Analysis and Visualisation Report

The idea of this visualisation is based on Clark-Fisher theory which is also called the three-sector theory. According to the theory, the main focus of an economy's activity shifts from the primary(agriculture), through the secondary(industry) and finally to the tertiary(services) sector. In other words, this theory assumes that the share of employment in each sector could be used to show that economies move through three stages.

Further, the theory believes that the process of shifting through 3 stages is related with per capita income. Therefore, a higher share of employment in services sector of one country means a higher economic development level.

Based on this assumption, dataset including 'share of employment by sector' was found with the hope that the analysis and visualisation derived from it could justify this assumption and show the process of economics moving through three sector.

• Description of dataset and transformation

1. Origin of dataset

Variables	File name	Description
Share of employment by sector by gender	GET_sector_share.csv From ILO	The original dataset was found from the web set of ILO (International labor Organisation). It includes 'share of employment by sector and gender' for 178 countries & 9 regions from 1991 to 2018 (data before 2013 is real data, data from 2013 is projected).
Output per worker by sector by gender	GET_labourproductivity.csv' From ILO	In order to analysis the correlation between output per worker and share of employment in each sector, the dataset including 'output per capita' with the same data structure was found from ILO.
GDP per Capita	GDP.csv From world bank	This variable is also used for correlation analysis. It might be more suitable to use GDP per capita to represent economic development state, since 'output per worker' from ILO was not given explicit definition.
Country code	disc.csv	Country code was use as intermediate to combine the map data and ILO data. Each region was also assigned to an index.
World map topojson	readme-world.json	

GDP per Capita and Output per worker are both included in the data for analysis process, in order to see whether they could justify each other.

2. Procedure of transformation

For transformation, the main methods used are from python packages panda.

(1) Load dataset

```
prod = pd.read_csv('GET_labourproductivity.csv')
```

```
sector_share = pd.read_csv('GET_sector_share.csv')
```

```
gdp = pd.read_csv('GDP.csv')
```

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

(2) Procedure of combining four csv datasets

```
# merge dataset share of sector with country code disc
sector_share_disc=pd.merge(sector_share, disc, on='Region_country', how='left', sort=False);
```

```
# transfrom GDP dataset with melt method to the sturctrue which could be later merged with other data
gdp_data=pd.melt(gdp, id_vars=['Country_code'], value_vars=["1991","1992","1993","1994","1995","1996","1997","1998",
                                                            "1999","2000","2001","2002","2003","2004","2005","2006",
                                                            "2007","2008","2009","2010","2011","2012","2013","2014",
                                                            "2015","2016"],
        var_name='Year', value_name='GDP_per_Capita')
```

```
# merge sector_share_disc with gdp_data
data_merge=pd.merge(sector_share_disc,gdp_data,on=['Country_code','Year'],how='left');
```

```
#merge data_merge with labor productivity
data_prod_gdp=pd.merge(data_merge,prod,on=['Year','Region_country'])
```

Finally, all four csv origin data set was combined together as dataframe 'data_prod_gdp'. And 'data_prod_gdp' was further used for two purpose: (1) initial analysis (2) further transformation for the convenience of presenting by Js.d3.

(3) Further transformation

The procedure was documented in the ipython file 'coursework_python.ipynb'.

This process is full of methods of melt, concat, merge, drop, rename, and is purely for converting dataframe 'data_prod_gdp' to the expected structures which are easy for the html construction part by Js.d3.

• Data analysis by python

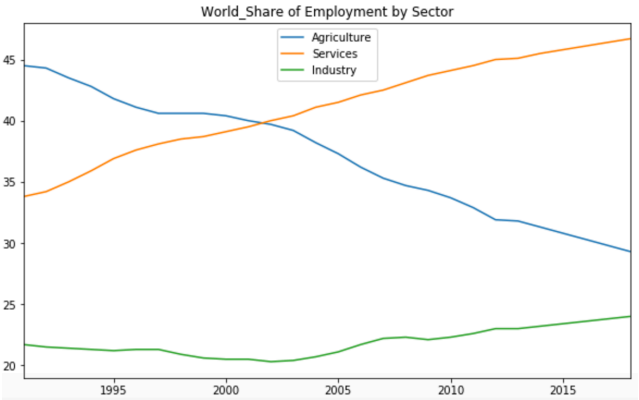
For data analysis, the main methods used are from python packages matplotlib, scipy.stats.

1. Does the data show the process of economic shifting through 3 stages?

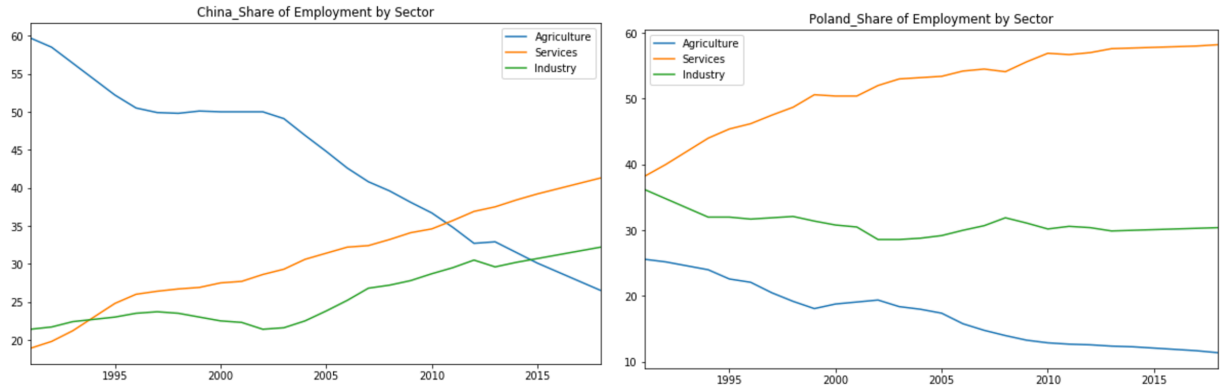
Time plot was used to plot shares of 3 sectors from 1991 to 2018 against time series.

```
ts=data_prod_gdp[data_prod_gdp['Region_country']=='World'][['Year','Agriculture','Services','Industry']]
years=ts['Year'].astype(int).tolist()
ts=ts.set_index([years])
ts.plot(figsize=(10,6),title='World_Share of Employment by Sector')
plt.show()
```

The plot for the world shows as the share of employment in Agricultures decreases, the other two increase.



The same figure also plotted for China and Poland. The two time plots show the two countries are at the different stage of the shifting process. China is in the secondary stage, and Poland is in the tertiary stage.



2. Is there any difference between female and male?

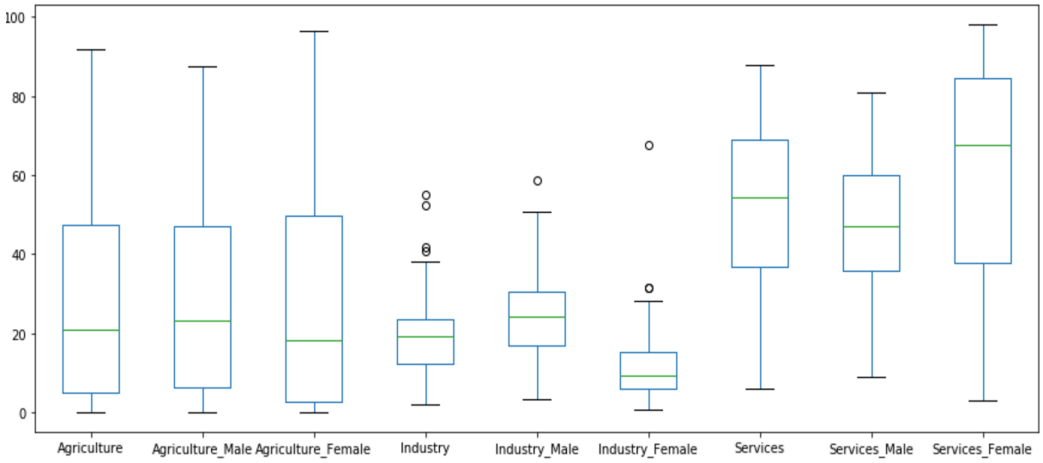
(1) Boxplot

The boxplot showing the distribution of share of employment by sector and gender for 178 countries in 2012.

```
data_prod_gdp_2012=data_prod_gdp[(data_prod_gdp['Year']=='2012')&(data_prod_gdp['Region_type']=='country')]
.drop(['GDP_per_Capita','Output_per_worker','Country_id','Region_id'],axis=1)
```

```
data_prod_gdp_2012.plot.box( figsize=(15, 6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1082724a8>
```



The boxplot below shows the median of shares of employment in agriculture, industry, services are around 20%, 20%, 54% respectively. The distance between lower and upper quartiles are huge for agriculture and services. Also, the share of employment for female tends to be lower than male in industry sector, and tends to be higher in services sector.

