Productivity Index - Input

Emily Markowitz

Feb. 05, 2020

Contents

1 Math Theory: General Total Factor Productivity (TFP) Equation

The general form of the TFP can be measured as aggregate output (Y) divided by real total inputs (X). Rates of TFP growth are constructed using the Törnqvist index approach. The TFP growth over two time periods is defined as:

$$ln(TFP_t/TFP_{t-1}) = \sum_{i=1}^{n}((\frac{R_{i,t} + R_{i,t-1}}{2}) * ln(\frac{Y_{i,t}}{Y_{i,t-1}}))) - \sum_{j=1}^{m}((\frac{W_{j,t} + W_{j,t-1}}{2}) * ln(\frac{X_{j,t}}{X_{j,t-1}})))$$

Such that:

- Output $= \sum_{i=1}^{n}((\frac{R_{it}+R_{it-1}}{2}) * ln(\frac{Y_{it}}{Y_{it-1}}))$
- Input $= \sum_{j=1}^{n}((\frac{W_{jt}+W_{jt-1}}{2}) * ln(\frac{X_{jt}}{X_{jt-1}}))$

where:

- Y_i are individual outputs. This will later be refered to as Q_i in the following equations.
- X_j are individual inputs
- R_i are output revenue shares
- W_j are input cost shares
- t and $t-1$ are time subscripts, where 1 is the minimum year in the dataset
- i is category, e.g., Finfish (=1), Shellfish (=2)
- s is species, e.g., Salmon, Alewife, Surf Clams

---

2 Output Method: From Price to Quantity Measures

2.0.1 Variable Summary

Variables

- Q are individual quantity outputs in pounds (lbs).
- V are individual value outputs in dollars ($)
- R are output revenue shares
- P are prices
- PC are price changes
- PI are price indicies, often defined by a price from a base year $baseyr$
- $baseyr$ is the year to base all indicides from

Inidicies

- t and $t-1$ are time subscripts, where 1 is the minimum year in the dataset
- i is category, e.g., Finfish (=1), Shellfish (=2)
- s is species, e.g., Salmon, Alewife, Surf Clams

3 Input Method: From Price to Quantity Measures

- Input = $\sum_{j=1}^{n}((\frac{W_{jt}+W_{jt-1}}{2}) * ln(\frac{X_{jt}}{X_{jt-1}}))$

3.0.1 Variable Summary

Variables

- Q are individual quantity outputs in pounds (lbs).
- V are individual value outputs in dollars ($)
- W are weights of the data to the portion
- P are prices
- PC are price changes
- PI are price indicies, often defined by a price from a base year *baseyr*
- *baseyr* is the year to base all indicides from

Inidicies

- t and $t-1$ are time subscripts, where 1 is the minimum year in the dataset
- i is category, e.g., Finfish (=1), Shellfish (=2)

categories

- Capital (K): Length of the vessel times the percent time spent in the fishery. Some vessels spend time in more than one fishery.
- Labor Input (L): Crew Size * Days at Sea.
- Energy (E): Estimated quantity based on econometric models.
- Materials (M): Estimated quantity of bait and ice based on econometric models.
- Service (S)

3.0.2 Data requirements

We need time series data for the price of all categories (P_t; e.g., Total), price of each category (i) ($P_{i=1}$; e.g., K, L, E, M, and S), and the quantity of each category (i) ($Q_{i=1}$; e.g., K, L, E, M, and S):

3.0.2.1 Edit Data

We amy not have information for each of the K, L, E, M, and S categories, so the code has to be dynamic enough to only take for the data that is available.

```r
temp<-read.csv(file = paste0(dir.data, "Tornqvist Index-Calculations_InputEx.csv"))
rownames(temp)<-temp$year
temp$year<-NULL

temp.q<-temp[,grepl(pattern = "Q", x = names(temp))]
temp$QE0_0Total<-rowSums(temp.q, na.rm = T)

temp<-orgional.data<-cbind.data.frame(temp)
```

	P1_0K	Q1_0K	P2_0L	Q2_0L	P3_0E	Q3_0E	P4_0M	Q4_0M	P5_0S	Q5_0S	QE0_0Total
2007	1.00	10	1.00	2000	1.00	100	1.00	800	1.00	5	2915
2008	1.10	12	1.10	1900	1.20	120	1.20	1000	1.20	7	3039
2009	1.20	13	1.11	1600	1.30	110	1.25	900	1.22	9	2632
2010	1.40	14	1.20	1300	1.40	90	1.40	700	1.27	13	2117
2011	1.41	16	1.30	1100	1.55	80	1.60	900	1.30	19	2115
2012	1.45	15	1.40	1050	1.80	100	1.70	1000	1.31	21	2186
2013	1.46	16	1.41	1000	2.10	100	1.90	1200	1.35	26	2342
2014	1.50	17	1.43	1050	2.30	110	2.10	1100	1.36	29	2306
2015	1.60	19	1.50	950	2.60	90	2.40	1000	1.38	34	2093
2016	1.65	21	1.51	900	3.00	100	2.70	1200	1.40	38	2259

3.0.2.2 The nameing conventions of the column names.

For example, in "V0_0Total":

- "V"… refers to the variable represented in the column (here V = "Value")
- …"0"… refers to the whole fishery
- …"_"… is simply a seperator in the title
- Since this is the total, …"0".. refers to the index of the species, which is not relevant since this is the sum of the category, hense = 0
- …"Total" is purely descriptive (here the total for the fishery), so you can follow along with what is happening!

Similarly for "Q2_0K":

- "Q"… refers to the variable represented in the column (here Q = "Quantity")
- …"2"… refers to the category index (here, = Capital (K))
- …"_"… is simply a seperator in the title
- …"0".. refers to the index of the input value, here these are the total (= 0).
- …"K" is purely descriptive (here the name of the species), so you can follow along with what is happening!

Here I am just going to do some housekeeping items

```r
  NameBaseTotal<-substr(x = names(temp)[grep(x = names(temp), pattern = "0Total")][1],
                        start = 3, stop = nchar(names(temp)[grep(x = names(temp),
                                                                 pattern = "0Total")][1]))

  # Find which columns in this table are quantity Columns - we will need this for later
  QColumns<-grep(pattern = paste0("Q[0-9]+"),
```

```
                x = names(temp))

    # Find which columns in this table are price Columns - we will need this for later
    PColumns<-grep(pattern = paste0("P[0-9]+"),
                x = names(temp))

      # Find which columns in this table are value Columns - we will need this for later
  VColumns<-rep_len(x = "", length.out = length(QColumns))
  for (j in 1:length(VColumns)){
    VColumns[j]<-paste0("V", substr(x = names(temp)[QColumns[j]],
                                    start = 2,
                                    stop = nchar(names(temp)[QColumns[j]])))
  }
```

3.0.3 Value for each category ($V_{i,t}$; e.g., K, L, ...)

We first calculate the value for each input category.

Price for a species (s) of category (i) in year (t) =

$$V_{i,t} = VP_{i,t} * / * Q_{i,t}$$

where:

- $P_{i,t}$ is the price per category (i), for each year (t)
- $Q_{i,t}$ is the quantity per category (i), for each year (t)
- $V_{i,t}$ is the value ($) per category (i), for each year (t)

Here we calculate the value for each category

```
#####Value for each category#####
tempV<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
for (c in 1:length(PColumns)) {

    NameBase<-substr(start = 2,
                  stop = nchar(names(temp)[PColumns[c]]),
                  x = names(temp)[PColumns[c]])

    Q0<-temp[,names(temp) %in% paste0("Q", NameBase)]
    P0<-temp[,names(temp) %in% paste0("P", NameBase)] #to make sure its the same column
  tempV[,c]<-P0*Q0
  names(tempV)[c]<-paste0("V", NameBase ) #name the column
}

tempV<-data.frame(tempV)
temp<-cbind.data.frame(temp, tempV)
```

	Other...	V1_0K	V2_0L	V3_0E	V4_0M	V5_0S
2007	...	10.00	2000.0	100	800	5.00
2008	...	13.20	2090.0	144	1200	8.40
2009	...	15.60	1776.0	143	1125	10.98
2010	...	19.60	1560.0	126	980	16.51
2011	...	22.56	1430.0	124	1440	24.70

