Productivity Index - Output

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

2.0.2 Data requirements

We need time series data for the value of all species (V_t; e.g., Total), value of all species in a category (i) ($V_{i=1}$; e.g., Finfish), value of each species in a category (i) ($V_{i=1,s=n}$; e.g., Salmon and Summer Flounder), quanity of all species in a category (i) (in lbs, $Q_{i=1}$; e.g., Finfish and others), and the quantity of each species in a category (i) ($Q_{i=1,s=n}$; e.g., Salmon and Flounder):

2.0.2.1 Edit Data

Here we summate the category and total V because there may be instances where these values may not be the sum of their parts (though they are here). The caluclation Price Index aims to deal with this potiental issue.

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

	Q1_1Salmon	Q1_2Cod	Q2_1Shrimp	Q2_2Clam	Q1_3Flounder	Q1_4SeaBass	QE0_0Total	QE1_0Fi
2007	NA	2000	100	150	NA	1000	3250	
2008	NA	1900	120	160	NA	1200	3380	
2009	NA	2000	110	140	NA	900	3150	
2010	20	2500	90	NA	NA	NA	2610	
2011	10	2400	80	NA	NA	NA	2490	
2012	12	2300	100	NA	NA	NA	2412	
2013	11	2000	100	140	NA	1000	3251	
2014	11	2300	110	110	NA	900	3431	
2015	10	2400	90	130	NA	1000	3630	
2016	15	2200	100	160	NA	1100	3575	

2.0.2.2 The nameing conventions of the column names.

For example, in "V1_0Finfish":

- "V"... refers to the variable represented in the column (here V = "Value")
- ..."1"... refers to the category index (here, = Finfish)
- ..."_"... 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
- ..."Finfish" is purely descriptive (here the name of the category), so you can follow along with what is happening!

Similarly for "Q2_2Clam":

- "Q"... refers to the variable represented in the column (here Q = "Quantity")
- ..."2"... refers to the category index (here, = Shellfish)
- ..."_"... is simply a seperator in the title
- ..."2".. refers to the index of the species, such that this organism happens to be the second species of this category.
- ..."Clams" is purely descriptive (here the name of the species), so you can follow along with what is happening!

We can do the structuring work in a function

This funciton standardizes the length of the category or species numbers e.g.,(numbers of 33, 440, and 1 are converted to 033, 440, and 001)

2.0.3 Lets get started

In most of the following examples, we will just focus on the finfish (i=1) side of the equation.

```
ii<-1 #The category index value
warnings.list<-list() #save issues
baseyr<-2010
```

Here I am just going to do some housekeeping:

```
    #If data are missing by the below percentage, remove data
    PercentMissingThreshold<-0.50


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

    NameBaseTotal<-substr(x = sort(names(temp)[grep(x = names(temp),
                                                    pattern = "0Total")], decreasing = T)[1],
                          start = 2, stop = nchar(sort(names(temp)[grep(x = names(temp),
                                                                        pattern = "0Total")], decreasing =

    VColumns<-grep(pattern = paste0("V", ii,"_"),
                   x = substr(x = names(temp),
                              start = 1,
                              stop = (2+nchar(ii))))

  NameBasecategory<-substr(start = 2,
                           stop = nchar(names(temp)[VColumns[(grepl(
                             pattern = paste0("V", ii,"_",
                                              numbers0(x = c(0, length(VColumns)-1))[1]),
                             x = names(temp)[VColumns]))]]),
                           x = names(temp)[VColumns[(grepl(
                             pattern = paste0("V", ii,"_",
                                              numbers0(x = c(0, length(VColumns)-1))[1]),
                             x = names(temp)[VColumns]))]])

  VColumns<-VColumns[!(grepl(pattern = paste0("V", ii,"_",
                                              numbers0(x = c(0, length(VColumns)-1))[1]),
                             x = names(temp)[VColumns]))]
```

2.0.4 Remove any V and Q data where V column has less data than the specifed *percentmissingthreshold*

```
VColumns0<-VColumns
QColumns0<-QColumns<-which(names(temp) %in%
                           paste0("Q", substr(x = names(temp)[VColumns],
                                              start = 2,
                                              stop = nchar(names(temp)[VColumns]))))
for (i in 1:length(VColumns)) {

  #if the percent missing is less in V or Q columns for a species than the percentmissingtrheshold, we
  if (sum(is.na(temp[VColumns[i]]))/nrow(temp) > PercentMissingThreshold | #V
      sum(is.na(temp[QColumns[i]]))/nrow(temp) > PercentMissingThreshold ) {#Q

    names(temp)[VColumns[i]]<-paste0("REMOVED_", names(temp)[VColumns[i]])
    VColumns0<-VColumns0[!(VColumns0 %in% VColumns[i])]
    names(temp)[QColumns[i]]<-paste0("REMOVED_", names(temp)[QColumns[i]])
    QColumns0<-QColumns0[!(QColumns0 %in% QColumns[i])]
```

```
  }
}

VColumns<-names(temp)[VColumns0]
QColumns<-names(temp)[QColumns0]
```

	Q1_1Salmon	Q1_2Cod	Q2_1Shrimp	Q2_2Clam	REMOVED_Q1_3Flounder	Q1_4SeaBass	QE0_0Tot
2007	NA	2000	100	150	NA	1000	325
2008	NA	1900	120	160	NA	1200	338
2009	NA	2000	110	140	NA	900	315
2010	20	2500	90	NA	NA	NA	261
2011	10	2400	80	NA	NA	NA	249
2012	12	2300	100	NA	NA	NA	241
2013	11	2000	100	140	NA	1000	325
2014	11	2300	110	110	NA	900	343
2015	10	2400	90	130	NA	1000	363
2016	15	2200	100	160	NA	1100	357

2.0.5 Caluclate Catagory Sums of V and Q

Because we removed some columns for not meeting a perecent missing threshold and those columns will not be used at all in any part of the analysis, we need to calculate the totals of V and Q for the catagories and the fishery as a whole.

```
names(temp)[grep(pattern = NameBasecategory, x = names(temp))]<-
  paste0("REMOVED_",
         names(temp)[grep(pattern = NameBasecategory, x = names(temp))])

  # Q
  temp.q<-temp[,grepl(pattern = paste0("Q", ii), x = substr(names(temp), start = 1, stop = 2)) ]
  temp.q<-data.frame(temp.q)
  if (ncol(temp.q)>1) {
    temp.q<-rowSums(temp.q, na.rm = T)
  }
  temp[ncol(temp)+1]<-temp.q
  names(temp)[ncol(temp)]<-paste0("QE",NameBasecategory)

  # V
  temp.v<-temp[,grepl(pattern = paste0("V", ii), x = substr(names(temp), start = 1, stop = 2)) ]
  temp.v<-data.frame(temp.v)
  if (ncol(temp.v)>1) {
    temp.v<-rowSums(temp.v, na.rm = T)
  }
  temp[ncol(temp)+1]<-temp.v
  names(temp)[ncol(temp)]<-paste0("V",NameBasecategory)
```

	REMOVED_QE1_0Finfish	REMOVED_V1_0Finfish	QE1_0Finfish	V1_0Finfish
2007	3000	3800	3000	2800
2008	3100	4020	3100	2820
2009	2900	3910	2900	3010
2010	2520	3190	2520	3190
2011	2410	3280	2410	3280

	REMOVED_QE1_0Finfish	REMOVED_V1_0Finfish	QE1_0Finfish	V1_0Finfish
2012	2312	3150	2312	3150
2013	3011	4080	3011	3080
2014	3211	4270	3211	3370
2015	3410	4700	3410	3700
2016	3315	4480	3315	3380

2.0.6 Price for each species ($P_{s,i,t}$; e.g., Salmon and Flounder)

We first measure output price for each species in each of the categories (e.g., Finfish & Others and Shellfish) using detailed landings time series data on value ($) and pounds (lbs).

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

$$P_{s,i,t} = V_{s,i,t}/Q_{s,i,t}$$

where:

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

Here we calculate the price for each species

```
    # Find which columns in this table are price Columns - we will need this for later
    PColumns<-paste0("P", substr(x = VColumns,
                                 start = 2,
                                 stop = nchar(VColumns)))

#####Price for each species#####
tempP<-data.frame(data = rep_len(x = NA, length.out = nrow(temp)))
for (c in 1:length(VColumns)) {

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

    Q0<-temp[,names(temp) %in% paste0("Q", NameBase)]
    V0<-temp[,names(temp) %in% paste0("V", NameBase)] #to make sure its the same column
  tempP[,c]<-V0/Q0
  names(tempP)[c]<-paste0("P", NameBase ) #name the column
}

  tempP<-as.matrix(tempP)
  tempP[tempP %in% Inf]<-NA
  tempP<-data.frame(tempP)
  temp<-cbind.data.frame(temp, data.frame(tempP))
```

	P1_1Salmon	P1_2Cod	P1_4SeaBass
1	NA	1.400000	NA
2	NA	1.421053	0.1000000
3	NA	1.450000	0.1222222

	P1_1Salmon	P1_2Cod	P1_4SeaBass
4	5.00000	1.200000	NA
5	10.00000	1.291667	NA
6	12.50000	1.260870	NA
7	16.36364	1.400000	0.1000000
8	15.45455	1.391304	NA
9	20.00000	1.458333	NA
10	12.00000	1.454546	NA

There may be instances where price cannot (or should not) be calculated because there is no or too few Q or V data for that species in a year or ever. The next goal will be to calculate the price change, so we need to have a value in there that won't show change. If we left a 0 in the spot, then the price change from 0 to the next year would be huge and misrepresented on the index. To avoid this, we have to deal with four senarios:

2.0.6.1 1. If there are instances for a species where there are too few pairs of V and/or Q are completely missing from the timeseries or where a percent of V is missing from the timeseries, we will remove the offending price columns entierly, so they don't influence the downstream price change or price index calculations.