(2) T-test

The previous finding could be justified by a group of T-test across different countries within the 9 regions. The null hypothesis for the T-test is the mean values for men and women are not significant different.

(2.1) Checking normality for male and female groups for all countries.

```
print ("Result of checking normality")
for i in regions:
    print (i+": "+"\r")
    print("    Female: "+str(stats.shapiro(data_prod_gdp[(data_prod_gdp['Region']== i)
                                &(data_prod_gdp['Region_type']== 'country')]['Services_Female'])));
    print("    Male:   "+str(stats.shapiro(data_prod_gdp[(data_prod_gdp['Region']== i)
                                &(data_prod_gdp['Region_type']== 'country')]['Services_Male'])));
```

The result of normality test shows normality.

(2.2) Performing T-test between male and female groups for all countries.

```
print ("T-test:")
for i in regions:
    print (i+": "+"\r")
    print ("      "+str(stats.ttest_ind(data_prod_gdp[(data_prod_gdp['Region']== i)
                                &(data_prod_gdp['Region_type']== 'country')]['Services_Female'],
                        data_prod_gdp[(data_prod_gdp['Region']== i)
                                &(data_prod_gdp['Region_type']== 'country')]['Services_Male']))
```

The result shows the values for female is higher than male for 8 regions except south Asia, but the results for 'Developed Economies & European Union' and 'East Asia' are not significant, since p-value are above 0.05.

The finding includes:

(1) The shares of employment in agriculture and services sector both have huge between differences countries, the development stage for different countries are quite diversified.

(2) The proportion of women working in services sector is higher than men, and the proportion of men working in industry sector is higher than women.

(3) A higher proportion of women tend to work in services sector.

3. Whether the shifting process through 3 sectors is correlated with an economy's development state?

3.1 Use of indicator

Share of employment in services sector is used as indicator of the stage of the shifting process;
GDP per Capita/Output per worker is used as indicator of the economic development state.

3.2 Hypothesis of Pearson correlation

The null hypothesis is there is no correlation between the two factors (r=0). Then if the test result is not significant (p-value < 0.05), we can reject the null hypothesis, and conclude that they are correlated.

3.3 Perspectives of detecting their correlation

We could interpret the correlation between the two factors from two perspectives:

1) For the same time period, countries or regions having higher share in services sector tend to have higher GDP per capita.

2) For the same country or region, as share in services sector increases, GDP per capita also increase.

3.4 Checking the hypothesis of correlation for the same period across different countries

Checking the assumption that **countries or regions having higher share in services sector** tend to have **higher GDP per capita.**

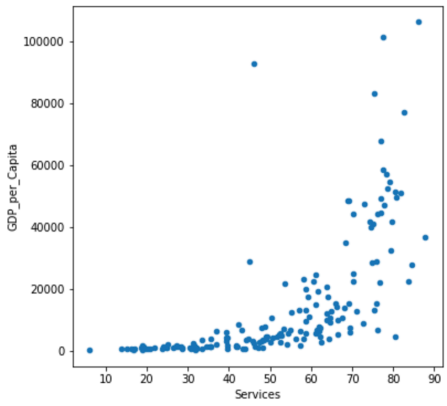
(1) Scatterplot between share of employment in services and GDP per capita for each country in 2012

```
data_prod_gdp[(data_prod_gdp['Year']=='2012')].plot.scatter(x='Services',y='GDP_per_Capita', figsize=(6, 6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x109d753c8>
```

```
plt.show()
```

According to the scatterplot below, it seems there is a slightly positive correlation between the two factors. Countries which have a higher share in services sector, tend to have higher GDP per capita.



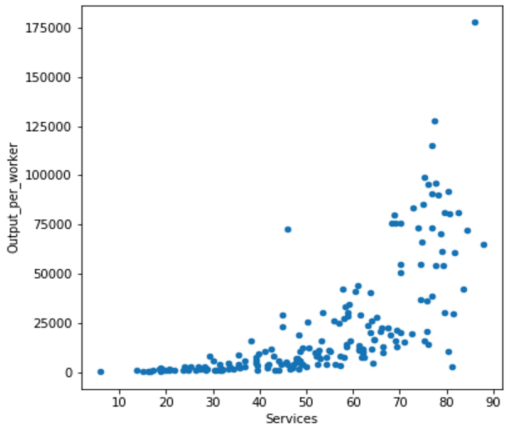
(2) Scatterplot between share in services sector and output per worker for each country in 2012

```
data_prod_gdp[(data_prod_gdp['Year']=='2012')].plot.scatter(x='Services',y='Output_per_worker', figsize=(6, 6))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x109d8c908>
```

```
plt.show()
```

The plot shows the same result as the previous one.



(3) Test Pearson's correlation between share in services sector and output per worker

The results are all significant for 9 regions. And the Pearson R of East Asia is the highest which means a higher correlation, and the Pearson R of Middle East is the lowest.

```
print ("Pearson Correlation between share of services sector and output per worker for all countries: ")
for i in regions:
    print (i+"\t" + str(stats.stats.pearsonr(
    data_prod_gdp[(data_prod_gdp['Region']== i)&(data_prod_gdp['Region_type']== 'country')]['Services'],
    data_prod_gdp[(data_prod_gdp['Region']== i)&(data_prod_gdp['Region_type']== 'country')]['Output_per_worker']
    )));
```

```
Pearson Correlation between share of services sector and output per worker for all countries:
Developed Economies & European Union    (0.72709804916185017, 4.3744914877881772e-171)
Central & South Eastern Europe (non-EU) & CIS   (0.43833623834977992, 4.4417825729078361e-25)
East Asia       (0.90578688547842268, 2.8574744616006861e-74)
South East Asia & the Pacific   (0.86007561877736194, 4.7740048968503959e-116)
South Asia      (0.8643894261510191, 3.1993514623810591e-68)
Latin America & the Caribbean   (0.73138182586336498, 1.8617936879778915e-141)
Middle East     (0.31692346049709474, 6.175103174362043e-10)
North Africa    (0.8663155580403934, 6.4442735059027228e-52)
Sub-Saharan Africa      (0.64299513812556197, 6.4649519761867419e-148)
```

(4) Test Pearson's correlation between share in services sector and GDP per capita

