figure 5.4: Pieces of interaction-tool implementation codes

5.2.5 Forepart Version

In earlier versions, I didn't know about how to implement interaction, so I could only sort and combine multiple images in layout to form an interface.

We can find that the limitations and visual clutter of this website have brought a lot of troubles to user analysis. Without the guidance of a small topic, users will not know where to start when facing the visualisation of many different data. At the same time, the lack of interaction design leads to the loss of many experiences for users of this version of the website. In Figure 5.5, in the short term migration column chart, we can see the checkboxes of the interactive selection at that time, which enables us to select multiple column charts. But the overlapping of histogram blocks a lot of data that should have been seen by users, so the design of checkboxes is meaningless, which will affect the user's independent analysis of data. Therefore, in the later version, I cancelled the plan and changed it to select menu.



☑ 2008
☑ 2009
☑ 2010
☑ 2011
☑ 2012
☑ 2013
☑ 2014
☑ 2015
☑ 2016
☑ 2017

☑ 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
☑ Lewisham
☑ Merton
☑ Newham
☑ Redbridge
☑ Richmond upon Thames

Figure 5.5: Earlier version

5.2.6 Final Version

In the final version, I achieved the interaction design of the web interfacethe web could be ran on the bokeh serve.

We can see that there are three main buttons to switch the interface, to enable users to browse immigration data of different topics. At the same time, in Figure 5.6, checkboxes were changed to select menu design. Compared with the previous version, I also added the data visualization chart of the theme reason for migration.

