Economic Complexity: Final Report

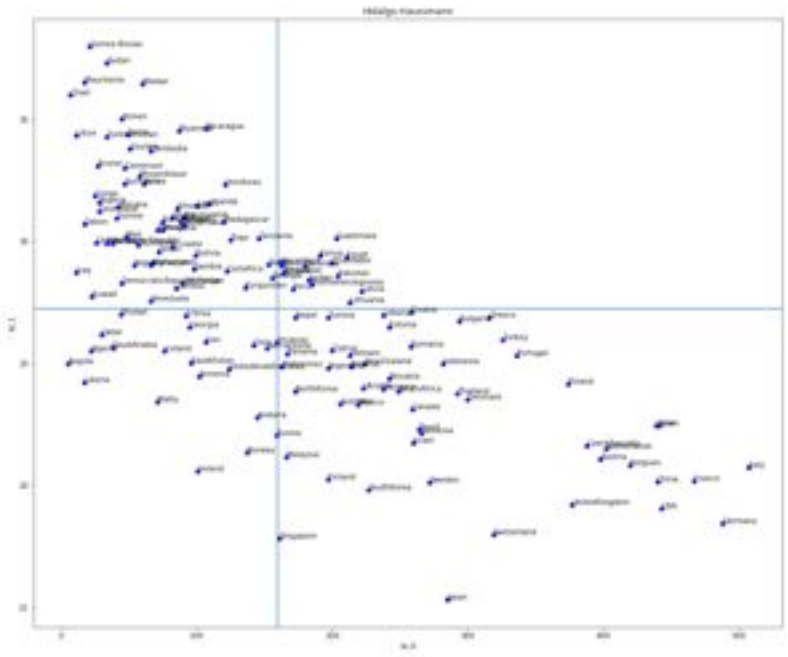
Due on Sat, jul 20, 2019

Prof.Benzi

Hossein Feizollah zadeh khoiee

Question 1

Apply algorithms: Qp, Fc → Free choice (ex. Comparison among country, restriction to only countries with fitness higher than a threshold)



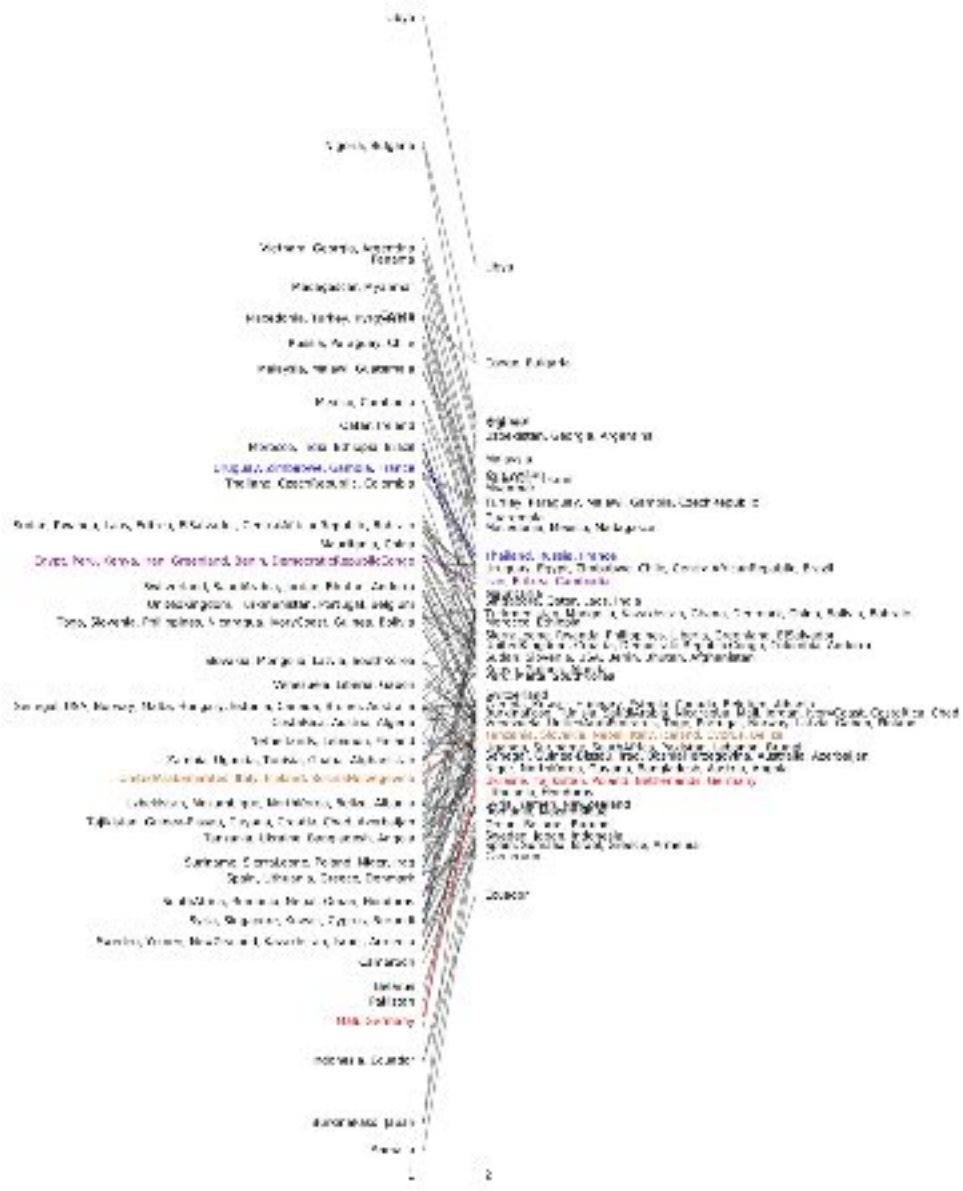
Before applying the **Qp,Fc algorithms** we need to understand what are the categories that distinguish countries in the figure above we can see based on bipartite network which connecting countries to products we can categorize countries to four different groups:

- Non-Diversified Countries Producing Standard Products (Up-left area)
- Diversified Countries Producing Standard Products(Up-right area)
- Non-Diversified Countries Producing Exclusive Products(Down-left area)
- Diversified Countries Producing Exclusive Products(Down-right area)

```
1
2  plt.figure(1,figsize=(18,15))
3  plt.title('Hidalgo-Haussmann')
4  plt.axvline(x=average(kc0))
5  plt.axhline(y=average(kc1))
6  plt.xlabel('kc,0')
7  plt.ylabel('kc,1')
8  plt.scatter(kc0,kc1, c='blue', label='AS')
9  for i, txt in enumerate(df['Country'][0]):
10     plt.annotate(txt, (kc0[i],kc1[i]))
11 plt.savefig("hidalgo.png")
```

Listing 1: Python code for figure above

Question 1 continued on next page...



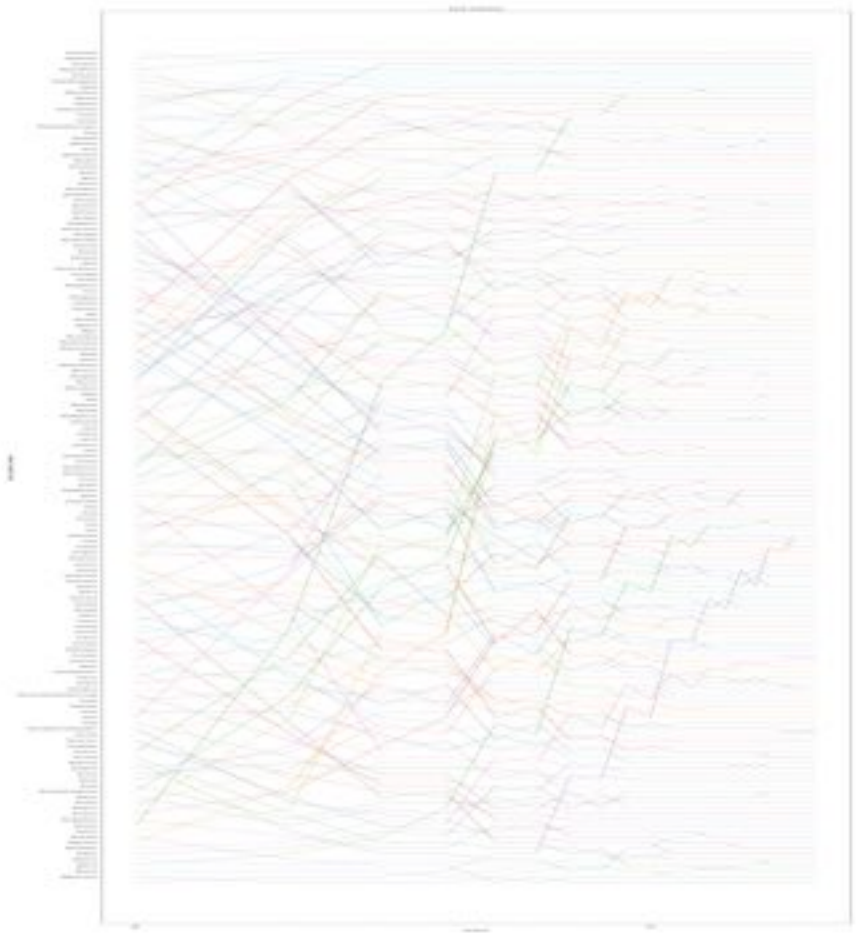
In this figure I consider all countries in our dataset and applying the Qp, Fc algorithm on them and continue the iterations until the algorithm reach to the convergence point or terminate after 1000 iteration(in this case all of them converge after 22 iteration), later I just picked the first point and last point of iteration and create a **Bump Chart** for all the countries in our database.

```
f = slope(data,width =15,height= 19,kind='interval',marker = None,font_size=12,color
    =color,savename='ranking.png',dpi=500)
```

Listing 2: the figure above created with the help of slop package

If we just consider the **Qp, Fc algorithm** then we can also show that in each iteration what will happen for a country as it shows in the figure below

Question 1 continued on next page...



in this part of the question I focus on countries with fitness higher than a threshold, the threshold here is chosen to be the average of the fitness for all countries. which in the year 2004 is ??

Question 2

Sensitivity: add an error and see changes:

1. **Sensitivity on Mcp: add 5%-20% error $\rightarrow$ randomly change 0-1 in matrix Mcp**