	Other…	V1_0K	V2_0L	V3_0E	V4_0M	V5_0S
2012	…	21.75	1470.0	180	1700	27.51
2013	…	23.36	1410.0	210	2280	35.10
2014	…	25.50	1501.5	253	2310	39.44
2015	…	30.40	1425.0	234	2400	46.92
2016	…	34.65	1359.0	300	3240	53.20

3.0.4 Sum value of the the fishery

$$V_t = \sum_{i=1}^{m} V_{i,t}$$

where:

- V_t is the value of the fishery by year

```
temp[,ncol(temp)+1]<-rowSums(tempV, na.rm = T)
names(temp)[ncol(temp)]<-paste0("V", NameBaseTotal) #name the column
```

	Other…	V2_0L	V3_0E	V4_0M	V5_0S	V0_0Total
2007	…	2000.0	100	800	5.00	2915.00
2008	…	2090.0	144	1200	8.40	3455.60
2009	…	1776.0	143	1125	10.98	3070.58
2010	…	1560.0	126	980	16.51	2702.11
2011	…	1430.0	124	1440	24.70	3041.26
2012	…	1470.0	180	1700	27.51	3399.26
2013	…	1410.0	210	2280	35.10	3958.46
2014	…	1501.5	253	2310	39.44	4129.44
2015	…	1425.0	234	2400	46.92	4136.32
2016	…	1359.0	300	3240	53.20	4986.85

3.0.5 Weight of each category in the fishery ($W_{i,t}$; e.g., K, L, …)

$$W_{i,t} = (P_{i,t} * Q_{i,t})/V_t$$

where:

- $W_{i,t}$ is the weight of the category (i) for each year (t) in the fishery

	W1_0K	W2_0L	W3_0E	W4_0M	W5_0S
2007	0.0034305	0.6861063	0.0343053	0.2744425	0.0017153
2008	0.0038199	0.6048154	0.0416715	0.3472624	0.0024308
2009	0.0050805	0.5783924	0.0465710	0.3663803	0.0035759
2010	0.0072536	0.5773266	0.0466302	0.3626795	0.0061100
2011	0.0074180	0.4701999	0.0407726	0.4734880	0.0081216
2012	0.0063985	0.4324471	0.0529527	0.5001088	0.0080929
2013	0.0059013	0.3561991	0.0530509	0.5759816	0.0088671
2014	0.0061752	0.3636086	0.0612674	0.5593979	0.0095509
2015	0.0073495	0.3445091	0.0565720	0.5802259	0.0113434
2016	0.0069483	0.2725167	0.0601582	0.6497087	0.0106681

3.0.6 Price Changes for each category ($PC_{i,t}$ aka $\Delta ln(P_{i,t})$; e.g., K, L, ...)

$$PC_{i,t} = ln(\frac{P_{i,t}}{P_{i,t-1}}) = \sum_{s=1}^{n}([\frac{W_{i,t}+W_{i,t-1}}{2}] * [ln(\frac{P_{i,t}}{P_{i,t-1}}]) = \sum_{s=1}^{n}([\frac{W_{i,t}+W_{i,t-1}}{2}] * [ln(P_{i,t}) - ln(P_{i,t-1})])$$

Where:

- category's (i) Price for each category (i), for each year (t) = $P_{i,t}$
- category's (i) Weight for each category (i), for each year (t) = $W_{i,t}$

Which can be adapted to this function/macro:

```
#A function to caluclate the price change
print(PriceChange)
```

```
## function(R0, P0) {
##   PC0<-rep_len(x = 0, length.out = length(P0))
##   for (t in 2:length(P0)) {
##     temp1<-((R0[t]+R0[t-1])/2)
##     temp2<-ln(P0[t]/P0[t-1])
##     PC0[t]<-temp1*temp2
##   }
##   return(PC0)
## }
## <bytecode: 0x00000000477d3488>
```

Now put it into practice for the total dataset:

```
#Find which columns in this table are price and revenue share columns
tempPC<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
for (c in 1:length(PColumns)){
  #For nameing columns
    NameBase<-substr(start = 2,
                     stop = nchar(names(temp)[QColumns[c]]),
                     x = names(temp)[QColumns[c]])

  # Calculate
  P0<-temp[, names(temp) %in% paste0("P", NameBase)]
  R0<-temp[, names(temp) %in% paste0("W", NameBase)] #to make sure its the same column
  tempPC[,c]<-PriceChange(R0, P0)
  names(tempPC)[c]<-paste0("PC", NameBase ) #name the column
}

temp<-cbind.data.frame(temp, tempPC)
```

For reference, here are the Price Changes for each species ($PC_{s,i,t}$):

	PC1_0K	PC2_0L	PC3_0E	PC4_0M	PC5_0S
2007	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2008	0.0003455	0.0615190	0.0069261	0.0566751	0.0003780
2009	0.0003872	0.0053539	0.0035316	0.0145662	0.0000496
2010	0.0009507	0.0450508	0.0034535	0.0413117	0.0001945
2011	0.0000522	0.0419234	0.0044480	0.0558273	0.0001661
2012	0.0001932	0.0334467	0.0070075	0.0295120	0.0000621
2013	0.0000423	0.0028066	0.0081703	0.0598444	0.0002551
2014	0.0001632	0.0050692	0.0051999	0.0568164	0.0000680

	PC1_0K	PC2_0L	PC3_0E	PC4_0M	PC5_0S
2015	0.0004364	0.0169207	0.0072237	0.0760878	0.0001525
2016	0.0002200	0.0020499	0.0083521	0.0724327	0.0001584

3.0.7 Summed Price Change for the fishery (PC_t)

$$PC_t = \sum_{i=1}^{m} PC_{i,t}$$

Where:

- category's (i) Price Change of the fishery, for each year (t) = PC_t

```
temp[ncol(temp)+1]<-rowSums(tempPC, na.rm = T)
names(temp)[ncol(temp)]<-paste0("PC", NameBaseTotal)
```

	Other…	PC2_0L	PC3_0E	PC4_0M	PC5_0S	PC0_0Total
2007	…	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2008	…	0.0615190	0.0069261	0.0566751	0.0003780	0.1258437
2009	…	0.0053539	0.0035316	0.0145662	0.0000496	0.0238885
2010	…	0.0450508	0.0034535	0.0413117	0.0001945	0.0909612
2011	…	0.0419234	0.0044480	0.0558273	0.0001661	0.1024171
2012	…	0.0334467	0.0070075	0.0295120	0.0000621	0.0702215
2013	…	0.0028066	0.0081703	0.0598444	0.0002551	0.0711186
2014	…	0.0050692	0.0051999	0.0568164	0.0000680	0.0673165
2015	…	0.0169207	0.0072237	0.0760878	0.0001525	0.1008211
2016	…	0.0020499	0.0083521	0.0724327	0.0001584	0.0832131

3.0.8 Price for the entire commercial fishery (P_t)

$$P_t = P_{t-1} * [1 + ln(\frac{P_{i,t}}{P_{i,t-1}})] = P_{t-1} * [1 - PC_t]$$

Which is represented using this function:

```
#Note that the first row of this column is = 1
tempP<-c(1, rep_len(x = NA, length.out = nrow(temp)-1))

PC0<-temp[,paste0("PC", NameBaseTotal)] #this is equal to ln(P_it/P_it-1)

  #Since the first row is defined, we need to start at the second row
for (t in 2:length(tempP)){
    tempP[t]<-tempP[t-1]*(1+PC0[t])
}

temp[,ncol(temp)+1]<-tempP
names(temp)[ncol(temp)]<-paste0("P", NameBaseTotal)
```

	Other…	PC3_0E	PC4_0M	PC5_0S	PC0_0Total	P0_0Total
2007	…	0.0000000	0.0000000	0.0000000	0.0000000	1.000000
2008	…	0.0069261	0.0566751	0.0003780	0.1258437	1.125844

	Other...	PC3_0E	PC4_0M	PC5_0S	PC0_0Total	P0_0Total
2009	...	0.0035316	0.0145662	0.0000496	0.0238885	1.152738
2010	...	0.0034535	0.0413117	0.0001945	0.0909612	1.257593
2011	...	0.0044480	0.0558273	0.0001661	0.1024171	1.386392
2012	...	0.0070075	0.0295120	0.0000621	0.0702215	1.483746
2013	...	0.0081703	0.0598444	0.0002551	0.0711186	1.589268
2014	...	0.0051999	0.0568164	0.0000680	0.0673165	1.696252
2015	...	0.0072237	0.0760878	0.0001525	0.1008211	1.867270
2016	...	0.0083521	0.0724327	0.0001584	0.0832131	2.022652