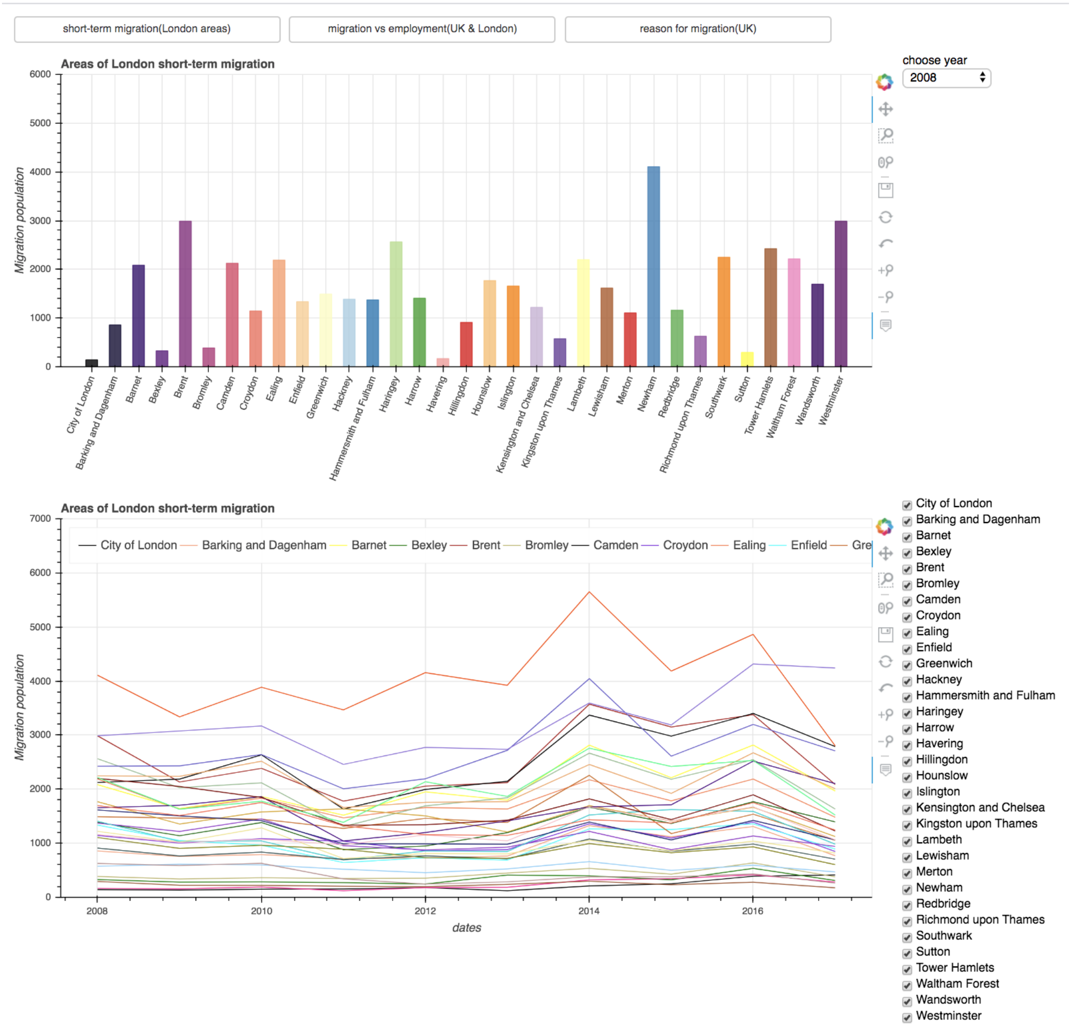


Figure 5.6: First interface

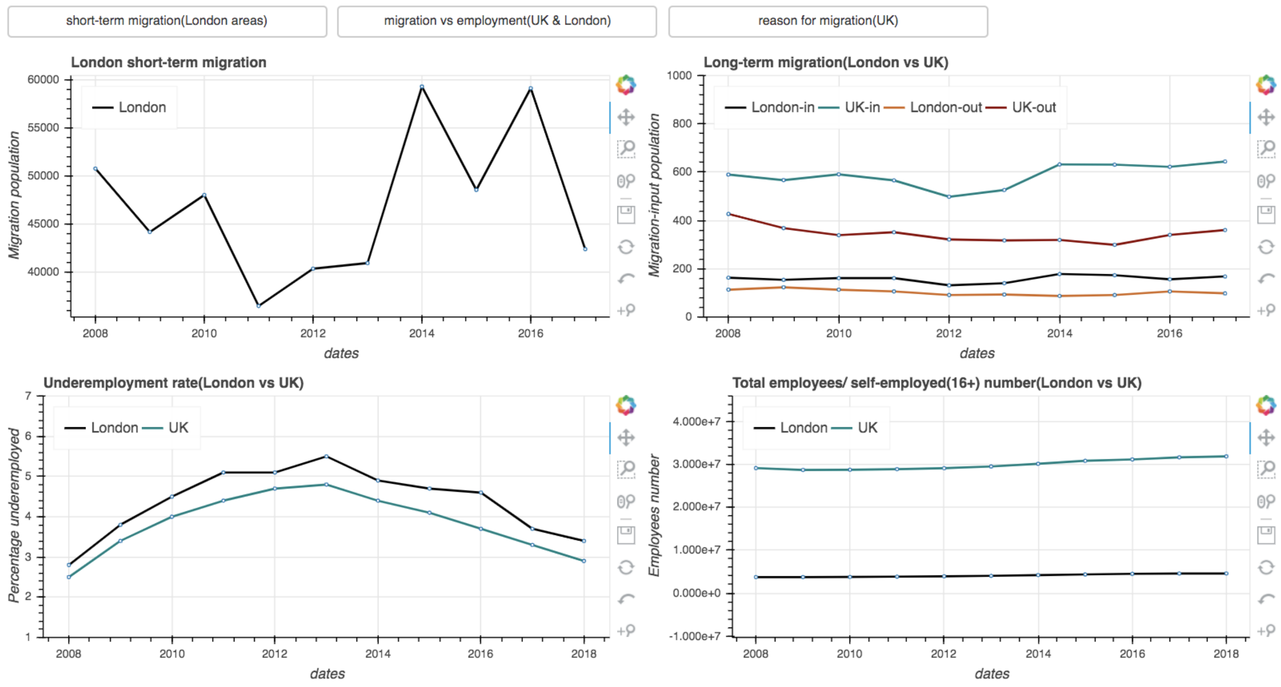


Figure 5.7: Second interface

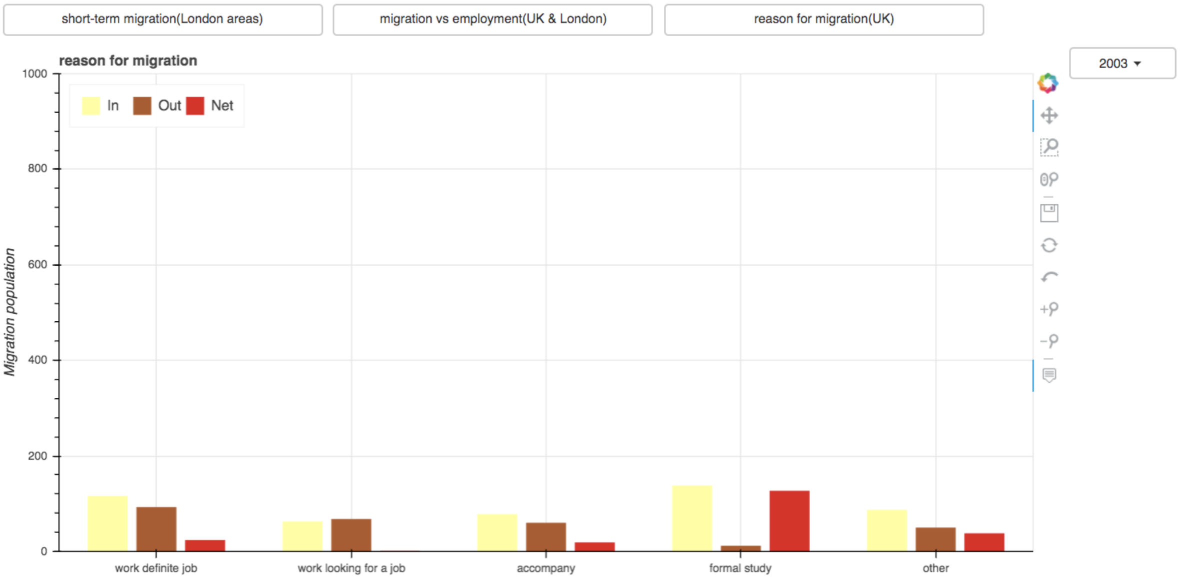


Figure 5.8: Third interface

Chapter 6

Legal, Social, Ethical and Professional Issues

The project is mainly based on data visualization and analysis, involving the real immigration data of the UK and London of the London data store, and the results do not represent any political position of the project author.

During design and implementation of the project, great concern has been shown to abide by the Code of Conduct Code of Good Practice which is issued by the British Computer Society (BCS). With any project, code of conduct should be abided by because if not can have serious legal and ethical implications. Rules also help protect work an intellectual property from being used unlawfully. Using material without authorisation of the owner is regarded unethical. Great care has been taken in this project to make sure any Open-Source code or libraries used are explicitly stated. This project consists of: my own work; Open-Source libraries on the internet.

The software consists of my own work unless explicitly stated so.

Chapter 7

Results/Evaluation

7.1 Software Testing

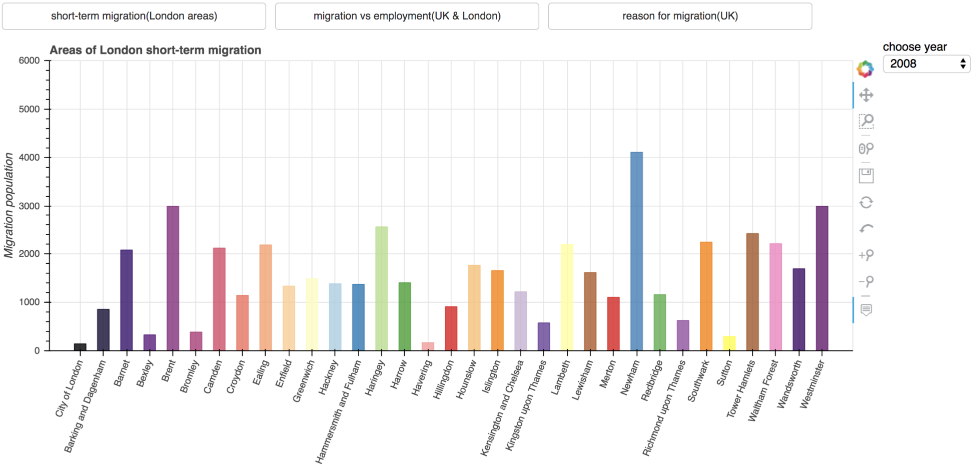
I let a few friends experience my web and evaluate it. In terms of interaction, most of the evaluations given are smooth and straightforward, and the communication of the site is also direct and clear. In terms of browsing visual data, the data visualisation of immigrants accounts for the majority, and the data analysis of immigrants may take the most percentage. The raters