    - **Understand how rank changes (Fc, Qp)**
    - **How network changes (C(p0,p2))**
2. **Sensitivity on RCA: choose a threshold $(1-x, 1+x)$ $\rightarrow$ check the robustness**

Question 2 continued on next page...

(a) Sensitivity on Mcp: add 5%-20% error → randomly change 0-1 in matrix Mcp

According to the Associated Press (1988), a New York Fish and Wildlife technician named Richard Thomas calculated the volume of dirt in a typical 25–30 foot (7.6–9.1 m) long woodchuck burrow and had determined that if the woodchuck had moved an equivalent volume of wood, it could move "about **700 pounds (320 kg)** on a good day, with the wind at his back".

(b) Sensitivity on RCA: choose a threshold $(1-x, 1+x)$ *→ check the robustness.*

A woodchuck can ingest 361.92 cm^3 (22.09 cu in) of wood per day. Assuming immediate expulsion on ingestion with a 5% retainment rate, a woodchuck could chuck **343.82 cm^3** of wood per day.

Question 3

Product network $L(p_1, p_2)$:

- **look at the potential growth as a function of fitness or GDP or other**
- **probability ditribution of nodes (by its links)**

for computing $L(p_1, p_2)$ we need to first compute $B(p_1, p_2)$ with the code below:

```

B_p1_p2 = np.dot(df['mcp'].T,df['mcp'])
```

Listing 3: Python code for figure above

then we use this algorithm to create $L(p_1, p_2)$:

Data: $B(p_1, p_2)$
Result: $L(p_1, p_2)$
initialization;
If diag $B(p_1, p_2)$ is not 0 replace it with 0 ;
If $B(p_1, p_2)$>0 and U_{p1} > U_{p2} replace the array with 1;
otherwise;
put 0 for the array;

Algorithm 1: $L(p_1, p_2)$ algorithms

```
U_p = B_p1_p2.diagonal()
B_p1_p2 = B_p1_p2 -diag(diag(B_p1_p2))
shape = B_p1_p2.shape
L_p1_p2 = np.zeros(shape)
for x in range(0, shape[0]):
    for y in range(0, shape[1]):
        if U_p[x] > U_p[y]:
            if B_p1_p2[x, y] >= 0:
                L_p1_p2[x, y] = 1
```

Question 3 continued on next page...