3.0.9 Price Index for the entire commercial fishery (PI_t)

$$PI_t = \frac{P_t}{P_{t=baseyr}}$$

In this example, we'll decide that the base year is 2010, for whatever reason. Notice that the $PI_{i,t=2007} = 1$

	Other...	PC4_0M	PC5_0S	PC0_0Total	P0_0Total	PI0_0Total
2007	...	0.0000000	0.0000000	0.0000000	1.000000	1.000000
2008	...	0.0566751	0.0003780	0.1258437	1.125844	1.125844
2009	...	0.0145662	0.0000496	0.0238885	1.152738	1.152738
2010	...	0.0413117	0.0001945	0.0909612	1.257593	1.257593
2011	...	0.0558273	0.0001661	0.1024171	1.386392	1.386392
2012	...	0.0295120	0.0000621	0.0702215	1.483746	1.483746
2013	...	0.0598444	0.0002551	0.0711186	1.589268	1.589268
2014	...	0.0568164	0.0000680	0.0673165	1.696252	1.696252
2015	...	0.0760878	0.0001525	0.1008211	1.867270	1.867270
2016	...	0.0724327	0.0001584	0.0832131	2.022652	2.022652

3.0.10 Total Implicit Quantity Input for the entire commercial fishery ($Q_t = Y_t$)

To get quantity estimates for total output using total value of landings divided by price index as follow: $Y = Q = V/I$

$$Q_t = V_t/PI_t$$

```
temp[,ncol(temp)+1]<-temp[,paste0("V", NameBaseTotal)]/temp[,paste0("PI", NameBaseTotal)]
names(temp)[ncol(temp)]<-paste0("Q", NameBaseTotal)
```

	Other...	PC5_0S	PC0_0Total	P0_0Total	PI0_0Total	Q0_0Total
2007	...	0.0000000	0.0000000	1.000000	1.000000	2915.000
2008	...	0.0003780	0.1258437	1.125844	1.125844	3069.343
2009	...	0.0000496	0.0238885	1.152738	1.152738	2663.727
2010	...	0.0001945	0.0909612	1.257593	1.257593	2148.637
2011	...	0.0001661	0.1024171	1.386392	1.386392	2193.651
2012	...	0.0000621	0.0702215	1.483746	1.483746	2290.998
2013	...	0.0002551	0.0711186	1.589268	1.589268	2490.744
2014	...	0.0000680	0.0673165	1.696252	1.696252	2434.449
2015	...	0.0001525	0.1008211	1.867270	1.867270	2215.169

	Other…	PC5_0S	PC0_0Total	P0_0Total	PI0_0Total	Q0_0Total
2016	…	0.0001584	0.0832131	2.022652	2.022652	2465.501

3.0.11 Total Implicit Quantity/Output Index

$$QI_t = Q_t / Q_{t=baseyr}$$

Where:

- QI is the sum of Q after these equations

```
temp$QI0_0Total<-temp$Q0_0Total/temp$Q0_0Total[rownames(temp) %in% baseyr]
```

	Other…	PC0_0Total	P0_0Total	PI0_0Total	Q0_0Total	QI0_0Total
2007	…	0.0000000	1.000000	1.000000	2915.000	1.0000000
2008	…	0.1258437	1.125844	1.125844	3069.343	1.0529477
2009	…	0.0238885	1.152738	1.152738	2663.727	0.9137999
2010	…	0.0909612	1.257593	1.257593	2148.637	0.7370966
2011	…	0.1024171	1.386392	1.386392	2193.651	0.7525389
2012	…	0.0702215	1.483746	1.483746	2290.998	0.7859341
2013	…	0.0711186	1.589268	1.589268	2490.744	0.8544575
2014	…	0.0673165	1.696252	1.696252	2434.449	0.8351455
2015	…	0.1008211	1.867270	1.867270	2215.169	0.7599208
2016	…	0.0832131	2.022652	2.022652	2465.501	0.8457979

3.0.12 Sum Total Implicit Quantity/Output Index (Optional)

$$QEI_t = QE_t / QE_{t=baseyr}$$

Where:

- QE is the sum of Q before these equations
- QEI is the index of the sum of Q before these equations

```
temp$QEI0_0Total<-temp$QE0_0Total/temp$QE0_0Total[rownames(temp) %in% baseyr]
```

	Other…	P0_0Total	PI0_0Total	Q0_0Total	QI0_0Total	QEI0_0Total
2007	…	1.000000	1.000000	2915.000	1.0000000	1.0000000
2008	…	1.125844	1.125844	3069.343	1.0529477	1.0425386
2009	…	1.152738	1.152738	2663.727	0.9137999	0.9029160
2010	…	1.257593	1.257593	2148.637	0.7370966	0.7262436
2011	…	1.386392	1.386392	2193.651	0.7525389	0.7255575
2012	…	1.483746	1.483746	2290.998	0.7859341	0.7499142
2013	…	1.589268	1.589268	2490.744	0.8544575	0.8034305
2014	…	1.696252	1.696252	2434.449	0.8351455	0.7910806
2015	…	1.867270	1.867270	2215.169	0.7599208	0.7180103
2016	…	2.022652	2.022652	2465.501	0.8457979	0.7749571

3.0.13 Solve Output portion of the equation for the Input Changes:

$$QC_t = \sum_{i=1}^{n}((\frac{R_{it} + R_{it-1}}{2}) * ln(\frac{Q_{it}}{Q_{it-1}}))$$

```
#Find which columns in this table are price and revenue share columns
tempQC<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
for (c in 1:length(PColumns)){
  #For nameing columns
    NameBase<-substr(start = 2,
                   stop = nchar(names(temp)[QColumns[c]]),
                   x = names(temp)[QColumns[c]])

  # Calculate
  P0<-temp[, names(temp) %in% paste0("Q", NameBase)]
  R0<-temp[, names(temp) %in% paste0("W", NameBase)] #to make sure its the same column
  tempQC[,c]<-PriceChange(R0, P0)
  names(tempQC)[c]<-paste0("QC", NameBase ) #name the column
}

temp<-cbind.data.frame(temp, tempQC)
```

For reference, here are the Price Changes for each species ($PC_{s,i,t}$):

	QC1_0K	QC2_0L	QC3_0E	QC4_0M	QC5_0S
2007	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2008	0.0006610	-0.0331078	0.0069261	0.0693647	0.0006975
2009	0.0003562	-0.1016673	-0.0038391	-0.0375949	0.0007548
2010	0.0004570	-0.1199864	-0.0093514	-0.0916116	0.0017809
2011	0.0009796	-0.0874968	-0.0051473	0.1050705	0.0027004
2012	-0.0004458	-0.0209956	0.0104571	0.0512893	0.0008114
2013	0.0003969	-0.0192391	0.0000000	0.0980972	0.0018111
2014	0.0003661	0.0175598	0.0054478	-0.0493955	0.0010056
2015	0.0007521	-0.0354354	-0.0118235	-0.0543089	0.0016618
2016	0.0007155	-0.0166804	0.0061494	0.1121218	0.0012241

3.0.14 Summed Price Change for the fishery (PC_t)

$$QC_t = \sum_{i=1}^{m} QC_{i,t}$$

Where:

- category's (i) Price Change of the fishery, for each year (t) = PC_t

```
temp[ncol(temp)+1]<-rowSums(tempQC, na.rm = T)
names(temp)[ncol(temp)]<-paste0("QC", NameBaseTotal)
```