can read the accurate trend of immigration data. Still, they also admit that there are too few other data to compare with immigration data, only the number of employees, unemployment rate and immigration reasons. This is a common shortcoming after evaluation. There are too few comparative analysis charts for this project, most of which can only analyse the time trend of immigrants; comparative analysis between regions; comparative analysis between long-term and short-term immigrants, etc.

At the same time, I also encountered a problem that hasn't been solved when I was coding, that is, when I was in plotline chart of short term migration, there were too many legends corresponding to the broken lines of this chart so that I couldn't see all the legends in a plot. I hope that in the future, we can try to create a new blank plot next to the line graph, and input the legend into the blank plot so that we can all represent it.

7.2 Applied Analysis

7.2.1 Short-Term Migration(London Areas)

Bar Chart



choose year 2008

Figure 7.1: short term migration bar chart(London areas)

In Figure 7.1, we can take the region as the comparison unit and make a comparative analysis from the top value of the histogram. We can see that except in 2017 (the most significant number of short-term immigrants in 2017 is Westminster, reaching more than 4000), the region with the most significant amount of short-term immigrants is Newham. The total value of short-term immigrants in Newham in 2014 reached more than 5000, which is the only region with more than 5000 from 2008-2017 in all areas. The lowest number of short-term immigrants in 2008-2017 was the City of London, Havering and Sutton.