The result is not significant only for Middle East. For the remaining regions, the result is significant and the Pearson R values for each region are similar with results of the previous test.

```
print ("Pearson Correlation between share of services sector and GDP per capita for all countries: ")
for i in regions:
    print (i + ": "+ "\t" + str(stats.stats.pearsonr(
    data_prod_gdp_1[(data_prod_gdp_1['Region']== i)&(data_prod_gdp_1['Region_type']== 'country')
                    ]['Services'],
    data_prod_gdp_1[(data_prod_gdp_1['Region']== i)&(data_prod_gdp_1['Region_type']== 'country')
                    ]['GDP_per_Capita']
    )));
```

```
Pearson Correlation between share of services sector and GDP per capita for all countries:
Developed Economies & European Union:   (0.73173769611649386, 1.692622590165951e-152)
Central & South Eastern Europe (non-EU) & CIS:  (0.52160824158566887, 2.2515326382782326e-31)
East Asia:      (0.76694235937711452, 1.8711753973806137e-25)
South East Asia & the Pacific:  (0.77570925261873436, 2.4194773328622941e-67)
South Asia:     (0.71700985594583511, 2.7654722228803568e-31)
Latin America & the Caribbean:  (0.75613506817498533, 6.3747084463079511e-130)
Middle East:    (0.064862349715926512, 0.26678904304359508)
North Africa:   (0.76793225352498617, 1.1924089656134695e-29)
Sub-Saharan Africa:     (0.53312673564969459, 9.2263513316918716e-80)
```

3.5 Checking the hypothesis of correlation for the same countries across 28 years

In addition, in order to detect whether 'GDP per capita increase as the share in services sector increases' for each country, the hypothesis of Pearson correlation was also check for each countries using code below.

```
#create list for all countries which have valid value for GDP_per_Capita
#data_prod_gdp_1 is the df which excludes NA
countries=data_prod_gdp_1[data_prod_gdp_1['Region_type']== 'country']['Region_country'].drop_duplicates()
```

```
#create a list to take record of the result of Pearson correlation for all contries.
result_correlation=[];
for i in countries:
    item=[]
    item.append(i)
    item.append(stats.stats.pearsonr(
    data_prod_gdp_1[(data_prod_gdp_1['Region_country']== i)]['Services'],
    data_prod_gdp_1[(data_prod_gdp_1['Region_country']== i)]['GDP_per_Capita']
    ))
    result_correlation.append(item)
```

```
#print out countries in which two factors have negative correlatlion or results are not siginificant
for i in result_correlation:
    if not ((i[1][0]) >0 and (i[1][0]) > 0.05):
        print (i)
```

The result shows there is only 19 countries having negative correlation between the two factors, which means as share in services sector increases, GDP per capita also decrease.

For all the other countries, the two factors we tested have positive correlation, and the corresponding results are significant.

```
['Bosnia and Herzegovina', (-0.82455502701483951, 2.3652440062225532e-06)]
['Georgia', (-0.33622870666734583, 0.10031626879505239)]
['Sri Lanka', (-0.15575920247615105, 0.45719090066447088)]
['Suriname', (-0.80847702790028897, 1.0104914237888825e-06)]
['Trinidad and Tobago', (-0.26982682419138759, 0.19209190893429834)]
['Uruguay', (-0.72640960056804149, 3.9314165646607687e-05)]
['Bahrain', (-0.41015420385632612, 0.041712723336867721)]
['Kuwait', (-0.48095272800967831, 0.02345461162535778)]
['Oman', (-0.92371951990554002, 4.6053145853612599e-11)]
['Qatar', (-0.91745057527089136, 1.106882489970466e-10)]
['Tunisia', (0.018029982881422976, 0.93183110876138719)]
['Botswana', (-0.94026160188679897, 3.0072314732033321e-12)]
['Central African Republic', (-0.0023818846654299298, 0.99098435140023156)]
['Comoros', (-0.62235074943684698, 0.00089372650941170473)]
['Eritrea', (-0.33610454115449007, 0.14737460934461258)]
['Gambia', (-0.77452151870879138, 5.5146209865297719e-06)]
['Ghana', (-0.61654986575818882, 0.0010300225414538709)]
['Zambia', (-0.8822507488425978, 5.5049183011055491e-09)]
['Zimbabwe', (-0.076271895001828252, 0.71708016404491271)]
```

• Main procedures of Implementation

In this part, only procedures which is worth pointing out will be mentioned, such as problem fixed or still not fixed, or some useful methods for future.

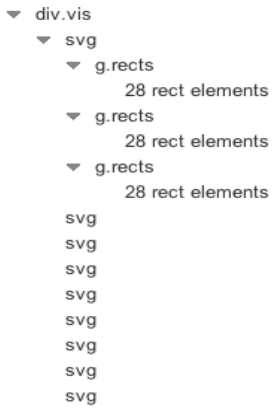
1. Small-multiple stacked bar chart

For showing the process of economies move through three stages, stacked bar chart was used. Since we have found that the situations are different for different countries/regions, we want to present the same stacked bar chart for each region, then a small-multiple stacked bar chart was used.

(1) Transforming the raw data to a efficient structure

Since Small-multiple chart has a hierarchical structure, and stacked bar chart also needs to stack data in to different groups. Therefore, we need a 2 layers nested data in this case.

(1.1) The structure of html elements we want to build.



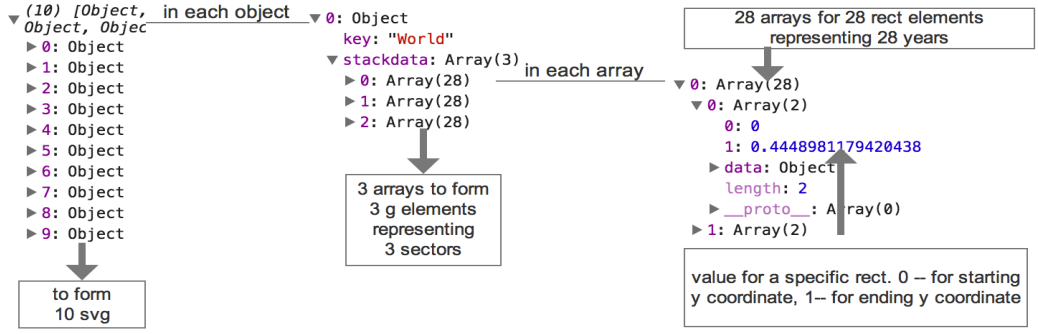
We need 10 'svg' since we need to build 10 stacked barchart for 10 regions. In each 'svg', as we have 3 sectors, 3 'g' elements are needed with each 'g' to represent one sector. And in each 'g' element, as there is 28 years, 28 'rect' elements are needed with each one representing values for a specific year.

(1.2) Method of transforming raw data to expected data structure

Methods used contain d3.nest() and d3.stack().

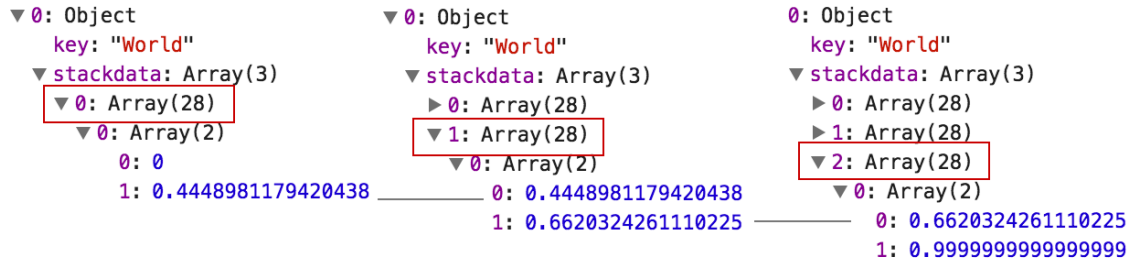
```
//////////////////// transform data ////////////////////////////////
//Data is nested by region
var regions_data = d3.nest()
    .key(function(d) { return d.Region; })
    .entries(CHRdata);
//with in each region, data is grouped into 3 array(sectors), and each array has 28 sub-array.
regions_data.forEach(function(s,i) {
    //console.log(d);
    var stack = d3.stack()
        .offset(d3.stackOffsetExpand);
    s.stackdata = stack.keys(Sector_keys)(s.values);
});
```

(1.3) The resulting structure of data



(1.4) The interesting property of the d3.stack() method.