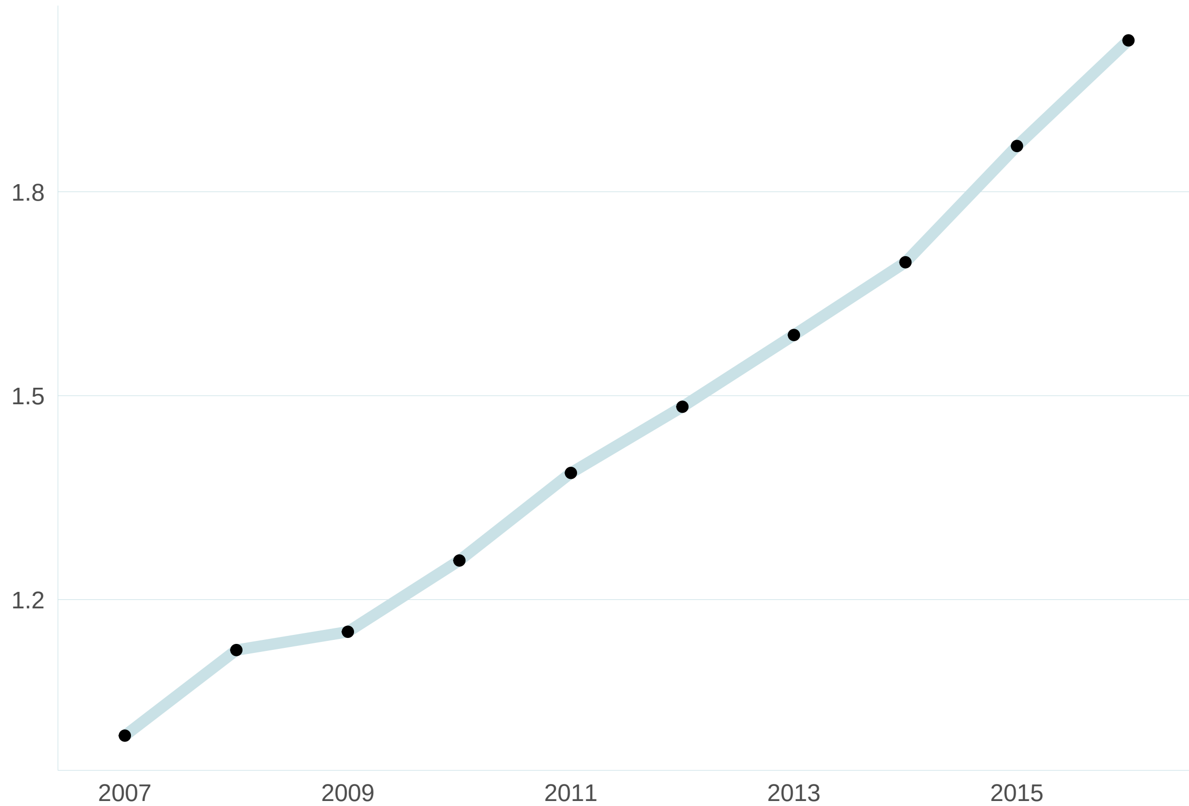
	Other...	QC2_0L	QC3_0E	QC4_0M	QC5_0S	QC0_0Total
2007	...	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2008	...	-0.0331078	0.0069261	0.0693647	0.0006975	0.0445415
2009	...	-0.1016673	-0.0038391	-0.0375949	0.0007548	-0.1419902

	Other…	QC2__0L	QC3__0E	QC4__0M	QC5__0S	QC0__0Total
2010	…	-0.1199864	-0.0093514	-0.0916116	0.0017809	-0.2187115
2011	…	-0.0874968	-0.0051473	0.1050705	0.0027004	0.0161064
2012	…	-0.0209956	0.0104571	0.0512893	0.0008114	0.0411164
2013	…	-0.0192391	0.0000000	0.0980972	0.0018111	0.0810662
2014	…	0.0175598	0.0054478	-0.0493955	0.0010056	-0.0250162
2015	…	-0.0354354	-0.0118235	-0.0543089	0.0016618	-0.0991538
2016	…	-0.0166804	0.0061494	0.1121218	0.0012241	0.1035303

	Q0__0Total	QI0__0Total	QC0__0Total
2007	2915.000	1.0000000	0.0000000
2008	3069.343	1.0529477	0.0445415
2009	2663.727	0.9137999	-0.1419902
2010	2148.637	0.7370966	-0.2187115
2011	2193.651	0.7525389	0.0161064
2012	2290.998	0.7859341	0.0411164
2013	2490.744	0.8544575	0.0810662
2014	2434.449	0.8351455	-0.0250162
2015	2215.169	0.7599208	-0.0991538
2016	2465.501	0.8457979	0.1035303

3.0.15 Graph 1

```
#A function I made to plot this pretty in ggplot2
plot1line(temp, PI = temp$PI0_0Total,
             NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



3.0.16 Graph 2

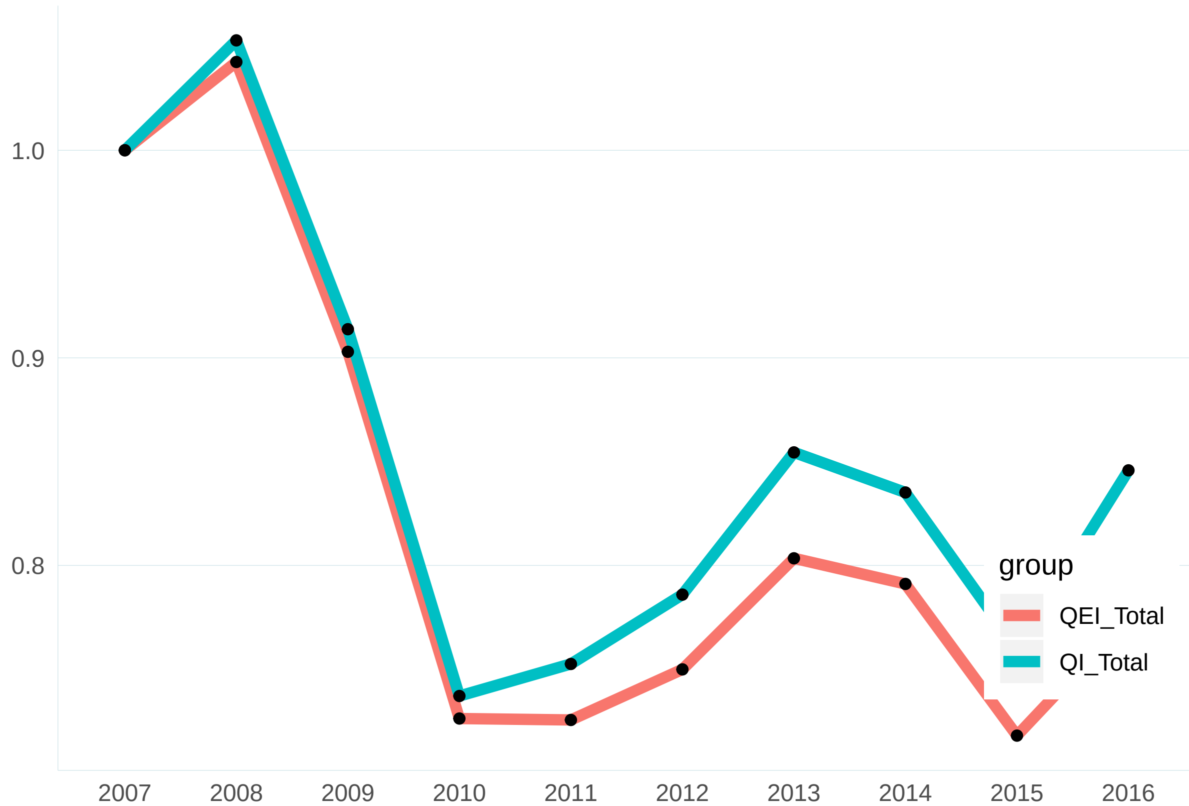
```
temp0<-temp
temp0$Year<-rownames(temp0)

tempA<-data.frame(temp0[,names(temp0) %in% c("Year", "QIO_0Total")])
names(tempA)<-c("Index", "Year")
tempA$group<-"QI_Total"
tempA$col<-NOAALightBlue

tempB<-data.frame(temp0[,names(temp0) %in% c("Year", "QEIO_0Total")])
names(tempB)<-c("Index", "Year")
tempB$group<-"QEI_Total"
tempB$col<-NOAADarkBlue

temp0<-rbind.data.frame(tempA, tempB)
rownames(temp0)<-NULL
temp0$col<-as.factor(temp0$col)

#A function I made to plot this pretty in ggplot2
plot2line(temp0, Year = temp0$Year, Index=temp0$Index, col = temp0$col, group = temp0$group,
                    NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