We can see that there is a significant gap in the number of immigrants between different regions of London, and the number of immigrants is not even among different areas. 2009 is the year in which the number of immigrants from different regions is most averagely distributed to each other.

Line Chart

In the line chart, we can observe the trend of the number of short-term immigrants in different regions of London with a float of years. In the Figure 7.2, we can see that before 2014, the

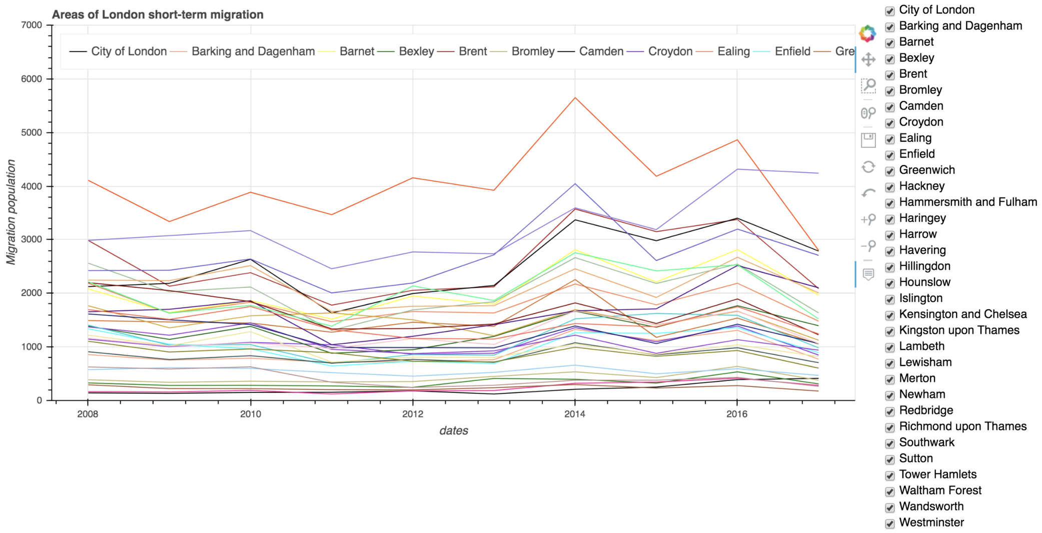


Figure 7.2: short term migration linec chart(London areas)

number of short-term immigrants in the various areas was in a general trend of growth, accompanied by small sections increase or decrease. The general trend in the short-term migration in 2014-2017 is a gradual decline.

City of London, Bexley, Bromley, Havering and Sutton are in a relatively gentle trend. In the graph, they approximately parallel to X-axis, indicating that the short-term immigrants in these areas are not popular and could be regarded as the areas with more indigenous people. However, in areas with a large number of short-term immigrants, the fluctuation is relatively large, such as Newham, Westminster and Brent. Harrow, Haringey and other regions are located in the middle height area of the line chart, which belongs to the trend of slight fluctuation. By 2017, the area with the most significant increase in the immigration area is Westminster. In 2017, the number of short-term immigrants in Westminster reached the highest in all regions, more than one thousand short-term immigrants units ahead of the second region Newham.

7.2.2 Migration & Employment

At this panel we have 4 charts, includes long term migration of UK and London, Short-term migration of London. they are build here for compare with the other two charts which are total employee number and the underemployment rate of London and UK.

Employee Number

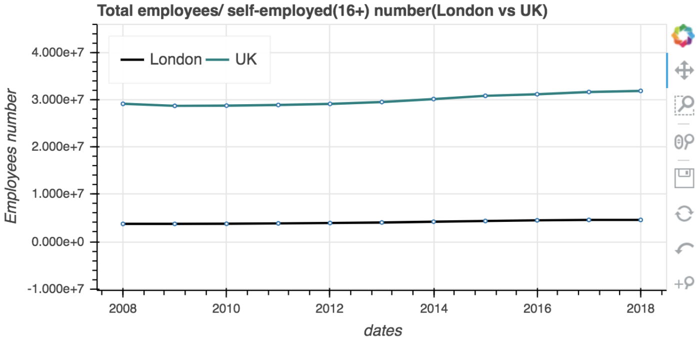


Figure 7.3: Total employees/self-employed(16+) number(London vs UK, 2008-2017)

In Figure 7.3, we can analyse that the total employee number in London and the UK is an increasing trend from 2008-2017, which is relatively gentle, and the trend is close to the parallel line. The UK employment curve has a higher slope and slightly steeper than London's.