The interesting and useful part of d3.stack() method:



d.stack[0][1] ==d.stack[1][0],
d.stack[1][1] ==d.stack[2][0],
d.stack[0][1] - d.stack[0][0] is the value for first sector,
d.stack[1][1] - d.stack[1][0] is the value for second sector,
d.stack[1][1] - d.stack[1][0] is the value for third sector.
This property is just useful to make a stacked bar chart.

(2) Problem I got stuck

(2.1) Updating data using 'onclick' event

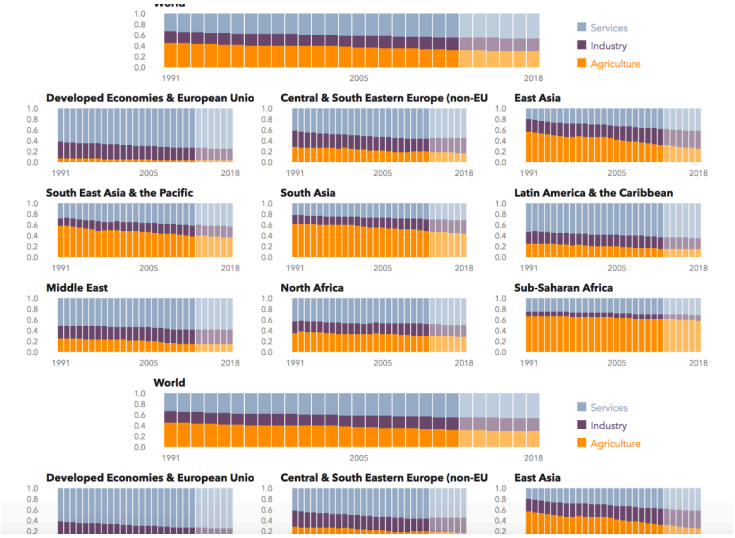
This problem appeared when trying to use the 3 buttons (Total/Male/Female) to update attributes of chart.

The expected result:

The expected result is the length of each bar which represents value for a specific sector in a specific year will be updated as new data is rebound to html elements.

Description of the problem:

When button was clicked, another 10 new 'svg' were created with new chart drawn in each 'svg'. The old 10 'svg' were still existed and attributes of 'rect' elements were not changed at all. The graph below shows the situation.



The reason for this problem:

In the initial version of js code, the regions_data.**forEach()** method was used as a loop to append one 'svg' to 'div' with in each loop. Therefore, 10 new 'svg' were created when the draw() function was called each time.

```
regions_data.forEach(function(d,i) {
    //console.log(d);
    var stack = d3.stack()
        .offset(d3.stackOffsetExpand);
    var stackdata = stack.keys(keys)(d.values);

    xScale
        .rangeRound([0, nScale*width]).paddingInner(0.1)
        .domain(data.map(function(d) { return d.Year; }));

    var CHRsvg = d3.select("div.smallmultiple")
        .append("svg")
        .attr("class", "barchart")
        //.attr("position", "absolute")
        .attr("height", height + margin.top + margin.bottom)
```

The first trial for fixing this problem:

Another div ('div.smallmultiple') was built to be positioned between the 'div.vis' and the 10 'svg' elements. And the 'div.smallmultiple' would be remove at first when draw() function was called each, so all the elements contained in 'div.smallmultiple' including the 10 'svg' would be removed. After that, 'div. smallmultiple' would be rebuild, so new 10 'svg' would then be built.

```
function drawbarchart(category) {

    d3.select(".smallmultiple").remove();

    var chartdiv = d3.select("#vis")
        .append("div")
        .attr("class", "smallmultiple");

    //filtering data
    data = bar_data.filter(function(d,i) {
        return d.Gender === category;
    });
```

Problem of this solution is: it needs to redraw everything including the elements which do not need to be updated.

The second trial for fixing this problem:

Instead of building a specific 'svg' within each loop of the forEach() method, selectAll('svg') was used before binding nested data(regions_data). Therefore, the first layer of regions_data (10 keys) was bound to 10 'svg' (10 'svg' were created).

```
var CHRsvg = d3.select("#barchart")
    .selectAll("svg")
    .data(regions_data)
    .enter()
    .append("svg")
    .attr("class", "barchart")
    .attr("height", CHRheight + CHRmargin.top + CHRmargin.bottom)
    //.attr("width", width + margin.left + margin.right)
    .attr("width", function(d,i) { return i==0 ? 3*(CHRwidth+CHRmargin.left+CHRm
    .append("g")
    .attr("transform", function(d,i) { return i==0 ? "translate(" + 8*CHRmargin.
    :"translate(" + CHRmargin.left + "," + CHRmargin.top + ")" })
```

Then, the second layer of regions_data (3 array in d.stackdata) was bound to 3 'g.rects' (at the same time creating 3 'g.rects').

```
var rects = CHRsvg
    .selectAll('g.rects')
    .data(function(d) {return d.stackdata})
    .enter()
    .append("g") // 3 'g.rects' was created
    .attr("class","rects");
```

At last, values nested in the third layer of regions_data was used to set attribute of the 'rect' elements.

```
rects
    .append("g") //used as container for all 28 'rect' elements
    .attr("class","rect")
    .attr("fill", function(d) {return zScale(d.key);})
    .selectAll('rect.rect')
    .data(function(d) { return d;})
    .enter()
    .append("rect")
    .attr("class","rect")
    .attr("x", function(d) {return  d.data["Region"] == "World" ? 1.5*xScale(d.data.Year
```

Problem of this solution is: Although it stopped to append new 10 'svg', the attributes of 'rect' elements were not changed at all. There was no change of the chart at all when button was clicked each time.

So, a new problem was raised.

The reason behind this might be there is some intermediate layer ('g' elements) which do not hold data at all between 'svg' elements and 'rect' elements. The existing of these elements were not triggered by data. So when regions_data was changed, new values can't be passed to the lowest layer - the 'rect' elements.

The third trial for fixing this problem:

The final solution is building an update() function, so that we could select the html elements of each layer which actually hold data, and then we would rebind new data to each of these layers in order to achieve an update. The problem was solved.

```
function update(category){
    //filtering data
    CHRdata = bar_data.filter(function(d,i) {
        return d.Gender === category;
    });

    //Data is nested by region
    var regions_data = d3.nest()
        .key(function(d) { return d.Region; })
        .entries(CHRdata);

    console.log(regions_data);

    regions_data.forEach(function(s,i) {
        //console.log(d);
        var stack = d3.stack()
            .offset(d3.stackOffsetExpand);
        s.stackdata = stack.keys(Sector_keys)(s.values);
        console.log(s.stackdata)
    });

    var CHRsvg = d3.select("#barchart").selectAll("svg.barchart").data(regions_data);

    var rects = CHRsvg.selectAll("g.rect")
        .data(function(d) {return d.stackdata});

    rects.selectAll("rect")
        .data(function(d) { return d;})
        .attr("y", function(d) { return yScale(d[1]);})
        .attr("height", function(d) { return yScale(d[0]) - yScale(d[1]);})
};
```

2. Scatterplot

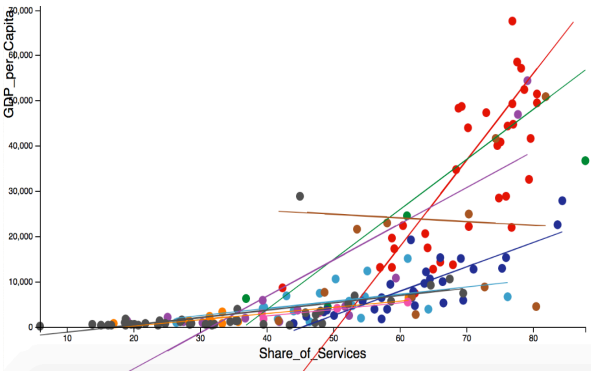
(1) Fitting a regression line for dots

Function for regression:

In order to show correlation between the two factors (Share of employment and GDP per capita), a function was used to fit a linear regression line for those dots. The purpose of this function is transforming the original coordinates (x, y values) of dots to the coordinates of the regression line.