3.1 Do same analysis via a function!

Now that we know the method, we can simplify most of it into a function and do this whole analysis in 4 easy steps:

- A. Import and Edit data
- B. Enter base year
- C. Run the function
- D. Obtain the implicit quantity estimates

3.1.1 Function to calculate the Implicit Quanity Input at Fishery Level

```
print(ImplicitQuantityInput)
```

```
## function(temp, baseyr, calcQEI = F){
##
##   ####HOUSEKEEPING
##   NameBaseTotal<-substr(x = names(temp)[grep(x = names(temp), pattern = "0Total")][1],
##                         start = 3, stop = nchar(names(temp)[grep(x = names(temp),
##                                                                   pattern = "0Total")][1]))
##
##   # Find which columns in this table are quantity Columns - we will need this for later
##   QColumns<-grep(pattern = paste0("Q[0-9]+"),
```

```
##                  x = names(temp))
##
##    # Find which columns in this table are price Columns - we will need this for later
##    PColumns<-grep(pattern = paste0("P[0-9]+"),
##                  x = names(temp))
##
##    # Find which columns in this table are value Columns - we will need this for later
##    VColumns<-rep_len(x = "", length.out = length(QColumns))
##    for (j in 1:length(VColumns)){
##      VColumns[j]<-paste0("V", substr(x = names(temp)[QColumns[j]],
##                                      start = 2,
##                                      stop = nchar(names(temp)[QColumns[j]])))
##    }
##
##    ###Value for each category ($V_{i,t}$; e.g., K, L, ...)
##    # We first calculate the value for each input category.
##
##    # Price for a species (s) of category (i) in year (t) =
##    #   $$V_{i,t} = VP_{i,t}*/*Q_{i,t}$$
##
##    # where:
##    # - $P_{i,t}$ is the price per category (i), for each year (t)
##    # - $Q_{i,t}$ is the quantity per category (i), for each year (t)
##    # - $V_{i,t}$ is the value ($) per category (i), for each year (t)
##
##    # Here we calculate the value for each category
##    tempV<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
##    for (c in 1:length(PColumns)) {
##
##      NameBase<-substr(start = 2,
##                       stop = nchar(names(temp)[PColumns[c]]),
##                       x = names(temp)[PColumns[c]])
##
##      Q0<-temp[,names(temp) %in% paste0("Q", NameBase)]
##      P0<-temp[,names(temp) %in% paste0("P", NameBase)] #to make sure its the same column
##      tempV[,c]<-P0*Q0
##      names(tempV)[c]<-paste0("V", NameBase ) #name the column
##    }
##
##    tempV<-data.frame(tempV)
##    temp<-cbind.data.frame(temp, tempV)
##
##    ###Sum value of the the fishery
##    # $$V_t = \sum_{i=1}^{m} V_{i,t}$$
##
##    # where:
##    # - $V_{t}$ is the value of the fishery by year
##
##    temp[,ncol(temp)+1]<-rowSums(tempV, na.rm = T)
##    names(temp)[ncol(temp)]<-paste0("V", NameBaseTotal) #name the column
##
##    ##Weight of each category in the fishery ($W_{i,t}$; e.g., K, L, ...)
##    # $$W_{i,t} = (P_{i,t}*Q_{i,t})/V_{t}$$
##
```

```
##    # where:
##    # - $W_{i,t}$ is the weight of the category (i) for each year (t) in the fishery
##
##    tempW<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
##    for (c in 1:length(VColumns)) {
##
##      #for renaming the columns
##      NameBase<-substr(start = 2,
##                       stop = nchar(names(temp)[QColumns[c]]),
##                       x = names(temp)[QColumns[c]])
##
##      V<-temp[,grep(pattern = paste0("V", NameBaseTotal), x = names(temp))]
##      V0<-temp[,names(temp) %in% paste0("V", NameBase)] #to make sure its the same column
##      tempW[,c]<-V0/V
##      names(tempW)[c]<-paste0("W", NameBase ) #name the column
##    }
##
##    tempW<-data.frame(tempW)
##    temp<-cbind.data.frame(temp, tempW)
##
##    ###Price Changes for each species ($PC_{s,i,t}$ aka $\Delta ln(P_{s,i,t})$; e.g., Salmon and Flound
##    # $$PC_{i,t} = ln(\frac{P_{i,t}}{P_{i,t-1}}) = \sum_{s=1}^n([\frac{W_{i,t} + W_{i,t-1}}{2}] * [ln(
##
##    # Where:
##    # - category's (i) Price for each category (i), for each year (t) = $P_{i,t}$
##    # - category's (i) Weight for each category (i), for each year (t) = $W_{i,t}$
##
##    #Find which columns in this table are price and revenue share columns
##    tempPC<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
##    for (c in 1:length(PColumns)){
##      #For nameing columns
##      NameBase<-substr(start = 2,
##                       stop = nchar(names(temp)[QColumns[c]]),
##                       x = names(temp)[QColumns[c]])
##
##      # Calculate
##      P0<-temp[, names(temp) %in% paste0("P", NameBase)]
##      R0<-temp[, names(temp) %in% paste0("W", NameBase)] #to make sure its the same column
##      tempPC[,c]<-PriceChange(R0, P0)
##      names(tempPC)[c]<-paste0("PC", NameBase ) #name the column
##    }
##
##    temp<-cbind.data.frame(temp, tempPC)
##
##    ###Summed Price Change for the fishery ($PC_{t}$)
##    # $$PC_{t} = \sum_{i=1}^{m} PC_{i,t}$$
##
##    # Where:
##    # - category's (i) Price Change of the fishery, for each year (t) = $PC_{t}$
##
##    temp[ncol(temp)+1]<-rowSums(tempPC, na.rm = T)
##    names(temp)[ncol(temp)]<-paste0("PC", NameBaseTotal)
##
##    ###Price for the entire commercial fishery ($P_t$)
```

```
##    # $$P_t = P_{t-1}*[1+ln(\frac{P_{i,t}}{P_{i,t-1}})] = P_{t-1}*[1-PC_{t}]$$
##
##    #Note that the first row of this column is = 1
##    tempP<-c(1, rep_len(x = NA, length.out = nrow(temp)-1))
##
##    PC0<-temp[,paste0("PC", NameBaseTotal)] #this is equal to ln(P_it/P_it-1)
##
##    #Since the first row is defined, we need to start at the second row
##    for (t in 2:length(tempP)){
##      tempP[t]<-tempP[t-1]*(1+PC0[t])
##    }
##
##    temp[,ncol(temp)+1]<-tempP
##    names(temp)[ncol(temp)]<-paste0("P", NameBaseTotal)
##
##    ###Price Index for the entire commercial fishery ($PI_t$)
##    # $$PI_{t} = \frac{P_{t}}{P_{t=baseyr}}$$
##
##    # In this example, we'll decide that the base year is 2010, for whatever reason. Notice that the $I
##
##    temp[,ncol(temp)+1]<-temp[,paste0("P", NameBaseTotal)]/temp[rownames(temp) %in% baseyr,paste0("P",
##    names(temp)[ncol(temp)]<-paste0("PI", NameBaseTotal)
##
##    ### Total Implicit Quantity Input for the entire commercial fishery ($Q_t = Y_t$)
##    # To get quantity estimates for total output using total value of landings divided by price index a
##    # Note here that all columns of $V$ are being used, despite having been removed earlier in the anal
##    # $$Q_{t}=V_{t}/PI_{t}$$
##
##    temp[,ncol(temp)+1]<-temp[,paste0("V", NameBaseTotal)]/temp[,paste0("PI", NameBaseTotal)]
##    names(temp)[ncol(temp)]<-paste0("Q", NameBaseTotal)
##
##    ### Total Implicit Quantity/Output Index
##    # $$QI_t = Q_t/Q_{t=baseyr}$$
##
##    # Where:
##    # - $QI$ is the sum of Q after these equations
##
##    temp[,ncol(temp)+1]<-temp[,paste0("Q", NameBaseTotal)]/temp[rownames(temp) %in% baseyr, paste0("Q"
##    names(temp)[ncol(temp)]<-paste0("QI", NameBaseTotal)
##
##    ### Sum Total Implicit Quantity/Output Index (Optional)
##    # $$QEI_t = QE_t/QE_{t=baseyr}$$
##
##    # Where:
##    # - $QE$ is the sum of Q before these equations
##    # - $QEI$ is the index of the sum of Q before these equations
##
##    temp[,ncol(temp)+1]<-temp[,paste0("QE", NameBaseTotal)]/temp[rownames(temp) %in% baseyr, paste0("QE
##    names(temp)[ncol(temp)]<-paste0("QEI", NameBaseTotal)
##    ### Solve Output portion of the equation for the Input Changes:
##    # $$QC_t = \sum_{i=1}^n((\frac{R_{it} + R_{it-1}}{2}) * ln(\frac{Q_{it}}{Q_{it-1}}))$$
##
##    #Find which columns in this table are price and revenue share columns
##    tempQC<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
```

```
##   for (c in 1:length(PColumns)){
##     #For nameing columns
##     NameBase<-substr(start = 2,
##                      stop = nchar(names(temp)[QColumns[c]]),
##                      x = names(temp)[QColumns[c]])
##
##     # Calculate
##     P0<-temp[, names(temp) %in% paste0("Q", NameBase)]
##     R0<-temp[, names(temp) %in% paste0("W", NameBase)] #to make sure its the same column
##     tempQC[,c]<-PriceChange(R0, P0)
##     names(tempQC)[c]<-paste0("QC", NameBase ) #name the column
##   }
##
##   temp<-cbind.data.frame(temp, tempQC)
##
##   ###Summed Price Change for the fishery ($PC_{t}$)
##   # $$QC_{t} = \sum_{i=1}^{m} QC_{i,t}$$
##
##   # Where:
##   # - category's (i) Price Change of the fishery, for each year (t) = $PC_{t}$
##
##   temp[ncol(temp)+1]<-rowSums(tempQC, na.rm = T)
##   names(temp)[ncol(temp)]<-paste0("QC", NameBaseTotal)
##
##   #Remove duplicate columns
##   temp<-temp[, !(grepl(pattern = "\\.[0-9]+", x = names(temp)))]
##   temp <- temp[, !duplicated(colnames(temp))]
##
##   return(temp)
##
## }
## <bytecode: 0x00000000493b2b28>
```