Now that we have the adjacency matrix $L(p_1, p_2)$ we can do several things:

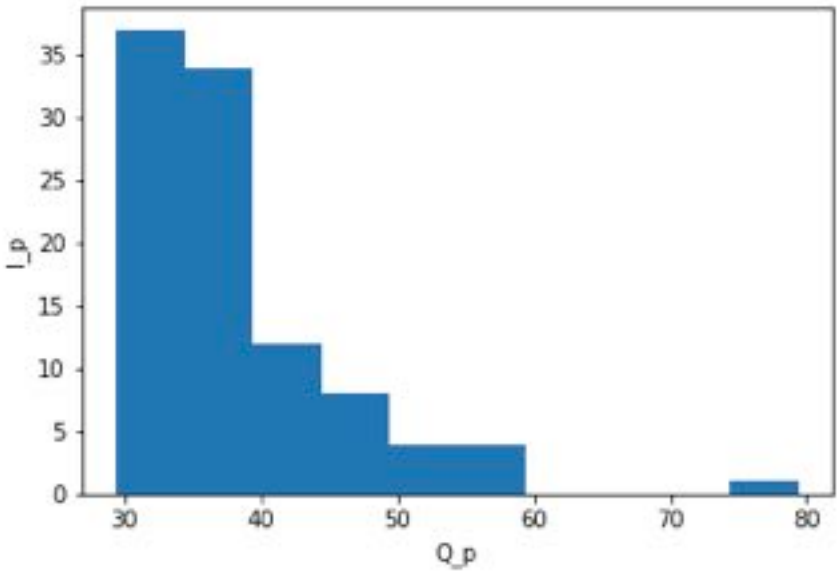
- computing the Pdf of $l(p) = \sum_{p_2} L(p_1, p_2)$ as a function of the Q_p
- we can use $B(p_1, p_2)$ to evaluate country potential increase in the fitness

(a) $Pdf(l(p), Q_p)$

```
shape = B_p1_p2.shape
L_p1_p2 = np.zeros(shape)
for x in range(0, shape[0]):
    for y in range(0, shape[1]):
        if Q_p[x] > Q_p[y]:
            if B_p1_p2[x, y] >= 0:
                L_p1_p2[x, y] = 1

l_p = L_p1_p2.sum(axis=0)
n, bins = numpy.histogram(l_p, bins=100, range=(0,100), weights=Q_p)
```

this distribution shows the number of products with complexity Q_p which are useful to build product of higher complexity.



(b) Country potential increase

(c) proximity matrix and graph of products

Proximity in the product space is defined as the minimum pair-wise conditional probability of co-exporting products p1 and p2. We can express this as a function of M as:

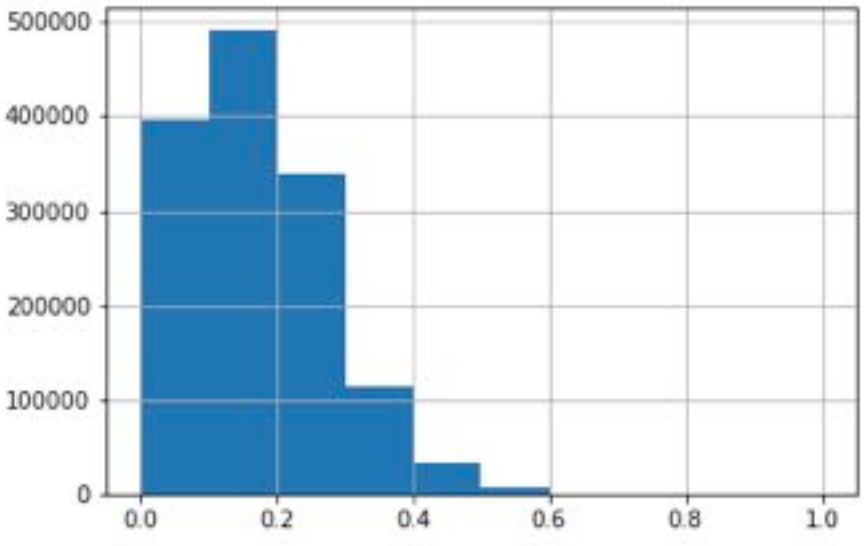
$$\Phi_{p_1 p_2} = min \frac{\sum_c M_{cp1} M_{cp2}}{\sum_c M_{cp1}} | \frac{\sum_c M_{cp1} M_{cp2}}{\sum_c M_{cp1}}$$

Question 3 continued on next page...

Proximity Φ is a quantity associated with a pair of products. We compare φ to $k_{p,0}$and $k_{p,1}$ by measuring the Euclidean distance in the $k_{p,0}$ and $k_{p,1}$ space:

$$\Delta_{p_1 p_2} = \sqrt{(k_{p_{1,0}} - k_{p_{2,0}})^2 + (k_{p_{1,1}} - k_{p_{2,1}})^2}$$

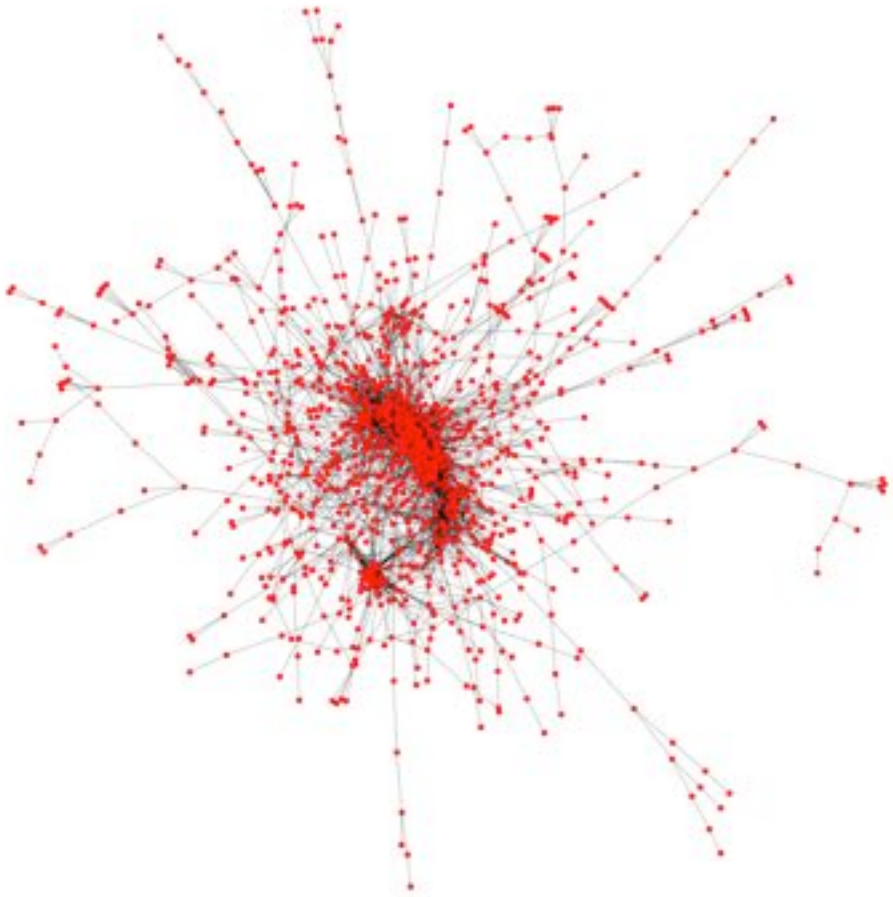
the distribution of product matrix for the year 2004 is shown in the figure below