Underemployment Rate

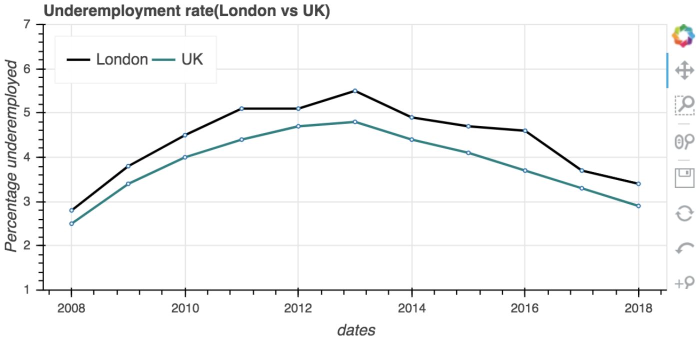


Figure 7.4: Underemployment rate (London vs UK, 2008-2017)

In Figure 7.4, we can analyse that the underemployment rate in London and the UK is increasing first and then decreasing from 2008-2017. And unexpectedly, the underemployment rate in London is more significant than that in the UK, because the London line chart is at the top of the UK line chart.

The underemployment rate peaked in London and the UK in 2013. In 2008, the underemployment rate in London and the UK was the lowest, and the gap between them was the smallest. The underemployment rate in London and the UK had the most significant difference in 2016.

Migration & Employment Comparing

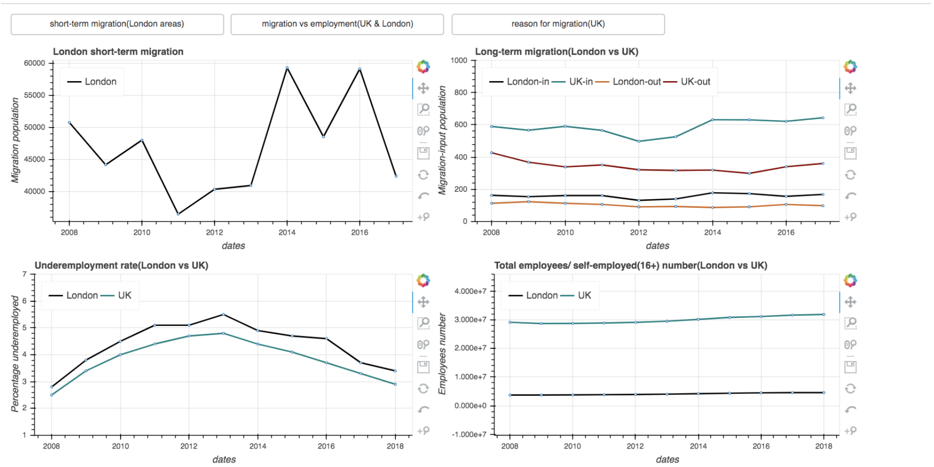


Figure 7.5: Second Interface

In Figure 7.5, we can compare and try to find the connection between employment and migration, including long-term international migration and short-term migration.

We can find that in the long run, the comparison between the long-term international immigration input curve and employment number curve is positively correlated. Whether UK or London, the final value they reach is higher than the starting value (although the international immigration input curve had a slight depression in 2012-2013, it does not affect the last growth trend). The employment number curve in London is as flat as the long-term international immigration input curve in London, and the slope is not as high as the employment curve and long-term international immigration input curve in Britain.

There are differences in the output curve. When the international migration output curve is relative to the employment number curve, the UK one shows a negative correlation, while London one indicates a positive relationship. Although there is a positive correlation between the employment number in London and the output of international migration, its slope is not as big as the input curve of international migration. So we can conclude that employment in the UK and London is positively correlated with long-term international migration net number(net = input - out).

Now watch the underemployment chart. We have analysed this curve before as a trend of growth firstly then followed by a decrease. The peak of growth is 2013, which is the same time as the curve of long-term international migration input depression in 2012-2013. At the same time, in the comparison between the underemployment rate curve and the short-term migration curve, we can find that the bottom of the short-term migration curve is just 2011-2013, which also matches the peak of the unemployment rate curve. We can boldly conclude that perhaps the increase in the unemployment rate is related to the decrease in immigrants' input, and there may be many factors among them. My analysis is that the decline of immigrants' input may lead to a reduction of local job opportunities, but it needs to be discussed, which is not a confirmed conclusion.