Let's say here that if 50% of the data is missing in a given $V_{s,i,t}$, don't calculate that species $P_{s,i,t}$

```
#Find which columns in this table are price Columns

cc<-c() #Empty
for (c in 1:length(VColumns)) {

  #If price could never be caluclated at any point in the timeseries (is 0/NaN/NA) for a column (c)
  #Remove the column from the analysis.
  #We will not be removing the column from the data, but simply remove it from the varaible "PColumns"
  if (#sum(temp[,PColumns[c]] %in% c(0, NA, NaN)) %in% nrow(temp) |
      sum(temp[,PColumns[c]] %in% c(0, NA, NaN))/nrow(temp) > PercentMissingThreshold) {
    cc<-c(cc, c)#Collect offending columns
  }

}

if (length(cc)>0){
  PColumns<-PColumns[-cc]
  # VColumns<-VColumns[-cc]
  # QColumns<-QColumns[-cc]
}
```

	P1_1Salmon	P1_2Cod
2007	NA	1.400000
2008	NA	1.421053
2009	NA	1.450000
2010	5.00000	1.200000
2011	10.00000	1.291667
2012	12.50000	1.260870
2013	16.36364	1.400000
2014	15.45455	1.391304
2015	20.00000	1.458333

	P1_1Salmon	P1_2Cod
2016	12.00000	1.454546

$$where \begin{cases} if: V_{i,t=1} = 0, then: V_{i,t=1} = V_{i,t=1+1...} \\ if: V_{i,t\neq 1} = 0, then: V_{i,t} = V_{i,t-1} \end{cases}$$

2.0.6.2 2. If the first value of P is 0 in a timeseries, we let the next available non-zero value of P in the timeseries inform the past.

```
for (c in 1:length(PColumns)) {

  #If the first value of the timeseries of this column (c) is 0/NaN/NA
  #Change the first value (and subsequent 0/NaN/NA values) to the first available non-0/NaN/NA value
  if (temp[1,PColumns[c]] %in% c(0, NA, NaN)) {
    findfirstvalue<-temp[which(!(temp[,PColumns[c]]  %in% c(0, NA, NaN))),
                         PColumns[c]][1]
    temp[1,PColumns[c]]<-findfirstvalue
  }
}
```

	P1_1Salmon	P1_2Cod
2007	5.00000	1.400000
2008	NA	1.421053
2009	NA	1.450000
2010	5.00000	1.200000
2011	10.00000	1.291667
2012	12.50000	1.260870
2013	16.36364	1.400000
2014	15.45455	1.391304
2015	20.00000	1.458333
2016	12.00000	1.454546

2.0.6.3 3. If there is a value in the middle of P's timeseries that is 0, we let the most recent past available non-zero of P in the timeseries inform the future.

```
#Find which columns in this table are price Columns
for (c in 1:length(PColumns)) {

  #If a middle value of the timeseries of this column (c) is 0/NaN/NA
  #Change the currently 0/NaN/NA value to the previous available non-0/NaN/NA value
  if (sum(temp[,PColumns[c]] %in% c(0, NA, NaN))>0) {
    troublenumber<-which(temp[,PColumns[c]] %in% c(0, NA, NaN))
    for (r in 1:length(troublenumber)){
      findlastvalue<-temp[troublenumber[r]-1, PColumns[c]][1]
      temp[troublenumber[r],PColumns[c]]<-findlastvalue
    }
  }
}
```

	P1_1Salmon	P1_2Cod
2007	5.00000	1.400000
2008	5.00000	1.421053
2009	5.00000	1.450000
2010	5.00000	1.200000
2011	10.00000	1.291667
2012	12.50000	1.260870
2013	16.36364	1.400000
2014	15.45455	1.391304
2015	20.00000	1.458333
2016	12.00000	1.454546

2.0.7 Value of species $VV_{i,t}$ where P was able to be calculated

$R_{i,t}$, defined and discussed in the subsequent step, will need to sum to 1 across all species in a category. Therefore, you will need to sum a new total of $V_{i,t}$ (called $VV_{i,t}$) for the category using only values for species that were used to calculate $P_{i,t}$ (called $V_{s,i,t,available}$).

$$VV_{i,t} = \sum_{s=1}^{n}(V_{s,i,t,available})$$

where:

- $VV_{i,t}$ is the new total of $V_{i,t}$ (called $VV_{i,t}$) for the category using only values for species that were used to calculate $P_{i,t}$
- $V_{s,i,t,available}$ are the $V_{s,i,t}$ where P were able to be calculated

```
VVColumns<-paste0("V", substr(x = PColumns, start = 2, stop = nchar(PColumns)))

temp0<-data.frame(temp[,names(temp) %in% VVColumns],
                  rowSums(temp[,names(temp) %in% VVColumns], na.rm = T))
names(temp0)[ncol(temp0)]<-paste0("VV",NameBasecategory)
temp0<-data.frame(temp0)
temp[ncol(temp)+1]<-temp0[ncol(temp0)]

temp0 %>%
    knitr::kable(row.names = T, booktabs = T)
```

	V1_1Salmon	V1_2Cod	VV1_0Finfish
2007	NA	2800	2800
2008	NA	2700	2700
2009	NA	2900	2900
2010	100	3000	3100
2011	100	3100	3200
2012	150	2900	3050
2013	180	2800	2980
2014	170	3200	3370
2015	200	3500	3700
2016	180	3200	3380

2.0.8 Revenue Share for each species ($R_{s,i,t}$; e.g., Salmon and Flounder)

$$R_{s,i,t} = V_{s,i,t}/VV_{i,t}$$

where:

- $R_{s,i,t}$ is the revenue share per individual species (s), category (i), for each year (t)
- $V_{s,i,t}$ is the value ($) per individual species (s), category (i), for each year (t)

Here we divide $V_{s,i,t}$ by $VV_{i,t}$ because $VV_{i,t}$ only includes species used to calculate $V_{s,i,t}$ as per the above price calculations.

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

  #for renaming the columns
    NameBase<-substr(start = 2,
                     stop = nchar(PColumns[c]),
                     x = PColumns[c])

  V<-(temp[,names(temp) %in% paste0("VV", NameBasecategory)])  # sum of V where P was calculated
  V0<-temp[,names(temp) %in% paste0("V", NameBase)] #V of species; to make sure its the same column
  tempR[,c]<-V0/V
  names(tempR)[c]<-paste0("R", NameBase ) #name the column
}

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

	R1_1Salmon	R1_2Cod
1	NA	1.0000000
2	NA	1.0000000
3	NA	1.0000000
4	0.0322581	0.9677419
5	0.0312500	0.9687500
6	0.0491803	0.9508197
7	0.0604027	0.9395973
8	0.0504451	0.9495549
9	0.0540541	0.9459459
10	0.0532544	0.9467456

As an additional check, let's make sure that each row sums to 1.

	x
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1

	x
10	1

2.0.9 Price Changes for each species ($PC_{s,i,t}$ aka $\Delta ln(P_{s,i,t})$; e.g., Salmon and Flounder)

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

Such that:

category's (i) Price Change = $ln(\frac{P_{i,t}}{P_{i,t-1}})$

category's (i) Price Change for each species (s) = $\frac{R_{s,i,t}+R_{s,i,t-1}}{2}$

category's (i) Revenue Share for each species (s) = $ln(\frac{P_{s,i,t}}{P_{s,i,t-1}}$

Which can be adapted to this function/macro:

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

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(PColumns[c]),
                     x = PColumns[c])