3.1.2 A. Import and Edit data

```r
temp<-read.csv(file = paste0(dir.data, "Tornqvist Index-Calculations_InputEx.csv"))
rownames(temp)<-temp$year
temp$year<-NULL

temp.q<-temp[,grepl(pattern = "Q", x = names(temp))]
temp$QE0_0Total<-rowSums(temp.q, na.rm = T)

temp<-orgional.data<-cbind.data.frame(temp)
```

3.1.3 B. Enter base year

```r
baseyr<-2007
```

3.1.4 C. Run the function

```
temp<-ImplicitQuantityInput(temp, baseyr, calcQEI = T)
```

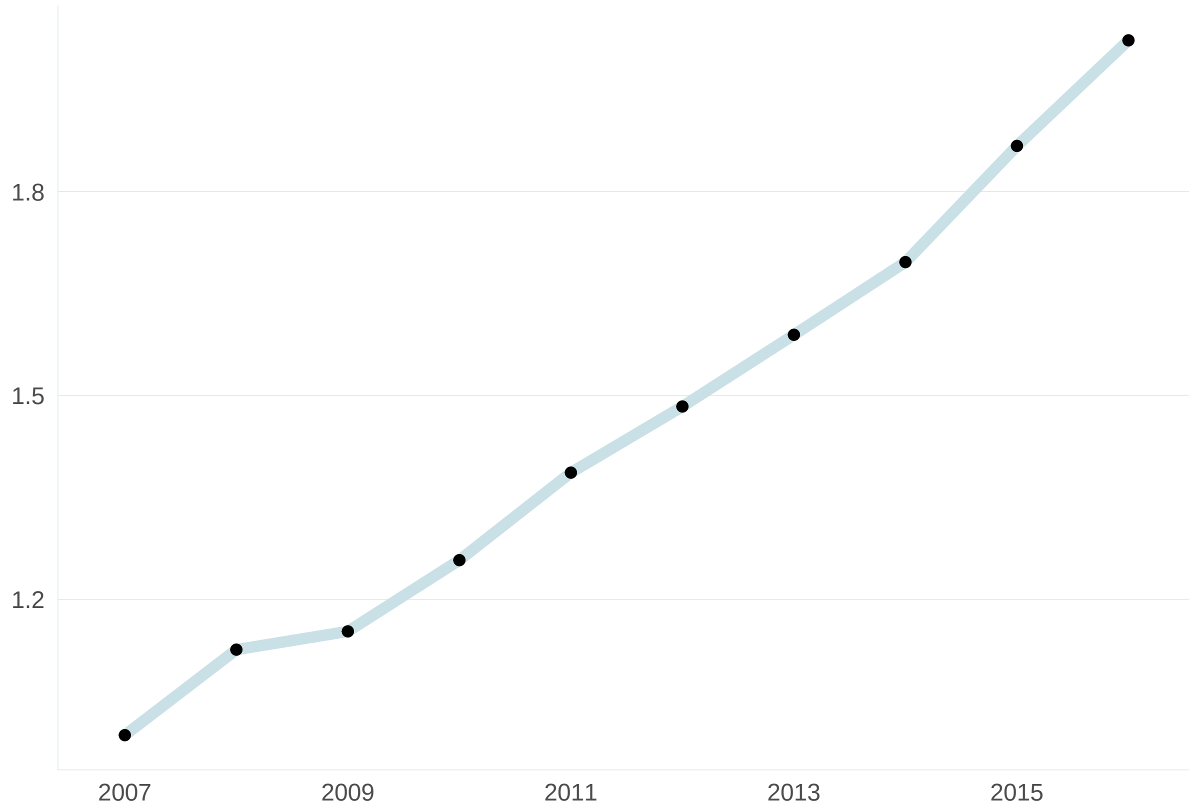
3.1.5 D. Obtain the implicit quantity estimates

	QE0_0Total	V0_0Total	PC0_0Total	P0_0Total	PI0_0Total	Q0_0Total	QI0_0Total	QEI0_0Total
2007	2915	2915.00	0.0000000	1.000000	1.000000	2915.000	1.0000000	1.0000000
2008	3039	3455.60	0.1258437	1.125844	1.125844	3069.343	1.0529477	1.0425386
2009	2632	3070.58	0.0238885	1.152738	1.152738	2663.727	0.9137999	0.9029160
2010	2117	2702.11	0.0909612	1.257593	1.257593	2148.637	0.7370966	0.7262436
2011	2115	3041.26	0.1024171	1.386392	1.386392	2193.651	0.7525389	0.7255575
2012	2186	3399.26	0.0702215	1.483746	1.483746	2290.998	0.7859341	0.7499142
2013	2342	3958.46	0.0711186	1.589268	1.589268	2490.744	0.8544575	0.8034305
2014	2306	4129.44	0.0673165	1.696252	1.696252	2434.449	0.8351455	0.7910806
2015	2093	4136.32	0.1008211	1.867270	1.867270	2215.169	0.7599208	0.7180103
2016	2259	4986.85	0.0832131	2.022652	2.022652	2465.501	0.8457979	0.7749571