Problem:

The coordinates of the fitted line produced by the regression function would be out of the y scales which were set earlier. Therefore, we could see there is some parts of lines under the x axis. The graph below shows the situation.



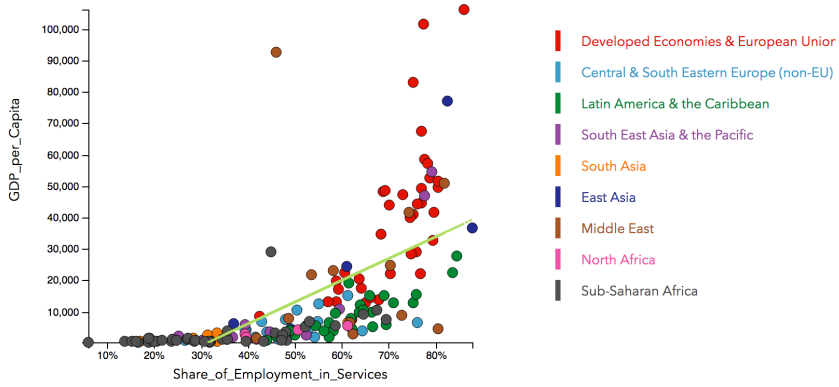
Solution:

The solution is quite straightforward. It just avoids pushing new coordinates which are out of y scale to the resulting array, since we do not need that much coordinates to draw a line.

```
for (var v = 0; v < values_length; v++) {
    x = values_x[v];
    y = x * m + b;
    if( y >= minYdomain && y <= maxYdomain
    ){
        result_values.push({x,y})
    }
}
```

(2) Interaction for selecting regions

For user to change region presented by the scatterplot, interactive legends are used.



But in the procedure of implementation, there were some issues which have been considered.

a. Whether to use Mouseover or Click on legend for interaction:

If mouseover event is used, then it is impossible for user to mouseover on the dots to see detail when they have already mouseovered on the name of region.

So, Click event was chosen.

b. Whether to hide dots from region not chosen or to filter original data to achieve the interaction:

Hiding dots from region not chosen:

This way is straightforward to achieve by changing opacity.

The problem is, after user clicking on the legend of a selected region, dots from region not chosen would be hidden. But at this time user might tend to mouseover on a specific dot to see detail

information, then the awkward scene appeared since the mouseout event of the dot itself would make all dots visible including those from regions not chosen.

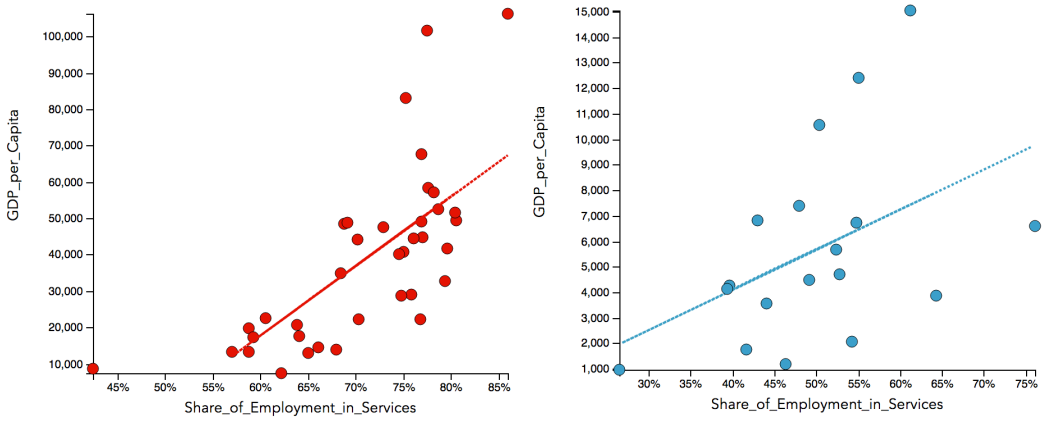
Filtering original data:

Filtering original data could avoid the problem mentioned previously. So it is used.

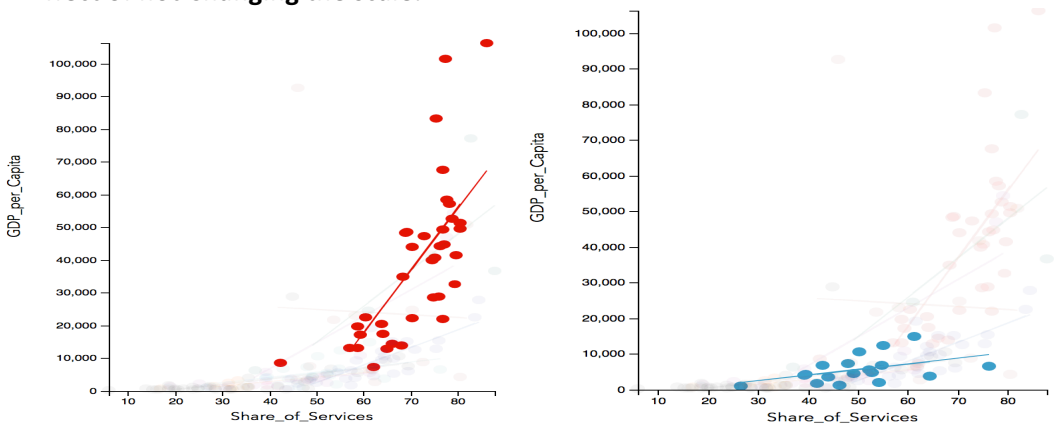
But it also has drawback, it means every time when user clicks the on the legend, we need to rebind data to 'svg', and redraw the whole chart.

Whether to change x, y scale of coordinate according filtered data ---- Scaling of axis

Effect of Changing the scale:



Effect of not Changing the scale:



It seems changing scale would be better, there is two reasons for that:

a. It makes much use of the coordinate, otherwise dots would be distributed at bottom the chart for region with low GDP.

b. Another factor to be consider is the slope of regression line. If we don't change x, y scale accordingly, the difference between the slope of regression line for different regions would be show obviously. Then the real question is **'Does higher slope mean higher correlation?' The answer is no, since it is the standardised slope identical to the correlation r.**

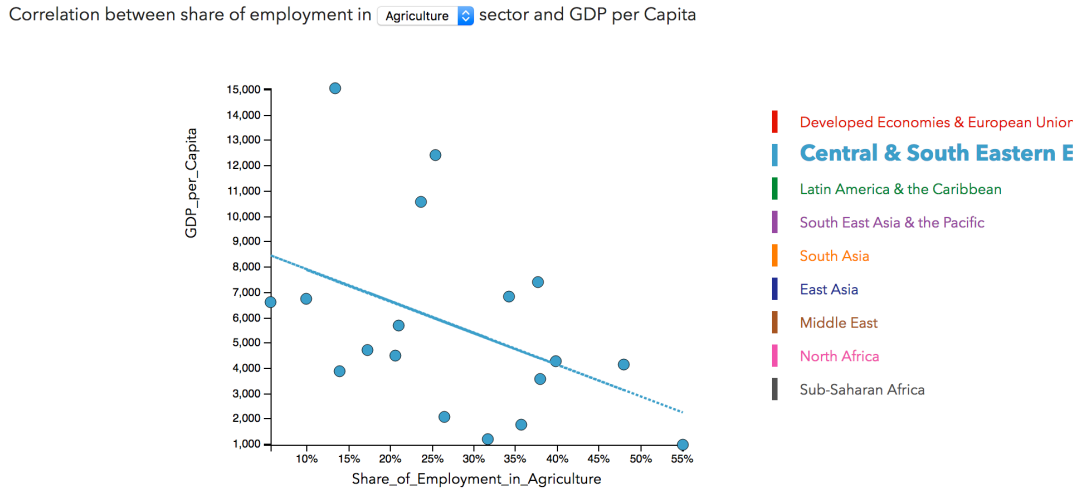
Therefore, we should make the purpose of drawing the regression line explicit:

The purpose of showing the regression line is just to indicate there is correlation between the two factors, and also whether the correlation is positive or negative. But the slope doesn't indicate the strength of their correlation.