  # Calculate
  P0<-temp[, names(temp) %in% paste0("P", NameBase)]
  R0<-temp[, names(temp) %in% paste0("R", 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)
temp[ncol(temp)+1]<-rowSums(tempPC, na.rm = T)
names(temp)[ncol(temp)]<-paste0("PC", NameBasecategory)
```

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

	PC1_1Salmon	PC1_2Cod
2007	0.0000000	0.0000000
2008	NA	0.0149257
2009	NA	0.0201657
2010	NA	-0.1861897
2011	0.0220102	0.0712743
2012	0.0089738	-0.0231613
2013	0.0147572	0.0989356
2014	-0.0031679	-0.0058852
2015	0.0134715	0.0445941

	PC1_1Salmon	PC1_2Cod
2016	-0.0274080	-0.0024612

And here is the summed ($\sum$) Price Change for the category:

	Other...	R1_1Salmon	R1_2Cod	PC1_1Salmon	PC1_2Cod	PC1_0Finfish
2007	...	NA	1.0000000	0.0000000	0.0000000	0.0000000
2008	...	NA	1.0000000	NA	0.0149257	0.0149257
2009	...	NA	1.0000000	NA	0.0201657	0.0201657
2010	...	0.0322581	0.9677419	NA	-0.1861897	-0.1861897
2011	...	0.0312500	0.9687500	0.0220102	0.0712743	0.0932846
2012	...	0.0491803	0.9508197	0.0089738	-0.0231613	-0.0141875
2013	...	0.0604027	0.9395973	0.0147572	0.0989356	0.1136927
2014	...	0.0504451	0.9495549	-0.0031679	-0.0058852	-0.0090532
2015	...	0.0540541	0.9459459	0.0134715	0.0445941	0.0580655
2016	...	0.0532544	0.9467456	-0.0274080	-0.0024612	-0.0298692

2.0.10 Price Index for the each category (PI_t)

We calculate the price index first by comparing by multiplying the previous years PI_{t-1} by that year's price change PC_t, where the PI of the first year $PI_{t=firstyear} = 1$

$$PI_t = PI_{t-1} * exp(ln(\frac{P_{i,t}}{P_{i,t-1}})) = PI_{t-1} * exp(PC_t)$$

Where

$$PI_{i,t_{firstyear}} = 1$$

```
#Note that the first row of this column is = 1
tempPI1<-data.frame(c(1, rep_len(x = NA, length.out = nrow(temp)-1)))
rownames(tempPI1)<-rownames(temp)

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

  # Calculate
for (t in 2:length(tempPI1)){  #Since the first row is defined, we need to start at the second row
    tempPI1[t]<-tempPI1[t-1]*exp(PC0[t])
}
```

Then, to change the price (calulated later) into base year dollars, we use the following equation:

$$PI_t = PI_t / PI_{t=baseyear}$$

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

```
tempPI2<-tempPI1/tempPI1[rownames(tempPI1) %in% baseyr,]
```

	tempPI1	tempPI2
2007	1.015038	NA

	tempPI1	tempPI2
2008	NA	NA
2009	NA	NA
2010	NA	NA
2011	NA	NA
2012	NA	NA
2013	NA	NA
2014	NA	NA
2015	NA	NA
2016	NA	NA

Which can be summarized in this function:

```
print(PriceIndex)
```

```
## function(temp, BaseColName, baseyr) {
##   ###Price Index for the entire commercial fishery ($PI_t$)
##
##   # We calculate the price index first by comparing by multiplying the previous years $PI_{t-1}$ by
##   # $$PI_t = PI_{t-1}*exp(ln(\frac{P_{i,t}}{P_{i,t-1}})) = PI_{t-1}*exp(PC_{t})$$
##   # Where
##   # $$PI_{i, t_{first year}} = 1$$
##
##   #Note that the first row of this column is = 1
##   tempPI1<-c(1, rep_len(x = NA, length.out = nrow(temp)-1))
##
##   PC0<-temp[,names(temp) %in% paste0("PC", BaseColName)] #this is equal to ln(P_it/P_it-1)
##
##   # Calculate
##   for (t in 2:length(tempPI1)){  #Since the first row is defined, we need to start at the second row
##     tempPI1[t]<-tempPI1[t-1]*exp(PC0[t])
##   }
##
##   tempPI1<-data.frame(tempPI1)
##   rownames(tempPI1)<-rownames(temp)
##
##   # Then, to change the price (calulated later) into base year dollars, we use the following equatio
##   # $$PI_{t} = PI_{t}/PI_{t = baseyear}$$
##   # In this example, we'll decide that the base year is `r baseyr`, for whatever reason. Notice that
##
##   tempPI2<-tempPI1/tempPI1[rownames(tempPI1) %in% baseyr,]
##
##   tempPI<-data.frame(tempPI2)
##   names(tempPI)<-paste0("PI", BaseColName)
##
##   return(tempPI)
## }
```

And we add the PI to the data

```
tempPI<-PriceIndex(temp, BaseColName = NameBasecategory, baseyr)
temp[ncol(temp)+1]<-(tempPI)
names(temp)[ncol(temp)]<-paste0("PI", NameBasecategory)
```

	PI1_0Finfish
2007	1.163111
2008	1.180602
2009	1.204651
2010	1.000000
2011	1.097774
2012	1.082309
2013	1.212628
2014	1.201699
2015	1.273542
2016	1.236065

2.0.11 Implicit Quantity/Output for each category ($Q_{i,t}$; Finfish & others and Shellfish)

Note here that all columns of V are being used, despite having been removed earlier in the analysis when PI could not be calculated and PI columns have functionally been removed from the analysis.

$$Q_{i,t} = V_{i,t}/PI_{i,t}$$

```
temp[,ncol(temp)+1]<-temp[,names(temp) %in% paste0("V", NameBasecategory)]/
  temp[,names(temp) %in% paste0("PI", NameBasecategory)]

names(temp)[ncol(temp)]<-paste0("Q", NameBasecategory)
```

	Other...	PC1_1Salmon	PC1_2Cod	PC1_0Finfish	PI1_0Finfish	Q1_0Finfish
2007	...	0.0000000	0.0000000	0.0000000	1.163111	2407.337
2008	...	NA	0.0149257	0.0149257	1.180602	2388.613
2009	...	NA	0.0201657	0.0201657	1.204651	2498.649
2010	...	NA	-0.1861897	-0.1861897	1.000000	3190.000
2011	...	0.0220102	0.0712743	0.0932846	1.097774	2987.864
2012	...	0.0089738	-0.0231613	-0.0141875	1.082309	2910.443
2013	...	0.0147572	0.0989356	0.1136927	1.212628	2539.938
2014	...	-0.0031679	-0.0058852	-0.0090532	1.201699	2804.362
2015	...	0.0134715	0.0445941	0.0580655	1.273542	2905.283
2016	...	-0.0274080	-0.0024612	-0.0298692	1.236065	2734.484

2.0.12 Analysis Warnings Checks

2.0.12.1 1. When back calculated, V_t did not equal $P_t * Q_t$

$$V_i = P_t * Q_i$$

```
  temp0<-temp[names(temp) %in% c(paste0("Q",NameBasecategory),
                                 paste0("PI",NameBasecategory),
                                 paste0("V",NameBasecategory))]

  temp0[,(ncol(temp0)+1)]<-temp0[,paste0("Q",NameBasecategory)]*temp0[,paste0("PI",NameBasecategory)]
  names(temp0)[ncol(temp0)]<-paste0("V", NameBasecategory, "_Check")
```

```
  if (sum(temp0[,paste0("V", NameBasecategory, "_Check")] %in%
          temp0[,paste0("V", NameBasecategory)]) == nrow(temp0)) {
    warnings.list[length(warnings.list)+1]<-"When back calculated, V_{i,t} did not equal PI_{i,t} * Q_{
    print("When back calculated, V_{i,t} did not equal PI_{i,t} * Q_{i,t}")
  }
```

	V1_0Finfish	PI1_0Finfish	Q1_0Finfish	V1_0Finfish_Check
2007	2800	1.163111	2407.337	2800
2008	2820	1.180602	2388.613	2820
2009	3010	1.204651	2498.649	3010
2010	3190	1.000000	3190.000	3190
2011	3280	1.097774	2987.864	3280
2012	3150	1.082309	2910.443	3150
2013	3080	1.212628	2539.938	3080
2014	3370	1.201699	2804.362	3370
2015	3700	1.273542	2905.283	3700
2016	3380	1.236065	2734.484	3380

2.0.12.2 2. When back calculated, Q_t did not equal V_t/PI_t

$$Q_{i,t} = V_t/PI_{i,t}$$

```
  temp0[,(ncol(temp0)+1)]<-temp0[,paste0("V",NameBasecategory)]/temp0[,paste0("PI",NameBasecategory)]
  names(temp0)[ncol(temp0)]<-paste0("Q", NameBasecategory, "_Check")

  if (sum(temp0[,paste0("Q", NameBasecategory, "_Check")] %in%
          temp0[,paste0("Q", NameBasecategory)]) == nrow(temp0)) {
    warnings.list[length(warnings.list)+1]<-"When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,
    print("When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}")
  }
```

```
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
```

	V1_0Finfish	PI1_0Finfish	Q1_0Finfish	V1_0Finfish_Check	Q1_0Finfish_Check
2007	2800	1.163111	2407.337	2800	2407.337
2008	2820	1.180602	2388.613	2820	2388.613
2009	3010	1.204651	2498.649	3010	2498.649
2010	3190	1.000000	3190.000	3190	3190.000
2011	3280	1.097774	2987.864	3280	2987.864
2012	3150	1.082309	2910.443	3150	2910.443
2013	3080	1.212628	2539.938	3080	2539.938
2014	3370	1.201699	2804.362	3370	2804.362
2015	3700	1.273542	2905.283	3700	2905.283
2016	3380	1.236065	2734.484	3380	2734.484

2.1 Redo Analysis for Shellfish

Pretending that we also did all of that work we just did for FinFish and Others for Shellfish and two species of shellfish, I'll use a function called "species.cat.level" that I will print out for you after this example:

```
ii<-2 #The category index value

tempS<-species.cat.level(temp, ii, baseyr, maxyr, minyr,
                         PercentMissingThreshold = PercentMissingThreshold,
                         warnings.list = warnings.list)
temp<-cbind.data.frame(temp, tempS[[1]])
warnings.list<-tempS[[2]]
###Remove duplicate columns
temp<-temp[, !(grepl(pattern = "\\.[0-9]+", x = names(temp)))]
temp<-temp[, !(grepl(pattern = "\\.[0-9]+", x = names(temp)))]
```

What does the Shellfish data look like?

	VV2_0Shellfish	R2_1Shrimp	R2_2Clam	PC2_0Shellfish	PI2_0Shellfish	Q2_0Shellfish
2007	1800	0.4444444	0.5555556	0.0000000	1.045118	1722.2938
2008	2200	0.4545455	0.5454545	0.0831894	1.135779	1936.9959
2009	1800	0.5000000	0.5000000	-0.0893363	1.038713	1732.9131
2010	700	1.0000000	NA	-0.0379828	1.000000	700.0000
2011	900	1.0000000	NA	0.3690975	1.446429	622.2222
2012	1000	1.0000000	NA	-0.1177830	1.285714	777.7778
2013	2200	0.5454545	0.4545455	0.1408848	1.480233	1486.2529
2014	2000	0.5500000	0.4500000	-0.0384432	1.424408	1404.0923
2015	2000	0.5000000	0.5000000	0.0260098	1.461942	1368.0430
2016	2300	0.5217391	0.4782609	-0.0156266	1.439275	1598.0270

2.1.1 Value for all fisheries for species where P was able to be calculated

$R_{i,t}$, defined and discussed in the subsequent step, will need to sum to 1 across all species in a category. Therefore, you will need to sum a new total of $V_{i,t}$ (called VV_t) for the category using only values for species that were used to calculate $PI_{i,t}$.

$$VV_t = \sum_{s=1}^{n} (VV_{i,t})$$

where:

- VV_t is the new total of $V_{i,t}$ for the entire fishery using only values for species that were used to calculate $P_{i,t}$

VV1	_0Finfish VV2	_0Shellfish VV0	_0Total
2007	2800	1800	4600
2008	2700	2200	4900
2009	2900	1800	4700
2010	3100	700	3800
2011	3200	900	4100
2012	3050	1000	4050
2013	2980	2200	5180
2014	3370	2000	5370
2015	3700	2000	5700
2016	3380	2300	5680

2.1.2 Revenue Share for the entire commercial fishery (R_t)

$$R_{i,t} = V_{i,t}/V_t$$

where:

- $R_{i,t}$ is the revenue share per individual species (s), category (i), for each year (t)
- $V_{i,t}$ is the value ($) per individual species (s), category (i), for each year (t)

Here, we don't use VV_t beacause we want to expand the proportion to include all of the species caught, regardless if they were used in the price calculations.

```
names(temp)[names(temp) %in% paste0("V", NameBaseTotal)]<-paste0("REMOVED_V", NameBaseTotal)

temp0<-temp[grep(x = names(temp),
                           pattern = paste0("V[1-9]+_", NumberOfSpecies))]
temp0<-temp0[,-(grep(x = names(temp0), pattern = c("VV")))]
temp0<-temp0[,-(grep(x = names(temp0), pattern = c("REMOVED_")))]

temp[ncol(temp)+1]<-rowSums(temp0, na.rm = T)
names(temp)[ncol(temp)]<-paste0("V", NameBaseTotal)

#remove duplicates
  temp<-temp[, !(grepl(pattern = "\\.[0-9]+", x = names(temp)))]
  temp <- temp[, !duplicated(colnames(temp))]

# temp$R1_0Finfish<-temp$VV1_0Finfish/temp$VV0_0Total
# temp$R2_0Shellfish<-temp$VV2_0Shellfish/temp$VV0_0Total

temp$R1_0Finfish<-temp$V1_0Finfish/temp$V0_0Total
temp$R2_0Shellfish<-temp$V2_0Shellfish/temp$V0_0Total
```

	R1_0Finfish	R2_0Shellfish	V1_0Finfish	V2_0Shellfish	V0_0Total
2007	0.6086957	0.3913043	2800	1800	4600
2008	0.5617530	0.4382470	2820	2200	5020
2009	0.6257796	0.3742204	3010	1800	4810
2010	0.8200514	0.1799486	3190	700	3890
2011	0.7846890	0.2153110	3280	900	4180
2012	0.7590361	0.2409639	3150	1000	4150
2013	0.5833333	0.4166667	3080	2200	5280
2014	0.6275605	0.3724395	3370	2000	5370
2015	0.6491228	0.3508772	3700	2000	5700
2016	0.5950704	0.4049296	3380	2300	5680

As an additional check, let's make sure that each row sums to 1.

	x
2007	1
2008	1
2009	1
2010	1
2011	1
2012	1

	x
2013	1
2014	1
2015	1
2016	1

2.1.3 Price Changes for the entire commercial fishery (PC_t)

Measure output price changes (PC_t) for total output (Q_t) using $R_{i,t}$ and $P_{i,t}$ estimates.

$$PC_t = ln(\frac{P_t}{P_{t-1}}) = \sum_{i=1}^{n}([\frac{R_{i,t} + R_{i,t-1}}{2}] * [ln(P_{i,t}) - ln(P_{i,t-1})])$$

```
temp$PC0_0Total<-rowSums(cbind(PriceChange(R0 = temp$R1_0Finfish, P0 = temp$PI1_0Finfish),
                               PriceChange(R0 = temp$R2_0Shellfish, P0 = temp$PI2_0Shellfish)),
                         na.rm = T)
```

	Other...	VV0_0Total	V0_0Total	R1_0Finfish	R2_0Shellfish	PC0_0Total
2007	...	4600	4600	0.6086957	0.3913043	0.0000000
2008	...	4900	5020	0.5617530	0.4382470	0.0432398
2009	...	4700	4810	0.6257796	0.3742204	-0.0243177
2010	...	3800	3890	0.8200514	0.1799486	-0.1451239
2011	...	4100	4180	0.7846890	0.2153110	0.1477934
2012	...	4050	4150	0.7590361	0.2409639	-0.0378216
2013	...	5180	5280	0.5833333	0.4166667	0.1226339
2014	...	5370	5370	0.6275605	0.3724395	-0.0206491
2015	...	5700	5700	0.6491228	0.3508772	0.0464723
2016	...	5680	5680	0.5950704	0.4049296	-0.0244869

2.1.4 Price Index for the entire commercial fishery (PI_t)

We calculate the price index first by comparing by multiplying the previous years PI_{t-1} by that year's price change PC_t, where the PI of the first year $PI_{t=firstyear} = 1$

$$PI_t = PI_{t-1} * exp(ln(\frac{P_{i,t}}{P_{i,t-1}})) = PI_{t-1} * exp(PC_t)$$

Where

$$PI_{i,t_{firstyear}} = 1$$

```
tempPI<-PriceIndex(temp, BaseColName = NameBaseTotal, baseyr)
temp[ncol(temp)+1]<-(tempPI)
names(temp)[ncol(temp)]<-paste0("PI", NameBaseTotal)
```

	PI0_0Total
2007	1.134511
2008	1.184643
2009	1.156183

	PI0_0Total
2010	1.000000
2011	1.159273
2012	1.116247
2013	1.261884
2014	1.236094
2015	1.294894
2016	1.263571

2.1.5 Total Implicit Quantity/Output 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$Q0_0Total<-temp$V0_0Total/temp$PI0_0Total
```

	x
1	4054.610
2	4237.563
3	4160.242
4	3890.000
5	3605.707
6	3717.816
7	4184.220
8	4344.329
9	4401.904
10	4495.195

2.1.6 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...	R2_0Shellfish	PC0_0Total	PI0_0Total	Q0_0Total	QI0_0Total
2007	...	0.3913043	0.0000000	1.134511	4054.610	1.0423162
2008	...	0.4382470	0.0432398	1.184643	4237.563	1.0893478
2009	...	0.3742204	-0.0243177	1.156183	4160.242	1.0694709
2010	...	0.1799486	-0.1451239	1.000000	3890.000	1.0000000
2011	...	0.2153110	0.1477934	1.159273	3605.707	0.9269169
2012	...	0.2409639	-0.0378216	1.116247	3717.816	0.9557368
2013	...	0.4166667	0.1226339	1.261884	4184.220	1.0756350
2014	...	0.3724395	-0.0206491	1.236094	4344.329	1.1167940
2015	...	0.3508772	0.0464723	1.294894	4401.904	1.1315949

	Other...	R2_0Shellfish	PC0_0Total	PI0_0Total	Q0_0Total	QI0_0Total
2016	...	0.4049296	-0.0244869	1.263571	4495.195	1.1555772

2.1.7 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...	PC0_0Total	PI0_0Total	Q0_0Total	QI0_0Total	QEI0_0Total
2007	...	0.0000000	1.134511	4054.610	1.0423162	1.2452107
2008	...	0.0432398	1.184643	4237.563	1.0893478	1.2950192
2009	...	-0.0243177	1.156183	4160.242	1.0694709	1.2068966
2010	...	-0.1451239	1.000000	3890.000	1.0000000	1.0000000
2011	...	0.1477934	1.159273	3605.707	0.9269169	0.9540230
2012	...	-0.0378216	1.116247	3717.816	0.9557368	0.9241379
2013	...	0.1226339	1.261884	4184.220	1.0756350	1.2455939
2014	...	-0.0206491	1.236094	4344.329	1.1167940	1.3145594
2015	...	0.0464723	1.294894	4401.904	1.1315949	1.3908046
2016	...	-0.0244869	1.263571	4495.195	1.1555772	1.3697318