3.1.6 Graph 1

For comparison, let's recreate those graphs to make sure we are getting the same output:

```
#A function I made to plot this pretty in ggplot2
plot1line(temp, PI = temp$PI0_0Total,
          NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



3.1.7 Graph 2

For comparison, let's recreate those graphs to make sure we are getting the same output:

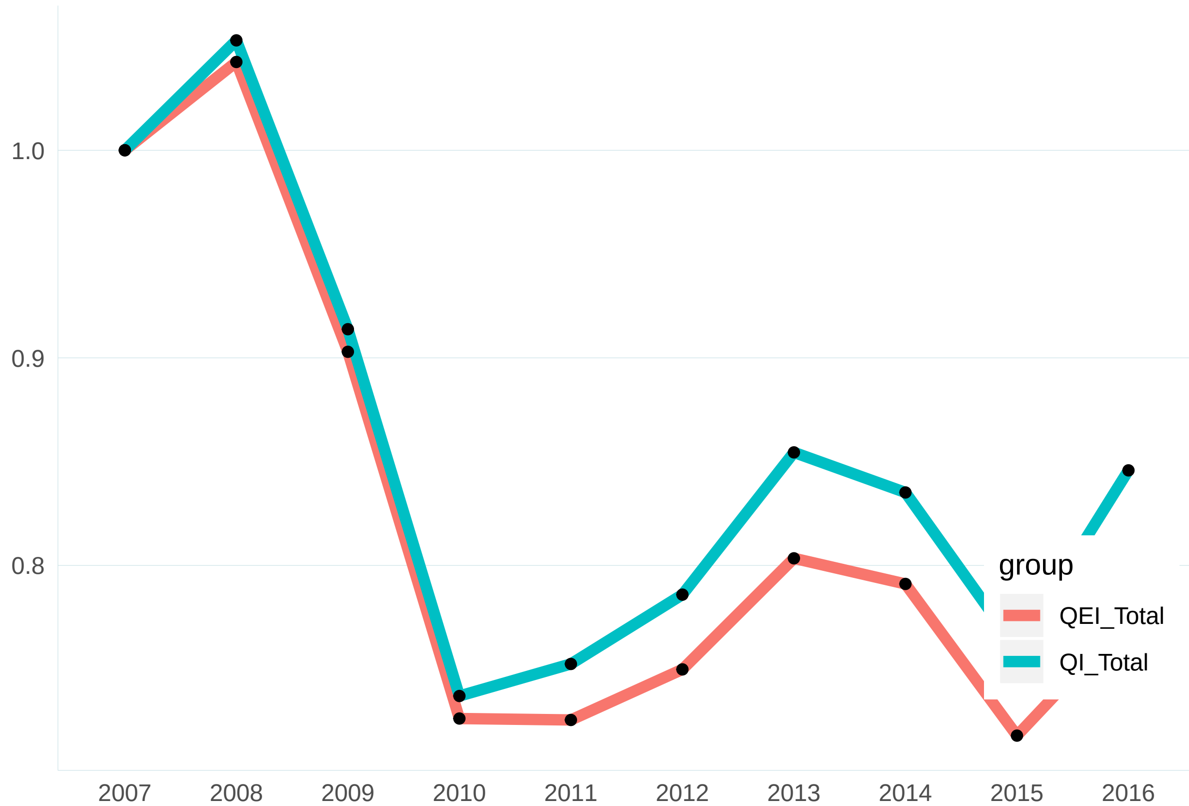
```
temp0<-temp
temp0$Year<-rownames(temp0)

tempA<-data.frame(temp0[,names(temp0) %in% c("Year", "QI0_0Total")])
names(tempA)<-c("Index", "Year")
tempA$group<-"QI_Total"
tempA$col<-NOAALightBlue

tempB<-data.frame(temp0[,names(temp0) %in% c("Year", "QEI0_0Total")])
names(tempB)<-c("Index", "Year")
tempB$group<-"QEI_Total"
tempB$col<-NOAADarkBlue

temp0<-rbind.data.frame(tempA, tempB)
rownames(temp0)<-NULL
temp0$col<-as.factor(temp0$col)

plot2line(temp0, Year = temp0$Year, Index=temp0$Index, col = temp0$col, group = temp0$group,
                    NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



4 Use examples to do the whole TFP Equation

4.1 A. Import and Edit data

Output

```
temp<-read.csv(file = paste0(dir.data, "Tornqvist Index-Calculations_OutputEx.csv"))
rownames(temp)<-temp$year
temp$year<-NULL

temp.q<-temp[,grepl(pattern = "Q", x = names(temp))]
temp.q$QE0_0Total<-rowSums(temp.q, na.rm = T)
temp.q$QE1_0Finfish<-rowSums(temp.q[,grepl(x = names(temp.q), pattern = "Q1") ], na.rm = T)
temp.q$QE2_0Shellfish<-rowSums(temp.q[,grepl(x = names(temp.q), pattern = "Q2") ], na.rm = T)

temp.v<-temp[,grepl(pattern = "V", x = names(temp))]
temp.v$V0_0Total<-rowSums(temp.v, na.rm = T)
temp.v$V1_0Finfish<-rowSums(temp.v[,grepl(x = names(temp.v), pattern = "V1") ], na.rm = T)
temp.v$V2_0Shellfish<-rowSums(temp.v[,grepl(x = names(temp.v), pattern = "V2") ], na.rm = T)

temp.output<-cbind.data.frame(temp.q, temp.v)
```

Input:

```r
temp<-read.csv(file = paste0(dir.data, "Tornqvist Index-Calculations_InputEx.csv"))
rownames(temp)<-temp$year
temp$year<-NULL

temp.q<-temp[,grepl(pattern = "Q", x = names(temp))]
temp$QE0_0Total<-rowSums(temp.q, na.rm = T)

temp.input<-orgional.data<-cbind.data.frame(temp)
```

4.2 B. Enter base year

```r
baseyr<-2007
```

4.3 C. Run the function

4.3.1 Function Method 1 to Calcilate the TFP

$$TFP_t = \frac{Y_t}{X_t}$$

$$TFPCR_t = ln(TFP_t/TFP_{t-1})$$

where

- $TFPCR_t$ is the change rate of the TFP over time t

```r
print(TFP_ChangeRate_Method1)
```

```
## function(temp.output, temp.input, baseyr, calcQEI = T, PercentMissingThreshold){
##
##   NumberOfSpecies<-numbers0(x = c(0, strsplit(x =
##                                                     strsplit(x = names(temp)[1],
##                                                              split = "_")[[1]][2],
##                                                   split = "[a-zA-Z]")[[1]][1]))[1]
##
##   #OUTPUT
##   temp00<-ImplicitQuantityOutput(temp.output, baseyr, calcQEI = T, PercentMissingThreshold)
##   temp<-temp00[[1]]
##   warnings.list0<-temp00[[2]]
##   figures.list0<-temp00[[3]]
##
##   names(temp)<-paste0(gsub(pattern = "Q", replacement = "Y",
##                            x = substr(x = names(temp), start = 1, stop = 1)),
##                       substr(x = names(temp), start = 2, stop = nchar(names(temp))))
##   temp.output1<-temp
##
##
##   #input
##   temp<-ImplicitQuantityInput(temp.input, baseyr, calcQEI)
##   names(temp)<-paste0(gsub(pattern = "Q", replacement = "X",
##                            x = substr(x = names(temp), start = 1, stop = 1)),
##                       substr(x = names(temp), start = 2, stop = nchar(names(temp))))
```

```
##   temp.input1<-temp
##
##   #Calculate
##   TFP1<-temp.output1[,grep(pattern = paste0("Y",NumberOfSpecies,".*Total"), x = names(temp.output1))]
##     temp.input1[,grep(pattern = paste0("X",NumberOfSpecies,".*Total"), x = names(temp.input1))]
##
##
##   TFP1_CR<-rep_len(x = 0, length.out = length(TFP1))
##   for (i in 2:length(TFP1)) {
##     TFP1_CR[i] <- ln(TFP1[i]/TFP1[i-1])
##   }
##
##   temp<-cbind.data.frame(TFP1, TFP1_CR,
##                              temp.output1,#[,grep(pattern = paste0("Y",NumberOfSpecies,".*Total"), x = na
##                                temp.input1)#[,grep(pattern = paste0("X",NumberOfSpecies,".*Total"), x = n
##
##   return(temp)
## }
## <bytecode: 0x00000000471720f0>
```

	Y_Total	X_Total	TFP	TFP_CR
2007	5600.000	2915.000	1.921098	0.0000000
2008	5988.260	3069.343	1.950991	0.0154407
2009	5588.490	2663.727	2.097997	0.0726455
2010	4421.125	2148.637	2.057642	-0.0194224
2011	4098.015	2193.651	1.868125	-0.0966251
2012	4225.432	2290.998	1.844363	-0.0128016
2013	5661.287	2490.744	2.272930	0.2089360
2014	5760.050	2434.449	2.366059	0.0401557
2015	5865.623	2215.169	2.647935	0.1125543
2016	6087.917	2465.501	2.469241	-0.0698690