Hence, we don't want to make the difference between the slope obvious, then we need to change scale of coordinate accordingly.

(3) Interaction for selecting sector

For user to change sector presented by the scatterplot, interactive legends are used.

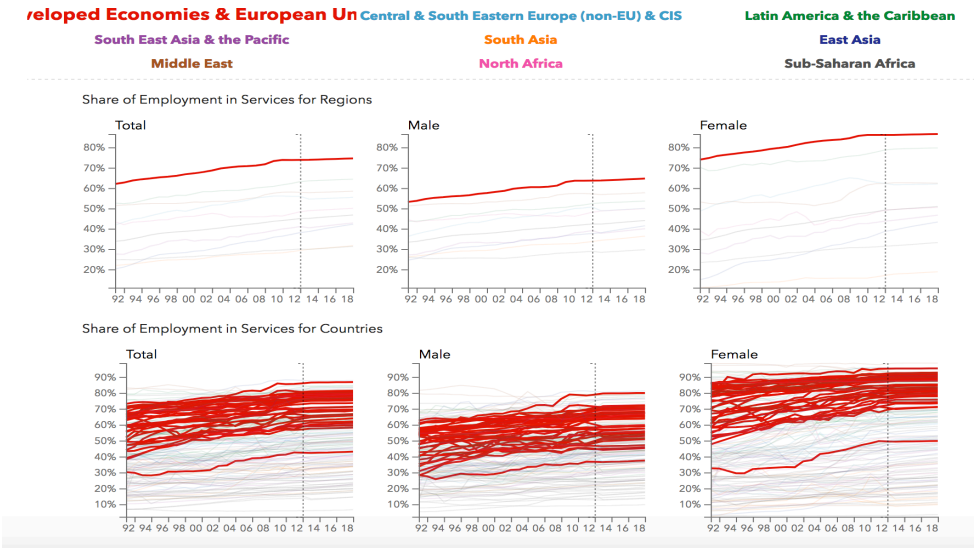


3. Small-multiple Linecharts

The purpose of the country linecharts and region linecharts is to compare the share of employment in services sector between male and female. Since small-multiple is used to draw 3 linecharts both for region and country section, so the data used here are also 2 layers nested data.

(1) Dynamic assigned id or class attribute of the convenience of selection

In this part of visualisation, the mouseover event are also used by interactive legends. But the different part is that, the elements needed to be active are not only the legend itself, but also the lines for the corresponding region and corresponding countries with in this region.



Therefore, we should assign a dynamic id (e.g. d.key) to each html element, so that we could select the specific elements, instead of using d3.select(this).

Assigning dynamic id to html element that we need to use in later selection.

```
countryLines
  .append("path")
  .attr("class", function(d) {return ("line_region_"+ RegionIdByCountryId[d.key]); })
  .attr("id", function(d) {return ("tag_country_"+ d.key); })
  .attr("d", function(d) { return LNline(d.values);})
  .style("stroke", function(d) { return LNcolorScale(RegionNameByCountryId[d.key]); })
  .attr('fill', 'none')
  .attr("opacity", 0.1)
  .on("mouseover",LNmouseOver)
  .on("mousemove",LNmouseMove)
  .on("mouseout",LNmouseOut);
```

Then in mouseover() function, we could select html elements by “id” or “class” to control their attributes.

```
function LGmouseover(d) {
  //console.log(d)
  d3.selectAll("g.country").selectAll("path")
    .transition()
    .duration(500)
    .style("opacity",0.1);
  d3.selectAll("path.line_region_"+RegionIdByRegion[d])
    .transition()
    .duration(500)
    .style("opacity",1)
    .style("stroke-width", 2);

  d3.selectAll("g.region").selectAll("path")
    .transition()
    .duration(500)
    .style("opacity",0.1);
  d3.selectAll("#tag_region_"+RegionIdByRegion[d])
    .transition()
    .duration(500)
    .style("opacity",1)
    .style("stroke-width", 2);

  d3.select(this)
    .transition()
    .duration(500)
    .style("font-size", "18px" );
};
```

(2) About code reuse

The codes for region linechart and country linechart have a high proportion of repetition, since they draw small-multiple linechart respectively. It is possible to reuse the section of define width/height/Scale/axis. It might be possible to combine and reuse codes from both parts by creating a function, but there are some parts of them which are different with each other:

a. They both use two layers nested data, but the key used for nested are not the same.
b. They have different y-scale.
c. Their tooltips show different contents.
d. They have different ways of assigning id and class, for different ways of later selection.

I have already combined the part of defining width/height/Scale/axis, but haven’t look in to combine the other parts.

4. World map

When slider is moved, updating attributes necessary instead of redrawing

In the world map section, there is a slider for year selection. When the year selected changes, the colour of the map and the content of tooltip will change accordingly. As the geographical shape of the map does not need to be changed, then it is possible to only update the color of map and content of tooltip.

So, an updatamap() function was built which only change ‘fill’ attributes of path, and redefine content of tooltip within mouseover() function.

In this way, when the slider is move, the topojson data doesn't need to be rebound to html elements, and the geographical shape doesn't need to be redrawn.

5. Parallel coordinates --- something tried but got problem

'Parallel coordinates' is a way of visualizing high-dimensional data. A set of points in an n-dimensional Cartesian coordinate system, would be transformed to n parallel lines, typically vertical and equally spaced. A point in n-dimensional space is represented as a polyline with vertices on the parallel axes.

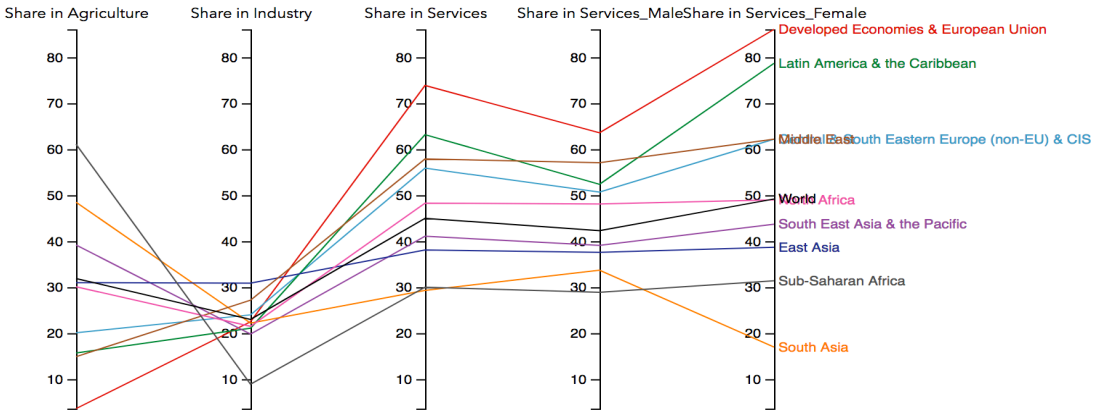
'Parallel coordinates' was also tried for our data. Unlike the scatter plot which could only 3 attributes for each country (3 dimensional data) with x, y coordinates, and colour, 'Parallel coordinates' could show high-dimensional data (as much as you want as it is sensible).