2.1.8 Solve Output portion of the equation for the Output Changes:

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

```
temp$QC0_0Total<-rowSums(cbind(PriceChange(R0 = temp$R1_0Finfish, P0 = temp$Q1_0Finfish),
                               PriceChange(R0 = temp$R2_0Shellfish, P0 = temp$Q2_0Shellfish)),
                         na.rm = T)
```

	Q0_0Total	QI0_0Total	QC0_0Total
2007	4054.610	1.0423162	0.0000000
2008	4237.563	1.0893478	0.0441588
2009	4160.242	1.0694709	-0.0184860
2010	3890.000	1.0000000	-0.0745842
2011	3605.707	0.9269169	-0.0758022
2012	3717.816	0.9557368	0.0306434
2013	4184.220	1.0756350	0.1215395
2014	4344.329	1.1167940	0.0375242
2015	4401.904	1.1315949	0.0131616
2016	4495.195	1.1555772	0.0210303

2.2 Analysis Warnings Checks

To make sure our analyses worked as inteded, let's see if we can back calculate our numbers.

We want the calcuated V to equal this check:

2.2.0.1 1. When back calculated, V_t did not equal $PI_t * Q_t$?

$$V_i = P_t * Q_i$$

```
  temp0<-temp[names(temp) %in% c(paste0("Q",NameBaseTotal),
                                 paste0("PI",NameBaseTotal),
                                        paste0("V",NameBaseTotal))]

  temp0[,(ncol(temp0)+1)]<-temp0[,paste0("Q",NameBaseTotal)]*temp0[,paste0("PI",NameBaseTotal)]
  names(temp0)[ncol(temp0)]<-paste0("V", NameBaseTotal, "_Check")

  if (sum(temp0[,paste0("V", NameBaseTotal, "_Check")] %in%
          temp0[,paste0("V", NameBaseTotal)]) == nrow(temp0))  {
  warnings.list[length(warnings.list)+1]<-"When back calculated, V_t did not equal PI_t * Q_t"
  print("When back calculated, V_t did not equal PI_t * Q_t")
  }
```

```
## [1] "When back calculated, V_t did not equal PI_t * Q_t"
```

	V0_0Total	PI0_0Total	Q0_0Total	V0_0Total_Check
2007	4600	1.134511	4054.610	4600
2008	5020	1.184643	4237.563	5020
2009	4810	1.156183	4160.242	4810
2010	3890	1.000000	3890.000	3890
2011	4180	1.159273	3605.707	4180
2012	4150	1.116247	3717.816	4150
2013	5280	1.261884	4184.220	5280
2014	5370	1.236094	4344.329	5370
2015	5700	1.294894	4401.904	5700
2016	5680	1.263571	4495.195	5680

2.2.0.2 2. When back calculated, Q_t did not equal V_t/P_t?

$$Q_{i,t} = V_t/P_{i,t}$$

```
  temp0[,(ncol(temp0)+1)]<-temp0[,paste0("V",NameBaseTotal)]/temp0[,paste0("PI",NameBaseTotal)]
  names(temp0)[ncol(temp0)]<-paste0("Q", NameBaseTotal, "_Check")

  if (sum(temp0[,paste0("Q", NameBaseTotal, "_Check")] %in%
          temp0[,paste0("Q", NameBaseTotal)]) == nrow(temp0))  {
  warnings.list[length(warnings.list)+1]<-"When back calculated, Q_t did not equal V_t/PI_t"
  print("When back calculated, Q_t did not equal V_t/PI_t")
  }
```

```
## [1] "When back calculated, Q_t did not equal V_t/PI_t"
```

	V0_0Total	PI0_0Total	Q0_0Total	V0_0Total_Check	Q0_0Total_Check
2007	4600	1.134511	4054.610	4600	4054.610
2008	5020	1.184643	4237.563	5020	4237.563
2009	4810	1.156183	4160.242	4810	4160.242
2010	3890	1.000000	3890.000	3890	3890.000
2011	4180	1.159273	3605.707	4180	3605.707
2012	4150	1.116247	3717.816	4150	3717.816
2013	5280	1.261884	4184.220	5280	4184.220
2014	5370	1.236094	4344.329	5370	4344.329
2015	5700	1.294894	4401.904	5700	4401.904
2016	5680	1.263571	4495.195	5680	4495.195

2.2.0.3 3. When back calculated, growth rate?

$$ln(Q_t/Q_{t-1}) = \sum((\frac{R_{i,t} - R_{i,t-1}}{2}) * ln(\frac{Q_{i,t}}{Q_{i,t-1}}))$$

```
names0<-c(paste0("Q",NameBaseTotal))
for (i in 1:ii) {
  names0<-c(names0,
            # names(temp)[grep(pattern = paste0("QE", i, "_", NumberOfSpecies), names(temp))
            #   [!(grep(pattern = paste0("QE", i, "_", NumberOfSpecies), names(temp))) %in%
            #                grep(pattern = paste0("REMOVED_"), names(temp))] ],
            names(temp)[grep(pattern = paste0("Q", i, "_", NumberOfSpecies), names(temp))],
            names(temp)[grep(pattern = paste0("R", i, "_", NumberOfSpecies), names(temp))])
}

temp0<-temp[,names0]

temp0[,(ncol(temp0)+1)]<-c(NA, ln(temp0[-nrow(temp0),paste0("Q",NameBaseTotal)]/
                                  temp0[-1,paste0("Q",NameBaseTotal)]))
names(temp0)[ncol(temp0)]<-"part1"

temp00<-data.frame()
for (i in 1:(ii)) {
  R0<-temp0[,grep(pattern = paste0("R", i), x = names(temp0))]
  Q0<-temp0[,grep(pattern = paste0("Q", i), x = names(temp0))]

  for (r in 2:(nrow(temp))){
    temp00[r,i]<-(((R0[r]-R0[r-1])/2) * ln(Q0[r] / Q0[r-1]) )
  }
}

temp0[,(ncol(temp0)+1)]<-rowSums(temp00)
names(temp0)[ncol(temp0)]<-"part2"

  if (sum(temp0[,"part1"] %in% temp0[,"part2"]) != nrow(temp0))  {
  warnings.list[length(warnings.list)+1]<-"When back calculated, ln(Q_t/Q_{t-1}) = did not equal sum( (
  print("When back calculated, ln(Q_t/Q_{t-1}) = did not equal sum( ( \frac{R_{i, t} - R_{i, t-1}}{2})
  }
```

```
## [1] "When back calculated, ln(Q_t/Q_{t-1}) = did not equal sum( ( \frac{R_{i, t} - R_{i, t-1}}{2})  =
```

	Q0_0Total	Q1_0Finfish	R1_0Finfish	Q2_0Shellfish	R2_0Shellfish	part1	part2
2007	4054.610	2407.337	0.6086957	1722.2938	0.3913043	NA	NA
2008	4237.563	2388.613	0.5617530	1936.9959	0.4382470	-0.0441338	0.0029407
2009	4160.242	2498.649	0.6257796	1732.9131	0.3742204	0.0184151	0.0050060
2010	3890.000	3190.000	0.8200514	700.0000	0.1799486	0.0671641	0.1117791
2011	3605.707	2987.864	0.7846890	622.2222	0.2153110	0.0758913	-0.0009251
2012	3717.816	2910.443	0.7590361	777.7778	0.2409639	-0.0306186	0.0031989
2013	4184.220	2539.938	0.5833333	1486.2529	0.4166667	-0.1181839	0.0688525
2014	4344.329	2804.362	0.6275605	1404.0923	0.3724395	-0.0375509	0.0034476
2015	4401.904	2905.283	0.6491228	1368.0430	0.3508772	-0.0131659	0.0006616
2016	4495.195	2734.484	0.5950704	1598.0270	0.4049296	-0.0209719	0.0058370

2.2.0.4 4. Missing Data

```
## [1] "Out of 5 columns, 0 of species V columns are completely empty, 1 of species Q columns are compl
```