4.3.2 Function Method 2 to Calcilate the TFP

$$ln(TFP_t/TFP_{t-1}) = \sum_{i=1}^{n}((\frac{R_{i,t}+R_{i,t-1}}{2}) * ln(\frac{Y_{i,t}}{Y_{i,t-1}}))) - \sum_{j=1}^{m}((\frac{W_{j,t}+W_{j,t-1}}{2}) * ln(\frac{X_{j,t}}{X_{j,t-1}})))$$

```r
print(TFP_ChangeRate_Method2)
```

```
## function(temp.output, temp.input, baseyr, calcQEI = T, PercentMissingThreshold){
##
##   #OUTPUT
##   temp00<-ImplicitQuantityOutput(temp.output, baseyr, calcQEI = T, PercentMissingThreshold)
##   temp<-temp00[[1]]
##   warnings.list0<-temp00[[2]]
##   figures.list0<-temp00[[3]]
##
##   names(temp)<-paste0(gsub(pattern = "Q", replacement = "Y",
##                                x = substr(x = names(temp), start = 1, stop = 1)),
##                         substr(x = names(temp), start = 2, stop = nchar(names(temp))))
##   temp.output1<-temp
##
```

```
##
##   #input
##   temp<-ImplicitQuantityInput(temp.input, baseyr, calcQEI)
##   names(temp)<-paste0(gsub(pattern = "Q", replacement = "X",
##                            x = substr(x = names(temp), start = 1, stop = 1)),
##                       substr(x = names(temp), start = 2, stop = nchar(names(temp))))
##   temp.input1<-temp
##
##   TFP <- temp.output1[,grep(pattern = "YC.*Total", x = names(temp.output1))] -
##     temp.input1[,grep(pattern = "XC.*Total", x = names(temp.input1))]
##
##   TFP2_CR<-data.frame(TFP)
##   names(TFP2_CR)<-"TFP2_CR"
##   rownames(TFP2_CR)<-rownames(temp.output1)
##
##   TFP2_CR<-cbind.data.frame(TFP2_CR,
##                             temp.output1,#[,grep(pattern = "YC.*Total", x = names(temp.output1))],
##                             temp.input1)#[,grep(pattern = "XC.*Total", x = names(temp.input1))])
##
##   return(TFP2_CR)
## }
## <bytecode: 0x000000004a807998>
```

```
TFP2_CR<-TFP_ChangeRate_Method2(temp.output, temp.input, baseyr, calcQEI = T, PercentMissingThreshold)
```

4.3.3 Method 1 using YE (summed, not calculated, output) for comparison

```
  NumberOfSpecies<-numbers0(x = c(0, strsplit(x =
                                               strsplit(x = names(temp)[1],
                                                        split = "_")[[1]][2],
                                             split = "[a-zA-Z]")[[1]][1]))[1]

  #OUTPUT
  temp00<-ImplicitQuantityOutput(temp.output, baseyr, calcQEI = T, PercentMissingThreshold)
temp<-temp00[[1]]
warnings.list0<-temp00[[2]]
figures.list0<-temp00[[3]]

  names(temp)<-paste0(gsub(pattern = "Q", replacement = "Y",
                           x = substr(x = names(temp), start = 1, stop = 1)),
                      substr(x = names(temp), start = 2, stop = nchar(names(temp))))
  temp.output1<-temp


  #input
  temp<-ImplicitQuantityInput(temp.input, baseyr, calcQEI = T)
  names(temp)<-paste0(gsub(pattern = "Q", replacement = "X",
                           x = substr(x = names(temp), start = 1, stop = 1)),
                      substr(x = names(temp), start = 2, stop = nchar(names(temp))))
  temp.input1<-temp

  #Calculate
  TFP1<-temp.output1[,grep(pattern = paste0("YE",NumberOfSpecies,".*Total"), x = names(temp.output1))] /
```

```
  temp.input1[,grep(pattern = paste0("X",NumberOfSpecies,".*Total"), x = names(temp.input1))]


TFP1E_CR<-rep_len(x = 0, length.out = length(TFP1))
for (i in 2:length(TFP1)) {
  TFP1E_CR[i] <- ln(TFP1[i]/TFP1[i-1])
}

TFP1E_CR<-data.frame(TFP1E_CR)
names(TFP1E_CR)<-c("TFP1_CR")
rownames(TFP1E_CR)<-rownames(temp.output1)
```

	TFP1_CR
2007	0.0000000
2008	-0.0123729
2009	0.0712640
2010	0.0268404
2011	-0.0678013
2012	-0.0752467
2013	0.2149126
2014	0.0767500
2015	0.1507726
2016	-0.1223337

4.3.4 Method 2 using YE (summed, not calculated, output) for comparison

```
#OUTPUT

temp00<-ImplicitQuantityOutput(temp.output, baseyr, calcQEI = T, PercentMissingThreshold)
temp<-temp00[[1]]
warnings.list0<-temp00[[2]]
figures.list0<-temp00[[3]]

names(temp)<-paste0(gsub(pattern = "Q", replacement = "Y",
                         x = substr(x = names(temp), start = 1, stop = 1)),
                    substr(x = names(temp), start = 2, stop = nchar(names(temp))))
temp.output1<-temp

  temp$YEC0_0Total<-rowSums(cbind(PriceChange(R0 = temp$R1_0Finfish, P0 = temp$YE1_0Finfish),
                          PriceChange(R0 = temp$R2_0Shellfish, P0 = temp$YE2_0Shellfish)),
                     na.rm = T)


#input
temp<-ImplicitQuantityInput(temp.input, baseyr, calcQEI = T)
names(temp)<-paste0(gsub(pattern = "Q", replacement = "X",
                         x = substr(x = names(temp), start = 1, stop = 1)),
                    substr(x = names(temp), start = 2, stop = nchar(names(temp))))
temp.input1<-temp
```

```
TFP <- temp.output1[,grep(pattern = "YC.*Total", x = names(temp.output1))] -
  temp.input1[,grep(pattern = "XC.*Total", x = names(temp.input1))]

TFP2E_CR<-data.frame(TFP)
names(TFP2E_CR)<-"TFP2_CR"
rownames(TFP2E_CR)<-rownames(temp.output1)
```

```
TFP2E_CR %>%
    knitr::kable(row.names = T, booktabs = T)
```

	TFP2_CR
2007	0.0000000
2008	0.0225110
2009	0.0728667
2010	-0.0187627
2011	-0.0919086
2012	-0.0104731
2013	0.2125162
2014	0.0422939
2015	0.1173105
2016	-0.0662898

4.4 D. Obtain the implicit quantity estimates

```
## Graph 1
# For comparison, let's recreate those graphs to make sure we are getting the same output:

#A function I made to plot this pretty in ggplot2
# TFP<-cbind.data.frame(TFP1_CR, TFP2_CR)
# plot2line(temp0 = TFP, PI = TFP$TFP1,
            # NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```

4.5 Graph 2: Compare methods

```
TFP$TFP1<-NULL
```

```
## Warning in TFP$TFP1 <- NULL: Coercing LHS to a list
```

```
TFP1_CR$Year<-rownames(TFP1_CR)
tempA<-data.frame(TFP1_CR[,names(TFP1_CR) %in% c("Year", "TFP1_CR")])
names(tempA)<-c("Index", "Year")
tempA$group<-"M1 with Calc. TFP"
tempA$col<-NOAALightBlue

TFP2_CR$Year<-rownames(TFP2_CR)
tempB<-data.frame(TFP2_CR[,names(TFP2_CR) %in% c("Year", "TFP2_CR")])
names(tempB)<-c("Index", "Year")
tempB$group<-"M2 with Calc. TFP"
```

```r
tempB$col<-NOAADarkBlue

TFP1E_CR$Year<-rownames(TFP1E_CR)
tempC<-data.frame(TFP1E_CR[,names(TFP1E_CR) %in% c("Year", "TFP1_CR")])
names(tempC)<-c("Index", "Year")
tempC$group<-"M1 with Sum. TFP"
tempC$col<-"#a66600"

TFP2E_CR$Year<-rownames(TFP2E_CR)
tempD<-data.frame(TFP2E_CR[,names(TFP2E_CR) %in% c("Year", "TFP2_CR")])
names(tempD)<-c("Index", "Year")
tempD$group<-"M2 with Sum. TFP"
tempD$col<-"#a60000"

temp0<-rbind.data.frame(tempA, tempB)#, tempC, tempD)
rownames(temp0)<-NULL
temp0$col<-as.factor(temp0$col)

plot2line(temp0, Year = temp0$Year, Index=temp0$Index, col = temp0$col, group = temp0$group,
                    NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```