this matrix shows the Product network for all of the products in matrix M_{cp}

productcode2	0	1	2	3	4	5	6	7	8	9	...	1160	1161	1162	1169
productcode1															
0	1.000000	0.290323	0.166667	0.210526	0.185185	0.163934	0.400000	0.333333	0.277778	0.388889	...	0.111111	0.055556	0.055556	0.111111
1	0.290323	1.000000	0.225806	0.354839	0.354839	0.311475	0.419355	0.161290	0.290323	0.161290	...	0.096774	0.064516	0.064516	0.064516
2	0.166667	0.225806	1.000000	0.052632	0.259259	0.163934	0.350000	0.062500	0.562500	0.125000	...	0.066667	0.250000	0.133333	0.133333
3	0.210526	0.354839	0.052632	1.000000	0.185185	0.196721	0.100000	0.187895	0.105263	0.263158	...	0.300000	0.000000	0.052632	0.300000
4	0.185185	0.354839	0.259259	0.185185	1.000000	0.295082	0.259259	0.274074	0.185185	0.274074	...	0.111111	0.111111	0.111111	0.148148
5	0.163934	0.311475	0.163934	0.196721	0.295082	1.000000	0.196721	0.081967	0.131148	0.081967	...	0.065574	0.049180	0.098361	0.049180
6	0.400000	0.419355	0.350000	0.100000	0.259259	0.196721	1.000000	0.400000	0.450000	0.300000	...	0.100000	0.100000	0.100000	0.100000
7	0.333333	0.161290	0.062500	0.157895	0.274074	0.081967	0.400000	1.000000	0.125000	0.312500	...	0.125000	0.000000	0.187500	0.062500
8	0.277778	0.290323	0.562500	0.105263	0.185185	0.131148	0.450000	0.125000	1.000000	0.125000	...	0.062500	0.125000	0.062500	0.062500
9	0.388889	0.161290	0.125000	0.263158	0.274074	0.081967	0.300000	0.312500	0.125000	1.000000	...	0.300000	0.062500	0.062500	0.125000

in the figure below we can see the product network of the year 2004 for all 1170 products each node represent a product.



```
plt.figure(1,figsize=(40,40))
products = df['Product'][0]
H=nx.relabel_nodes(ps,products)
nx.draw(ps)
plt.savefig("simple_path.png")
plt.show()
```

Listing 4: Python code for figure above

Question 4

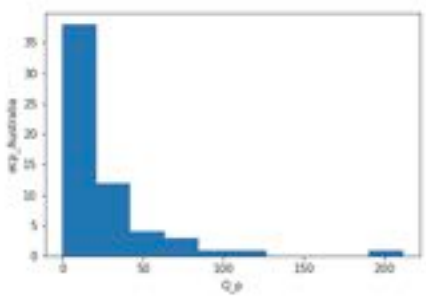
Change in Pdf for Qp for a given country:

1. $P(E_c * Q_p)$**: Pdf of money export * complexity product**
2. **free choice (ex. how much it changes in years).**

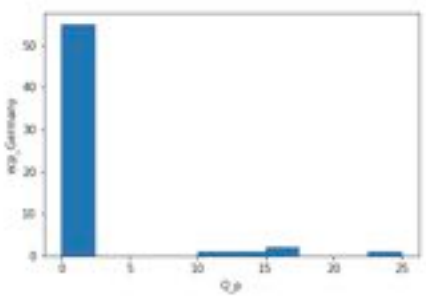
(a) Pdf of money export vs complexity product

By looking at the figure1 we can see that money export for different countries is based on how much that country is diversified in producing complex product for example in Iran

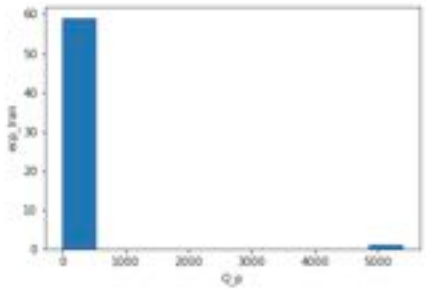
Question 4 continued on next page...