2.2.0.5 5. Removed Data

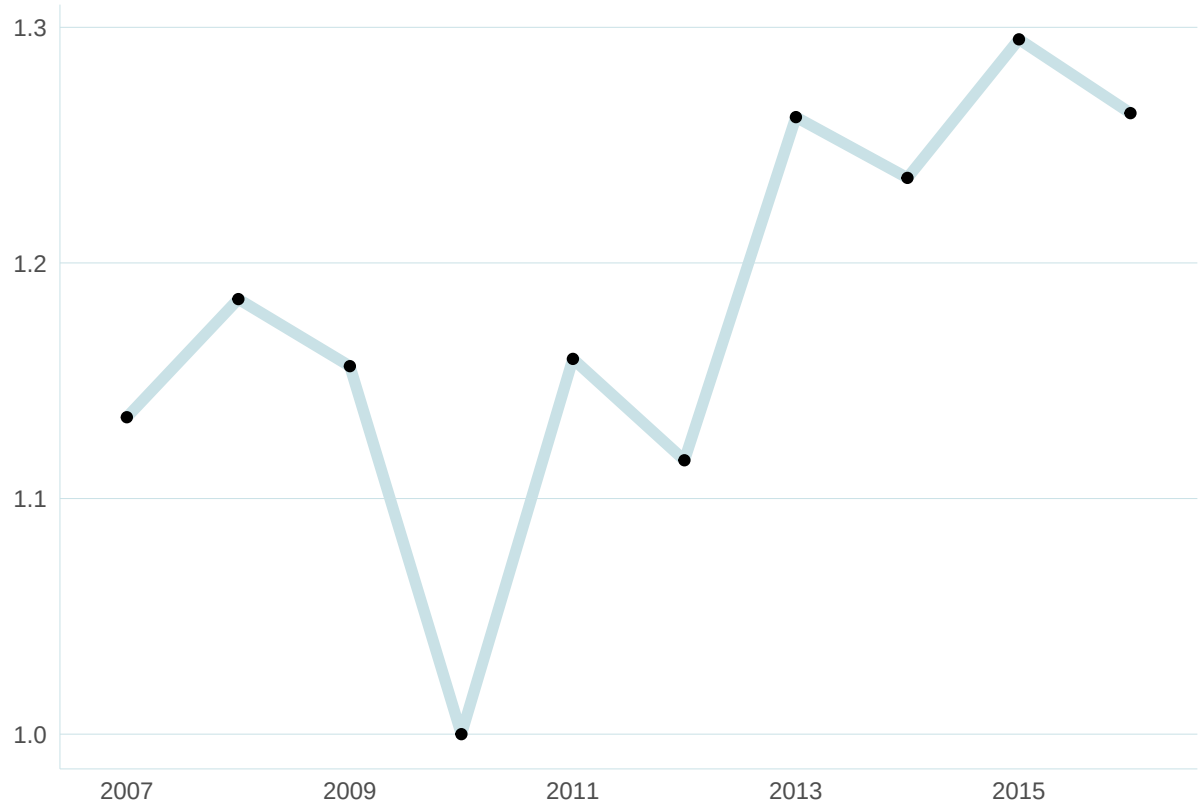
```
## [1] "Out of 6 columns, 1 of species V columns were removed, 1 of species Q columns were removed, and
```

	Q0_0Total	Q1_0Finfish	R1_0Finfish	Q2_0Shellfish	R2_0Shellfish	part1	part2
2007	4054.610	2407.337	0.6086957	1722.2938	0.3913043	NA	NA
2008	4237.563	2388.613	0.5617530	1936.9959	0.4382470	-0.0441338	0.0029407
2009	4160.242	2498.649	0.6257796	1732.9131	0.3742204	0.0184151	0.0050060
2010	3890.000	3190.000	0.8200514	700.0000	0.1799486	0.0671641	0.1117791
2011	3605.707	2987.864	0.7846890	622.2222	0.2153110	0.0758913	-0.0009251
2012	3717.816	2910.443	0.7590361	777.7778	0.2409639	-0.0306186	0.0031989
2013	4184.220	2539.938	0.5833333	1486.2529	0.4166667	-0.1181839	0.0688525
2014	4344.329	2804.362	0.6275605	1404.0923	0.3724395	-0.0375509	0.0034476
2015	4401.904	2905.283	0.6491228	1368.0430	0.3508772	-0.0131659	0.0006616
2016	4495.195	2734.484	0.5950704	1598.0270	0.4049296	-0.0209719	0.0058370

2.2.1 Graph 1: Price Index

In theory, PI should be negative slope after the baseyear and positive after the base year, but because this data was fabricated without thinking of this, we don't see that here. The index value for the base year is =1, however.

```
# A function I made to plot this pretty in ggplot2
temp <- temp[, !duplicated(colnames(temp))]
plot1line(temp, PI = temp$PI0_0Total,
            NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



2.2.2 Graph 2: Quantity Index Compare

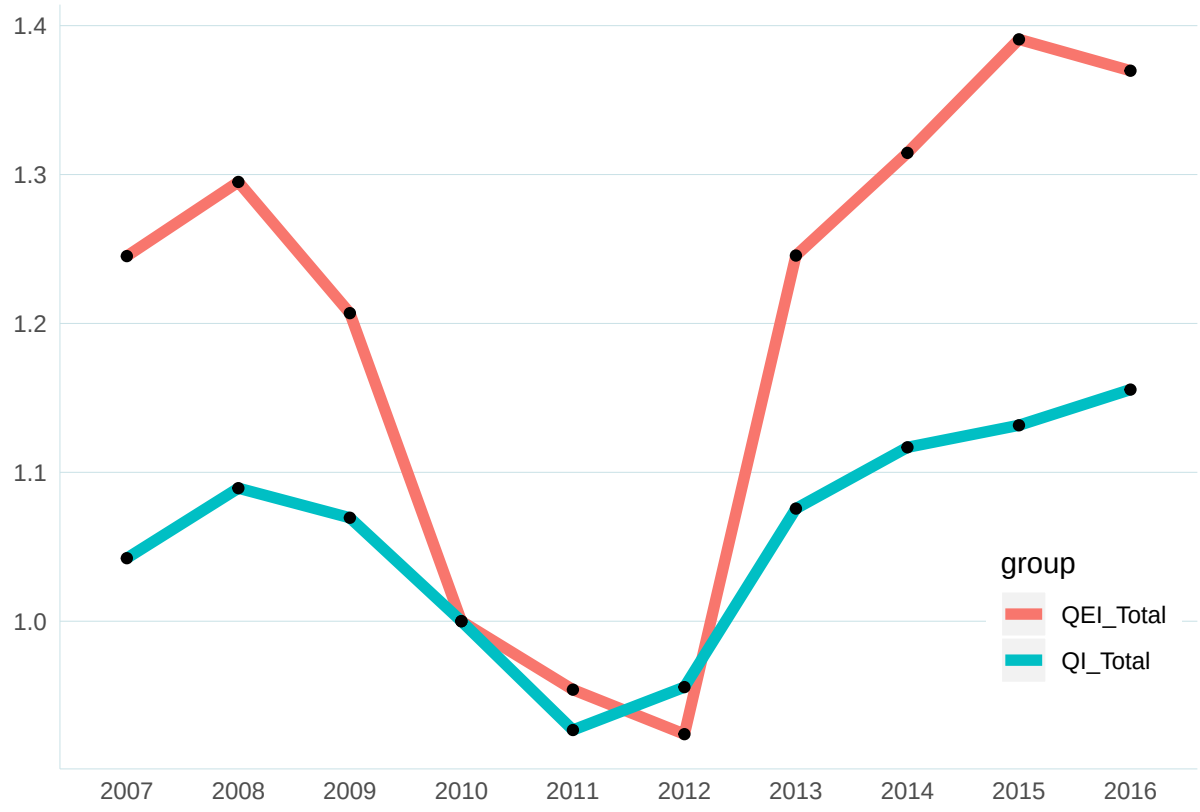
```
temp0<-temp
temp0$Year<-rownames(temp0)

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)

#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)
```



2.2.3 Graph 3: Quantity Compare

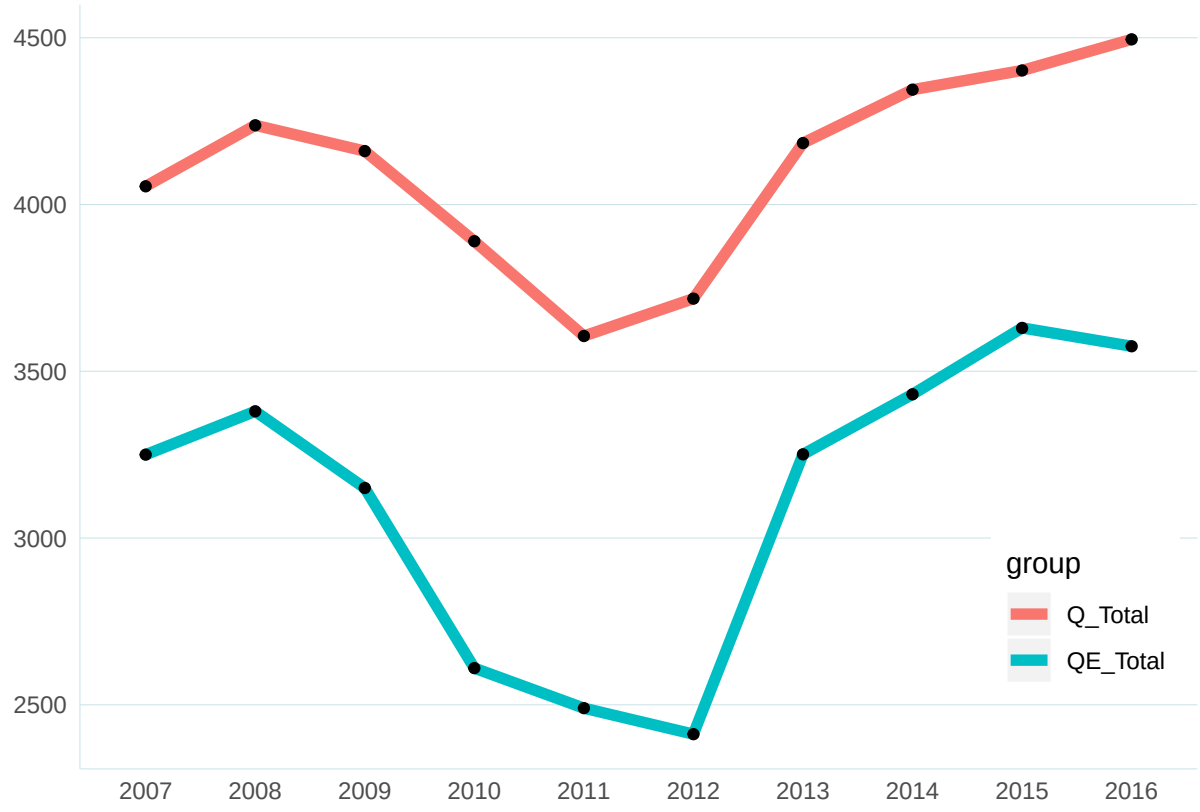
```
temp0<-temp
temp0$Year<-rownames(temp0)