So, instead of using a point to represent a country/region in Cartesian coordinate system, we use a polyline to represent a country/region in 'Parallel coordinates'.

(1) What has achieved

The graph below is what has achieved so far (also see parcoords.html). It shows 5 attributes for each region (although some of them are not independent of each other). According to the chart, we could see, in most of the region Female has higher share in services sector.

This 5 dimensional 'Parallel coordinates' was implemented using the same method as drawing a line chart, although there is also a package in d3 for 'Parallel coordinates'.

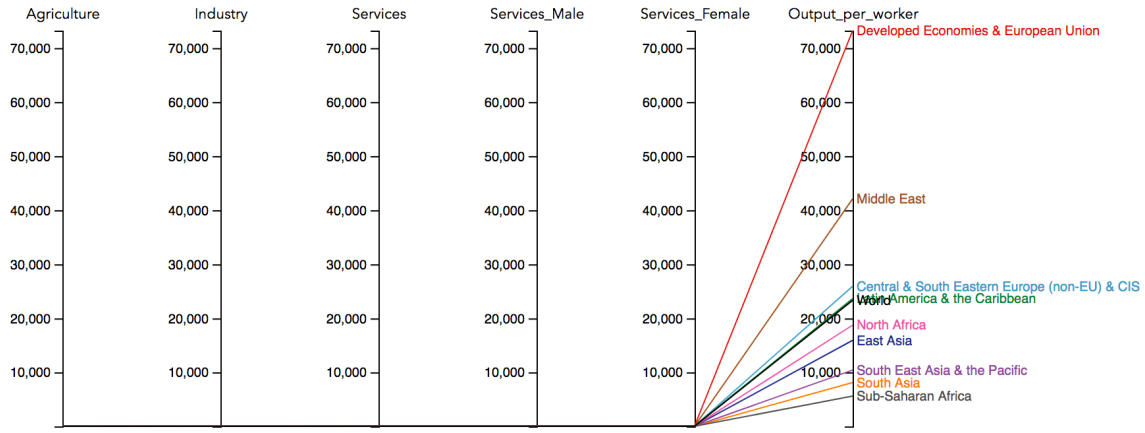


(2) Problems

a. The scaling of each dimension

A real 'Parallel coordinates' should set the scale of each dimension according to the range of each attribute, and appropriate scaling methods can reveal more informative views. But in the chart above, the scale of each dimension is same with each other, and is not set according to the range of each attribute, because it is implemented as drawing a line chart.

Since the 5 dimensions showed in the previous chart are all percentage, the problem seems not obvious. But after a new dimension which has range from 5000 to 70000 was added, the problem became unbearable since all dimension except output_per_worker became unseen.



b. Brushing effect

The two 'Parallel coordinates' chart above can't achieve brushing for selecting and highlighting a subset of the data.

If we want to draw a 'Parallel coordinates' for all 178 countries in our data set, then there would be 178 polylines. In this case, the brushing effect would be imperative.

So, the 'Parallel coordinates' section should be developed further.

• Evaluation of the visualisation work

1. Small-multiple stacked barchart

Merits:

a. Percentage stacked barchart
It is typical in describing the process of economic moving through three stage. It could show the process that as share of employments in Agriculture decreases, the other two increase.

b. Small-multiple
Small-multiple gives user a general view for the situation of all regions, also enables user to compare between them.

c. Buttons for shifting between genders
This is meant to provide visualisation for both male and female. Therefore, user could compare between genders.

Drawbacks:

a. Buttons for shifting between genders
It might be quite annoying to click the buttons continually to compare between genders.
Also, since there is no change of the style of button which has been clicked, it is difficult to distinguish whether the current visualisation is for Male or Total or Female.

2. Scatterplot

Merits:

For Scatterplot:
It is for describing correlation between to factor.
For regression line:

Fitting a regression line for those dots was meant to disclose correlation between x, y coordinates. This regression line could support user to understand the relationship between the two factor.

Drawbacks:

For Scatterplot:

This scatterplot only shows relationship between two factors.

But in our case, not only share in Services matter to GDP per capita, but the other two factors also matter. Share in all three sectors all have influences in GDP per capita.

For example, in East Asia, it seems the share in industry is negative correlated with GDP per capita, which means as share in industry decreases, GDP per capita increases. The reason behind this is the increase in GDP per capita is caused by increase in share in services.

Therefore, it is also meaningful to show relationship between more factors. This is why 'Parallel coordinates' needed to be considered.

For regression line:

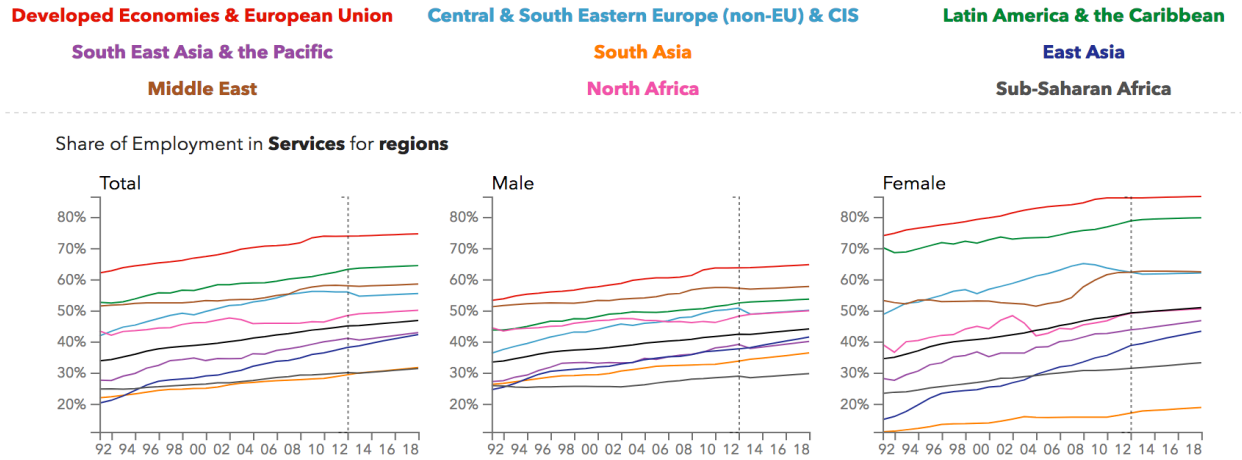
The regression line might be misleading, since user might use the slope of the line to interpret correlation between two factors. But actually, we should bear in mind, the slope of regression line is only used indicate the correlation is positive or negative. It is only equals to correlation r coefficient when both x and y coordinates are standardised.

3. Small multiple linechart

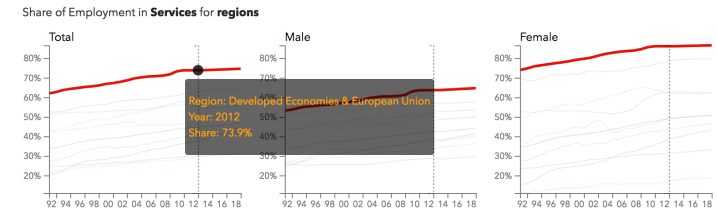
The country linecharts and region linecharts are supposed to show that higher proportion of women tend to work in services sector, they experienced a higher speed of shift from another two sectors to Services sector compared to men.

Merits:

The region linechars section fulfills most of the purposes of presenting it, since it gives answer to some questions explicitly: Do all regions have difference between share in services for male and female? How many the difference is? How are the shares in services for all regions distributed?



Mouseover event highlights related lines for a specific region.