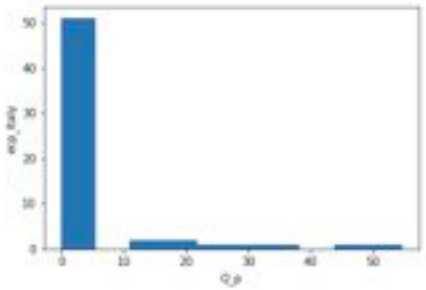
(a) Australia2012



(b) Germany2012



(c) Iran2012



(d) Italy2012

```
1
2 m, bins = numpy.histogram(ecp.T[61], bins=60, range=(0,60), weights=Q_p)
```

Listing 5: Python code for figure above

Figure 1: plots of Q_{rank} for different years for all countries

most of the money export is from Oil and mineral which has very low complexity and distribution is completely skewed to the not complex product, but for a country like Australia we have an smoother distribution because in this country income of export is not just because of simple products and this country has broad range of complexity in its products.

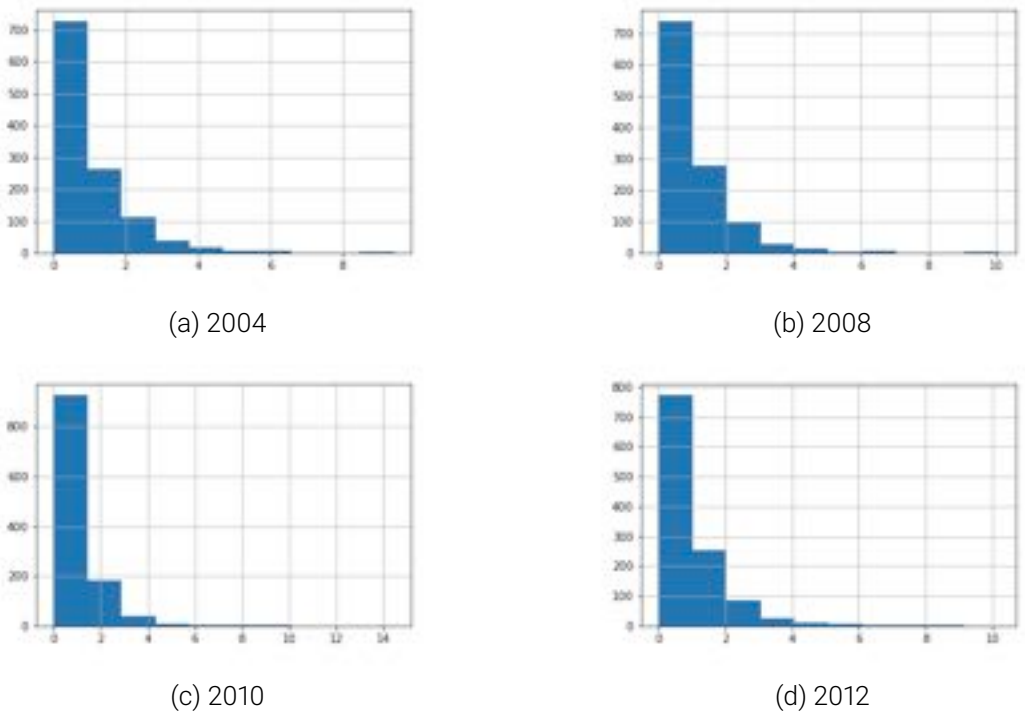
(b) changes in pdf of Q_p in different years

FURTHER STUDIES OF THE DATASET

Question 5

Can we provide any prediction

Question 5 continued on next page...



(a) 2004 (b) 2008

(c) 2010 (d) 2012

Figure 2: plots of Q_{rank} for different years for all countries

The figure below shows how distributions of Q_{rank} is changing for all countries and for four different years

Year	$Iran_{F_{rank}}$	$Italy_{F_{rank}}$
2004	0.81248	0.79704
2008	0.85313	0.8205
2010	0.79811	0.72796
2012	0.70858	0.6532