7.2.3 Reason For Migration

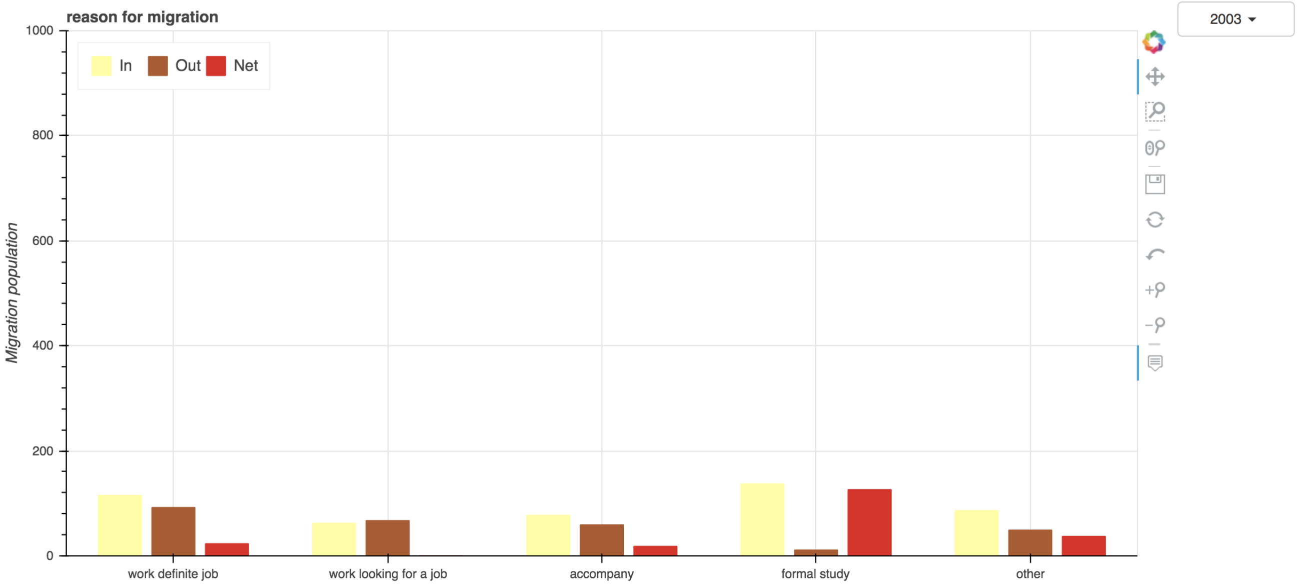


Figure 7.6: Migration Reason 2003

In Figure 7.6, we can see the multiple histograms of immigration reasons, which shows the number of reasons for immigration input, immigration output and net increase (decrease) of immigration in the UK.

When I browse the data of each year, I find that the immigration output of ”work looking for a job” module is the only one who has more immigration output than the input. ”Formal study” is the module in which the input of immigrants is much higher than the output, has the greatest positive net number. We can conclude that the reason for the most substantial proportion of the UK's immigration output is to leave the UK to find jobs. In contrast, the most significant percentage of immigration input is the international students who come to study in the UK.

When browsing different years, it can be found that before 2010, Britain's immigration input and output were basically no more than 200. However, after 2010, there has been a qualitative leap in the import and export of immigrants. We can see in the Figure 7.7, in 2010 the reasons for the migration input had generally reached more than 200, especially the import of international students, which has reached more than 900 and has lasted until 2011. It's astounding. After 2011, global student input has never reached more than 900. We can conclude that, combined with the previous long-term migration chart and short-term migration chart, 2010 is indeed a wave of international immigration into the UK.