Drawbacks:

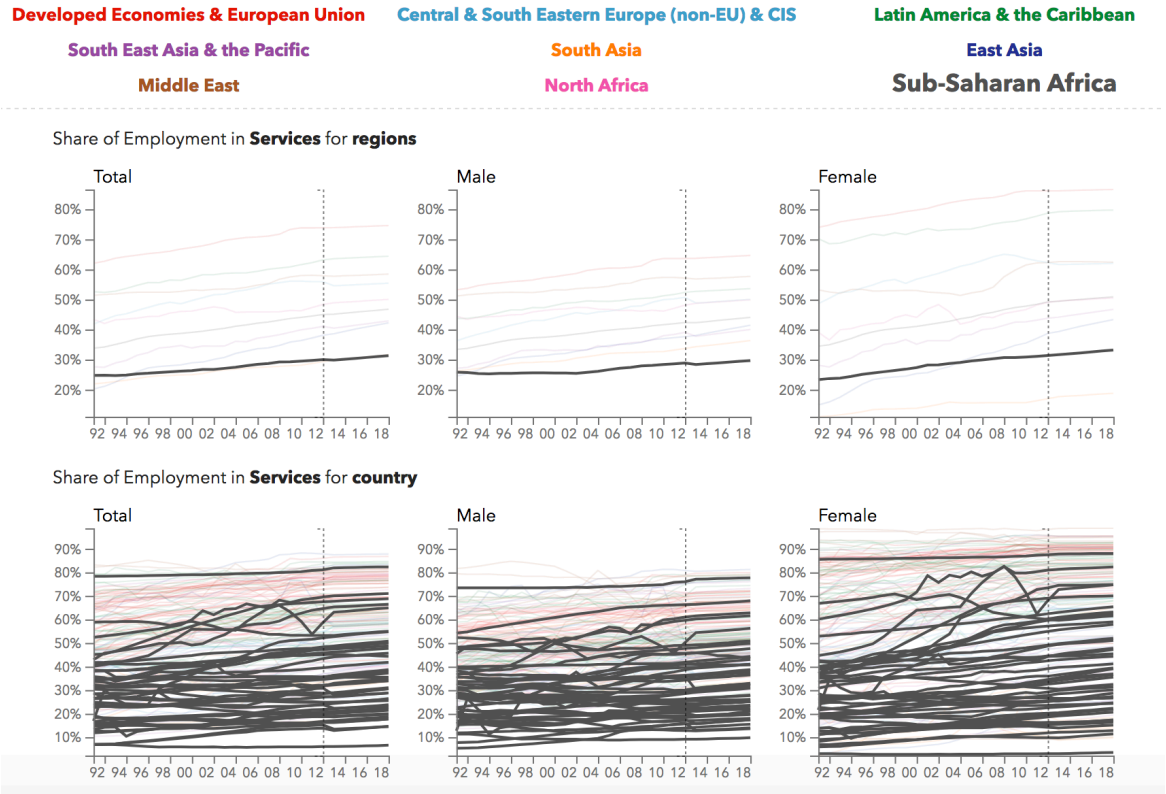
It shows something interesting, but it can't explain itself, since there are some external factors which are not included.

For example, from the chart for regions below, we could find 'Sub-Saharan Africa' has the lowest share in services sector. But in the chart for countries, there are a lot of countries have much higher share in services sector.

It is so interesting, in 'Sub-Saharan Africa', the shares in services sector for all countries in that region are distributed in wider range than other regions, while the share in services for the region itself is the lowest. This situation indicates economic development in 'Sub-Saharan Africa' are quite unbalanced.

The reason behind this situation might be the countries have higher share in services, also have low volume of employment. But the sad thing is the chart itself can't give the reason explicitly.

Further, if we are interested in what those countries are, the awkward thing is the charts for countries doesn't show name of country since it is impossible to show names for countries among so many lines.



4. The map section

Merits:

The world map gives a general view about the distribution of share in service around the world.It is easy to detect how unbalanced the situation is.

It also shows a moving forward economic development process, since the blue colour becomes darker when you move the slider.

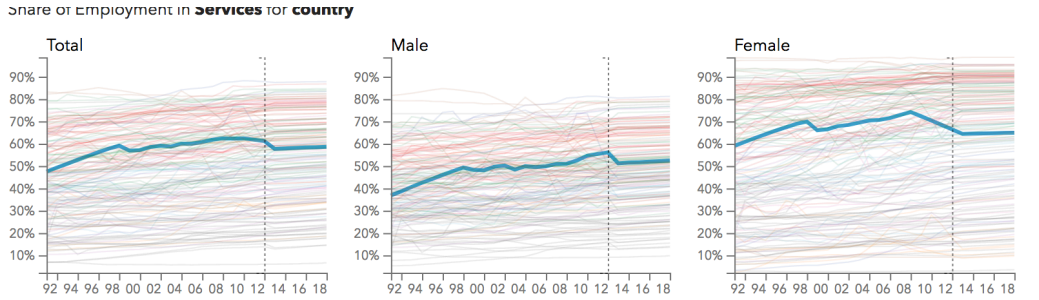
At the same time, by pointing out a specify country, it shows detail for a specific country.

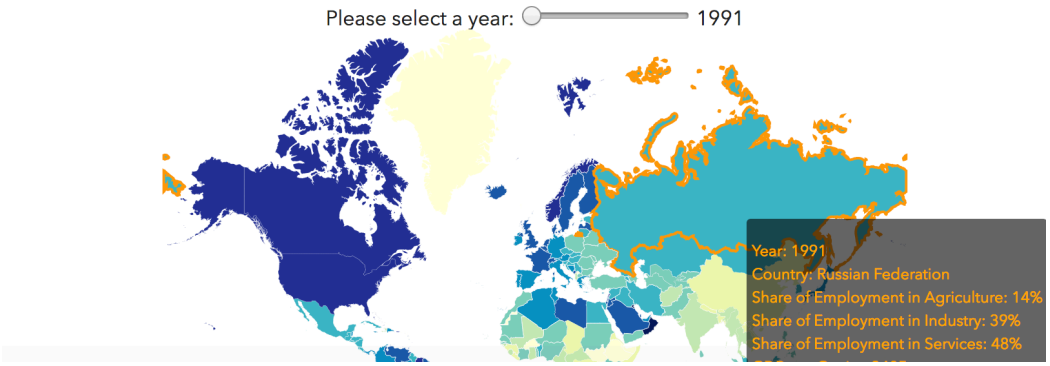
Drawbacks:

When mouseovering a specific countries, the tooltip shows. But it might be better to show the situation for this country along the time line.

In the graph below, when mouseovering on Russia, the related lines should be showed.

But there is a webpage layout problem, because it is impossible to place both the linechart for countries and the map within on screen. Therefore, for some countries (e.g. Australia) positioned in the bottom of map, the related lines would be unseen.





5. 'Explain a lot' or 'Let the user explore themselves'

The web page of this visualisation work is full of words. This is a problem. No explanation might make users get lost, too many explanations might leave no room for users to explore by themselves.

A good visualisation work should be able to explain itself.

• Conclusion

Overall, each section of this visualisation work fulfill the purpose of designing it and presenting it. It has high density of data ink, also provide interactive tools which enable users to discover by themselves.

But there might be a few redundancies (repetitions) among this visualisation work since same features of data are mentioned by different sections. Also, there are some findings need to be explained by external factors. So it is always worth planning thoroughly before implementing, therefore we could avoid redundancy and make much of every data ink.

Another point is d3.js is quite different with other language (python, R), someone tend to think it is a language which lacks structure. In this way, it might be better to always write clear comments for codes.

Last, for a starter of d3, java script, css, also visualisation, it might be easy to get obsessed with the syntax, logic and structure this new language, css style. But we should bear in mind, that the purpose of visualisation is to covey and communicate your finding and insight based on data.