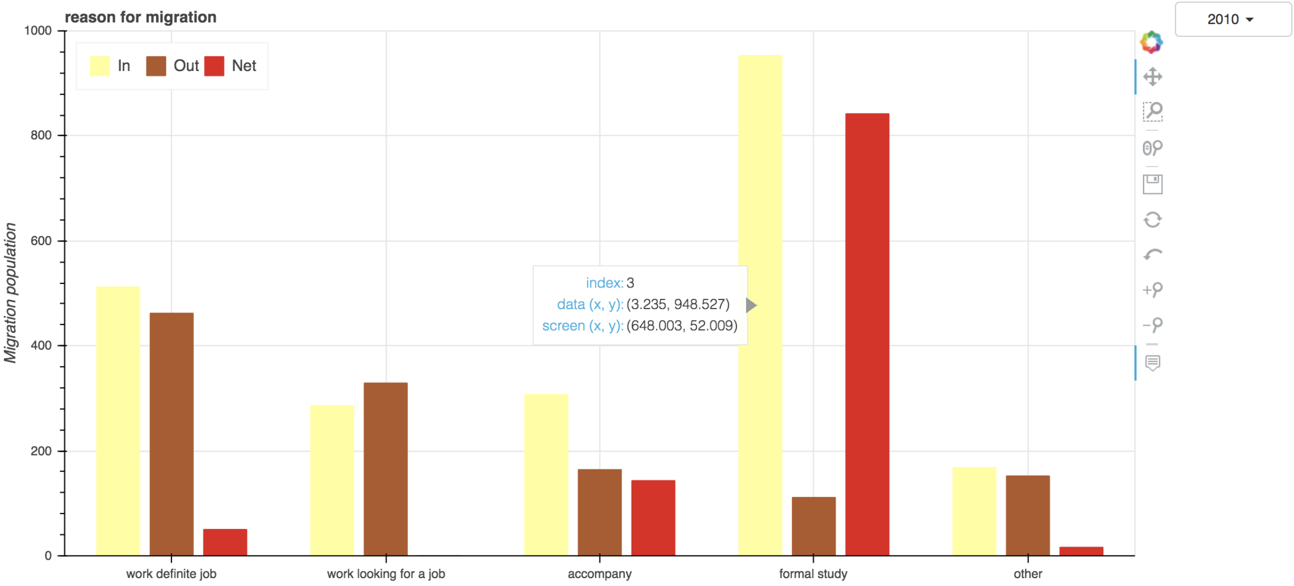


Figure 7.7: Migration Reason 2010

Chapter 8

Conclusion and Future Work

8.1 Unsolved Requirements

- Users can analyse the migration data with other parts of data of London.(At least 5 topics)

I didn't implement this user requirement. Because I was not familiar with the interaction design during the design process, so I spent too much time in the interaction design. Resulting in no time to import data of other topics to compare with the immigration data, so I didn't complete the user requirements of "at least five topics and immigration data comparative analysis".

8.2 Program Review/If I start from beginning

Looking back at the process of doing this project, I found it essential to find a clear goal at the beginning. Maybe it's because we didn't decide which topics could be discussed and contributed to society at the beginning, which led to wasting a lot of time to make meaningless choices. In the design process, the typesetting of which chart to analyse is very vague, so the software will not have guidance.

When I chose the theme of immigration, my research on this theme is not enough. What are the elements of the migration theme that need to be analysed? I haven't carefully read many essays about the migration topic. Their research will be beneficial for me to determine which direction to design the project.

If I do this project again, I will spend more time thinking about the theme selection, whether this theme contributes to society, whether it has enough value to analyse, and whether it is the theme that users want to study. And after topic selection, I will look for research papers on this topic, make an assessment of their research direction, and work out a meaningful research direction that I can do, which may cause the project more targeted, and enrich the background information of the project, and enable users to get correct conclusions in the desired field.

In terms of time management, I will also reexamine myself. I planned my future working time on the Gantt chart. Still, in reality, I didn't make the same arrangement as expected, which resulted in a considerable reduction in the time I spent writing reports and reading documents. Therefore, if I redo this project, I hope I can do it according to the planned time, and leave more time for reading background literature and writing report.

And for the tools part, I feel like I can use more tools and combine them together. For example, I can use flask to generate a web and combine with plenty of visualisation tools, bokeh is just one of them. More tools to combine together might form a better project.

For the plotting part, the default plot of short-term migration line chart is too complicated, too many lines appear on the figure. It actually affects the ornamental and makes users confused.

8.3 Learn from work

I learned a lot of new experience from this project. Python can refer to many resource packages in the application, and the simplicity of the language makes it easy for novices to get started. For example, combining openpyxl, panda, NumPy and bokeh will be enough to achieve many goals in data analysis. In the future, I hope to expand more possibilities in Python to develop new things.

Bokeh is a beautiful and straightforward visualisation software. I have learned many basic attribute requirements of visualisation drawing. The charts in other tools will require the features of the same attribute to plot a figure as well.

This project also extends my Thinking on data matrix arrangement, how to integrate data to draw dynamic and renewable interactive charts. In the import and extraction of data stage, ”for loop” is very helpful to extract row and column data. It is an excellent method to add data to a list by using the cell function of openpyxl. Bokeh’s ColumnDataSource function gives me a lot of thinking space. It forms the arrangement of data and provides convenience for data transformation and update in the future.

In the interaction design, the callback function of bokeh has a lot of possibilities. The CustomJS method makes JavaScript code have significant advantages in the use of bokeh. The combination of JavaScript and Python makes the processes of interaction design more flexible. Through my interpretation of the source code of the open-source project on the official bokeh website, I learned more about the various writing methods of the interactive update method. Each coder’s understanding of the project, structural layout and writing method are different, so the final update method is also different. Most of their coding functions are entirely different from mine, which is also the place where I need to learn to change. They will not take one graph after another Return. Instead, they wrote a method to generate datasets and plot. This provides more possibilities for future interaction methods. I don’t need too many figures for the scheme and save more code space. This makes me have a deeper understanding and Thinking of logic, writing methods, and project code layout.

I learned the importance of data analysis in the study of migration. The knowledge in statistics will be applied to the analysis of charts. These contents will not be a simple positive correlation, negative correlation and slope. Maybe a simple chart, students who have studied statistics and data science will gain more information.

Finally, I learned that data visualisation is not just a simple ”plot figure”. Behind the plot figure, it covers the philosophy of whether you can grasp the critical point. This key point can be what you need to understand when facing a pile of background data of the same subject. It requires you to have a deep understanding of the issue, and be ready for the research in this direction in the early stage. Therefore, when reading datasets, you should take notes, be prepared to visualise which data you want, whether they are worth plotting out by you, and whether they are worth analysing this chart by users. Therefore, the ability to grab core data in a pile of data is crucial.

On the other hand, whether these data are suitable for display and how they will be displayed are the best, which is also the place that data visualisation chemical authors need to think deeply. There are many kinds of charts, and more robust data visualisation experts can make a variety of beautiful charts combining different charts. I also genuinely feel the research on the characteristics of various charts from this project. What visual impact they bring to users, and what aspects users will pay attention to first, is an essential knowledge I have to learn.

8.4 Future Development/Challenges

There are many possibilities for the future extension of the project because the theme of migration is closely related to many factors.

I can add another six interfaces, which are about income, house price/rent, education level, crime rate, the support rate for Brexit and the relationship between COVID-19 and migration situation or international travelling situation. These factors will be the focus of people' attention.

In the design of plot figure, I will expand more ideas. For example, in the small theme of short term migration of London areas, I can make a stacked vertical bar chart, to have a more considerable improvement in both viewing and research. The application of the map is also feasible. We can use spatial data visualization to show the situation of immigrants in various regions. The colour of areas with more immigrants will be darker, and vice versa.

I will also add more widgets to the design in the future, such as sliders, to observe the time changes of the immigration chart.

Appendix A

Appendix

A.1 Definition

- ”The UK” stands for the United Kingdom.
- Due to the emergence of new types of short-term migration different from tourism in the international migration mode, the United Nations defines another type of migration, namely short-term migration, as ”people who have moved to a non resident country for at least three months and less than 12 months, except those who go to the country for entertainment, vacation, visit relatives and friends, business, medical treatment or religious worship purposes.”
- ”Long-term international migration” refers to ”people who have moved to a non resident country for at least one year, so that the destination country actually becomes its new resident country

A.2 Reading Resources/Data sets resources

The following are the documents I read when I wrote the project, and I also used some of them when I wrote the report.

References

[1] Foundation of Data Visualization. [Online]. https://serialmentor.com/dataviz/introduction.htmlugly-bad-and-wrong-figures

[2] What is visualization really for?.[Online]. https://arxiv.org/abs/1305.5670

[3] Visual Thinking for Design. (Colin Ware).

[4] Information Visualization: Perception for Design. (Colin Ware).

[5] Schneiderman's Mantra. [Online]. Feburuary 26, 2016 / HAMPDATAVISUALIZATION. https://hampdatavisualization.wordpress.com/2016/02/26/schneidermans-mantra/

[6] City Intelligence Data Design Guidelines.[Online]. June 18th, 2019. (Mike Brondbjerg). https://data.london.gov.uk/blog/city-intelligence-data-design-guidelines/

[7] Visual Vocabulary.[Online]. Inspired by the Graphic Continuum by Jon Schwabish and Severino Ribecca. https://ft-interactive.github.io/visual-vocabulary/

[8] MIgraiotn indicators.[Online]. Greater London Authority(GLA). https://data.london.gov.uk/dataset/migration-indicators

[9] Underemployment.[Online] Office for National Statistics (ONS). https://data.london.gov.uk/dataset/underemployment

[10] Migration and the Economy: Economic Realities, Social Impacts and Political Choices.[Online]. Lead author: Professor Ian Goldin, Professor of Globalisation and Development at the University of OxfordMigration and the Economy: Economic Realities, Social Impacts and Political Choices

[11] The Management Of International Migration: Short-Term".[Online]. Author:Jeannette Money,1998

Appendix B

User Guide

B.1 Instructions

1. Install python 2.7
2. Install bokeh(set the path to python 2.7 site-packages)
3. Install pandas(set the path to python 2.7 site-packages)
4. Install numpy(set the path to python 2.7 site-packages)
5. Install openpyxl(set the path to python 2.7 site-packages)
6. Open terminal, run: python -m bokeh serve –show web.py, wait for the respond from browser

Appendix C

Source Code

C.1 Source Code

Listing C.1: ug$_{f}yp_{r}eport_{t}emplates - master/FrontMatter/web.py$

```
'''
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
        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",
"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,

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


```

```python

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
    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
```