tempA<-data.frame(temp0[,names(temp0) %in% c("Year", "Q0_0Total")])
names(tempA)<-c("Quantity", "Year")
tempA$group<-"Q_Total"
tempA$col<-NOAALightBlue

tempB<-data.frame(temp0[,names(temp0) %in% c("Year", "QE0_0Total")])
names(tempB)<-c("Quantity", "Year")
tempB$group<-"QE_Total"
tempB$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$Year, Index=temp0$Quantity, col = temp0$col, group = temp0$group,
                    NOAALightBlue, NOAADarkBlue, NOAADarkGrey)
```



2.3 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

2.3.1 Function to calculate the Implicit Quanity Output at Species and category Level

```
# print(species.cat.level)
```

2.3.2 Function to calculate the Implicit Quanity Output at Fishery Level

```
# print(ImplicitQuantityOutput)
```

2.3.3 A. Import and Edit data

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

2.3.4 B. Enter base year

```
baseyr<-baseyr
```

2.3.5 C. Run the function

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

2.3.6 D. Obtain the implicit quantity estimates

	rep_len.x. . . NA..length.out. . . nrow.temp..	Q1_1Salmon	Q1_2Cod	Q2_1Shrimp	Q2_2Clam	Q1_3Flounde
2007	NA	NA	2000	100	150	NA
2008	NA	NA	1900	120	160	NA
2009	NA	NA	2000	110	140	NA
2010	NA	20	2500	90	NA	NA
2011	NA	10	2400	80	NA	NA
2012	NA	12	2300	100	NA	NA
2013	NA	11	2000	100	140	NA
2014	NA	11	2300	110	110	NA
2015	NA	10	2400	90	130	NA
2016	NA	15	2200	100	160	NA

Did all of the analyses work as intended?

```
print(warnings.list0)
```

```
## [[1]]
```

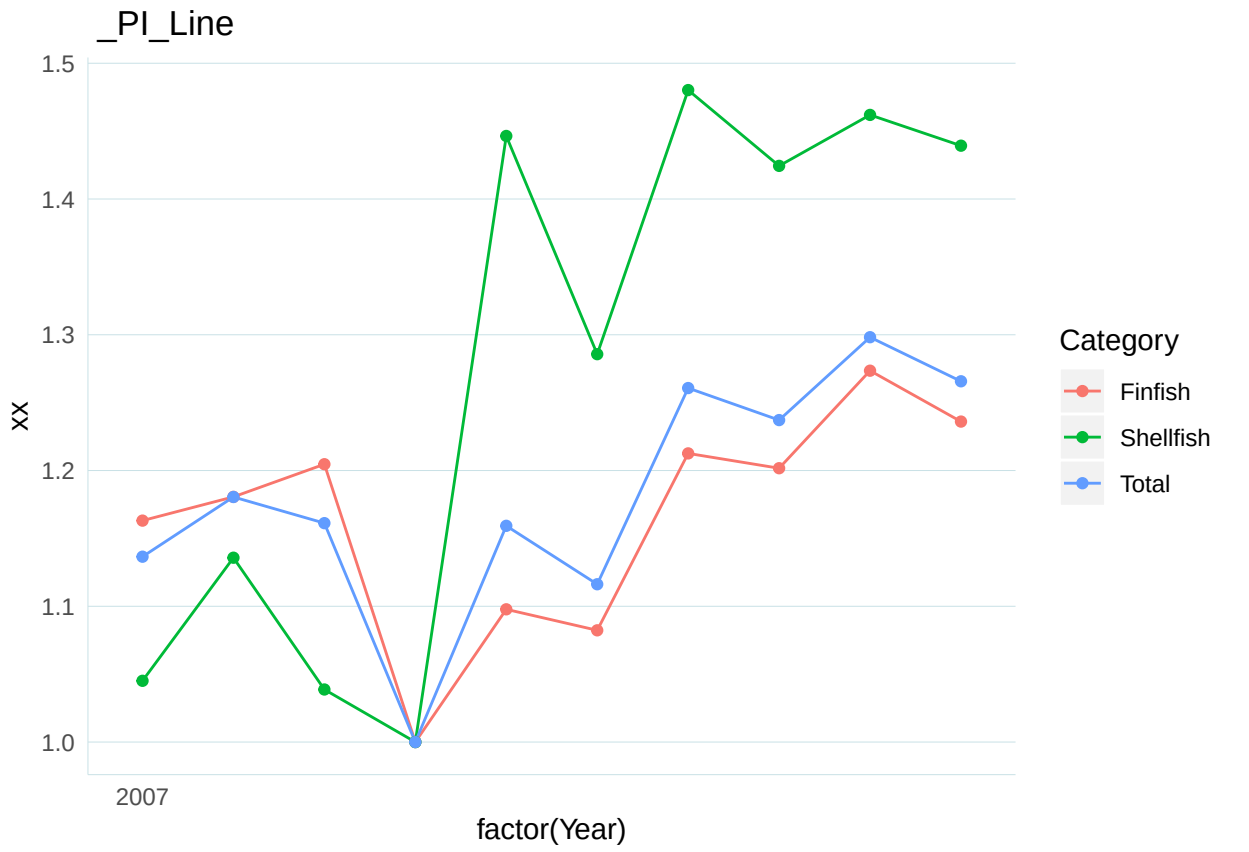
```
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
##
## [[2]]
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
##
## [[3]]
## [1] "When back calculated, Q_t did not equal V_t/PI_t"
##
## [[4]]
## [1] "When back calculated, ln(Q_t/Q_{t-1}) = did not equal sum( ( frac{R_{i, t} - R_{i, t-1}}{2})  *
##
## [[5]]
## [1] "Out of 6 columns, 0 of species V columns are completely empty, 1 of species Q columns are compl
```

2.3.7 E. Graph

2.3.7.1 Graph 1: Price Index

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

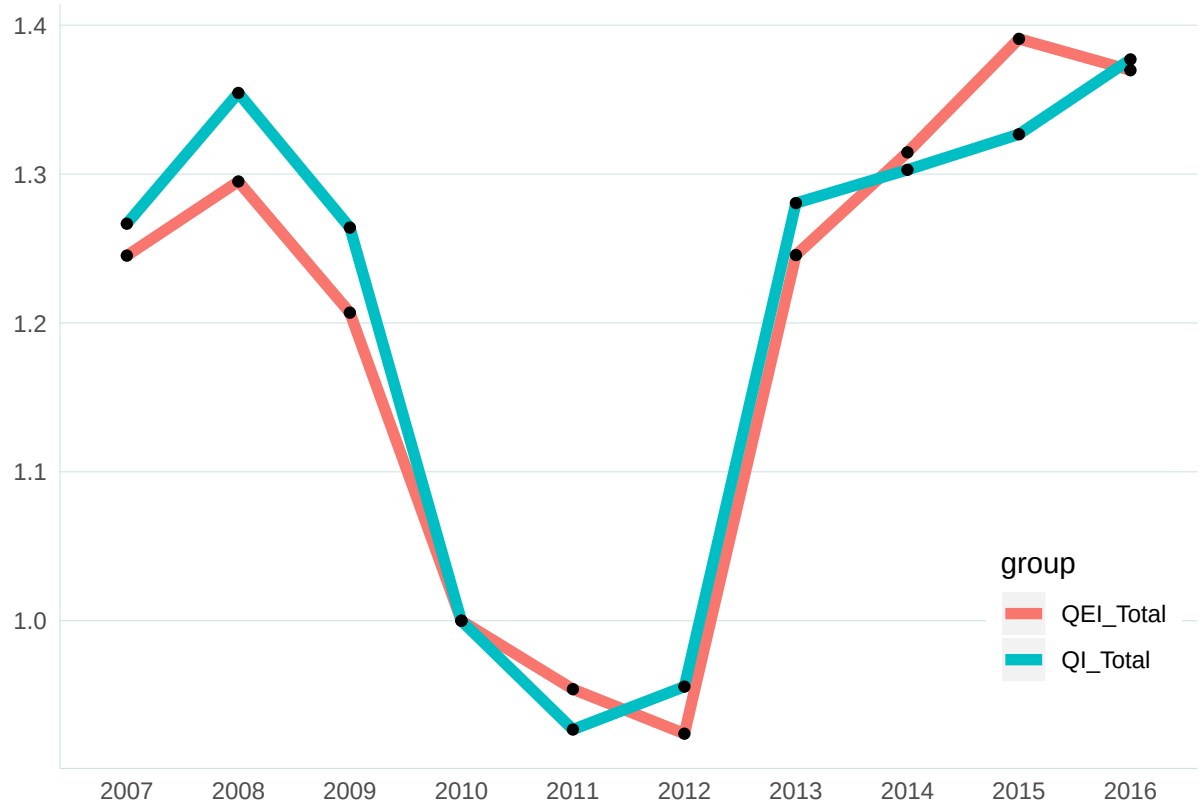
```
figures.list0$`_PI_Line`
```



2.3.7.2 Graph 2: Quantity Index Compare

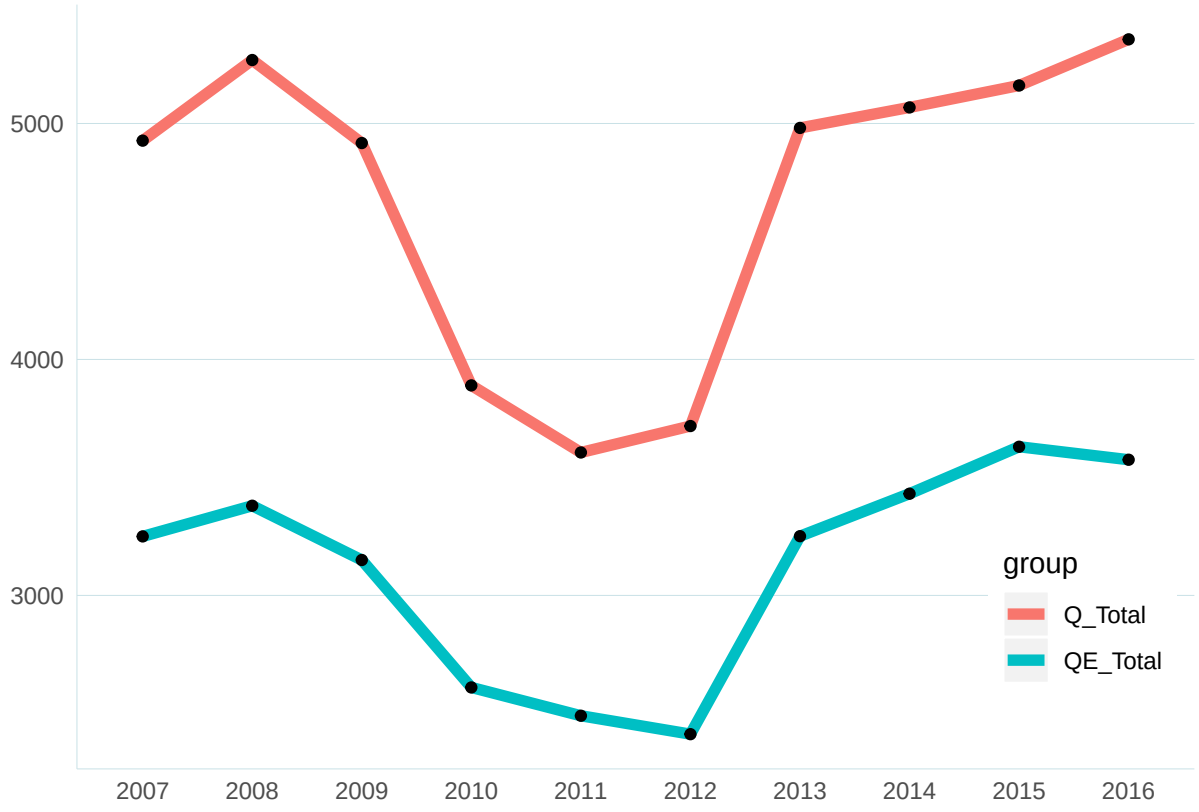
For comparison, let's recreate those graphs to make sure we are getting the same output:

```
figures.list0$`_QuantityIndexCompare`
```



2.3.7.3 Graph 3: Quantity Compare

```
figures.list0$`_QuantityCompare`
```



2.4 Practice with real data (For National Data)

2.4.1 A. Import and Edit data

Load and subset Data

```
#Load Data (This data has been edited to include category columns)
landings.data<-read.csv(file = paste0(dir.data, "landings_edited.csv"))
landings.data<-landings.data[landings.data$Year < 2018,] #FUS 2018 hasn't been published yet
landings.data<-landings.data[landings.data$State %in% unique(state.codes$NAME),]
region<-"National"
#We'll categorize by this column I already added to the data
category0 = "category.orig"
```

Summary information about the commercial dataset:

```
summary(landings.data[,c("Tsn", "Year", "State", "AFS.Name", "Pounds", "Dollars", "category.orig")])
```

```
##       Tsn              Year                 State                          AFS.Name          Pounds
##  Min.   :     0   Min.   :1950   West Florida  :10405   FINFISH **             :  1467   Min.   :
##  1st Qu.:160845   1st Qu.:1977   East Florida  : 8973   OYSTER, EASTERN        :  1187   1st Qu.:
##  Median :167674   Median :1995   New York      : 7106   SHARKS, UNCLASSIFIED **:  1169   Median :
##  Mean   :164501   Mean   :1991   California    : 6899   BLUEFISH               :  1103   Mean   :
##  3rd Qu.:169611   3rd Qu.:2008   North Carolina: 6436   SHAD, AMERICAN         :  1083   3rd Qu.:
##  Max.   :775091   Max.   :2017   New Jersey    : 5642   SQUIDS **              :  1027   Max.   :341
```

```
##  NA's   :98                      (Other)       :63642   (Other)                 :102067   NA's   :118
```

Edit/Restructure Data

```
temp00<-EditCommData(dat = landings.data, category0)
temp<-temp00[[1]]
```

2.4.2 B. Enter base year

```
baseyr<-2010
PercentMissingThreshold = 0.60
```

2.4.3 C. Run the function

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

2.4.4 D. Obtain the implicit quantity estimates

	REMOVED_V_Total	VV_Total	V_Total	PC_Total	QC_Total	PI_Total	Q_Total	QI_To
1950	4910008722	4875589398	4908497898	0.0000000	0.0000000	9.4019529	522072166	0.0666
1951	4468280396	4431918984	4466377584	-0.1393204	0.0448113	8.1792222	546063850	0.0696
1952	4454820075	4431689858	4453075858	0.0424945	-0.0454442	8.5342845	521786664	0.0665
1953	4541451462	4532278105	4540150905	-0.0406914	0.0602429	8.1939830	554083514	0.0707
1954	4783727020	4778675608	4782771308	-0.0462498	0.0982799	7.8236433	611322769	0.0780
1955	4864502898	4861921298	4863595098	0.0326971	-0.0161583	8.0836822	601655902	0.0767
1956	5315091181	5310423386	5314358886	-0.0201451	0.1083465	7.9224648	670796148	0.0855
1957	4812006928	4806371776	4811335276	-0.0096772	-0.0898861	7.8461675	613208332	0.0782
1958	4814125255	4806062106	4813710706	-0.0440851	0.0441001	7.5077819	641162829	0.0818
1959	5136276317	5128238707	5135938307	0.0830124	-0.0183587	8.1576201	629587827	0.0803
1960	5014090964	5006672662	5013880662	0.0721937	-0.0963180	8.7683284	571817164	0.0729
1961	5281445600	5271994100	5281061500	-0.0486803	0.1003269	8.3517065	632333227	0.0806
1962	5483719015	5475902315	5483670915	-0.0269309	0.0651460	8.1297890	674515769	0.0860
1963	4940953280	4934595280	4940885680	-0.0114944	-0.0924241	8.0368768	614776835	0.0784
1964	4658198362	4649826662	4658145962	-0.0692904	0.0101452	7.4988537	621181073	0.0792
1965	4883101538	4876195638	4883084238	-0.0692627	0.1167864	6.9970421	697878354	0.0890
1966	4406698497	4401510597	4406594597	-0.0924061	-0.0101037	6.3794472	690748661	0.0881
1967	4134834274	4129040974	4134767374	0.0991418	-0.1628276	7.0443314	586963776	0.0748
1968	4374304496	4367212996	4374214096	-0.0487513	0.1046928	6.7091479	651977597	0.0831
1969	4451508990	4445896590	4451412090	-0.1407621	0.1585088	5.8282098	763770052	0.0974
1970	4937890925	4934740380	4937749680	-0.0904910	0.1942375	5.3239681	927456658	0.1183
1971	5174335253	5172825523	5174217353	-0.0006939	0.0479531	5.3202750	972546975	0.1240
1972	4986532651	4985931355	4986310542	-0.1052940	0.0685065	4.7885660	1041295148	0.1328
1973	4999752596	4998934055	4999369896	-0.4854865	0.4880929	2.9468725	1696500216	0.2164
1974	5142256144	5139074161	5139376036	0.0256909	0.0019803	3.0235611	1699775805	0.2168
1975	5077858104	5077565864	5077574366	0.0921765	-0.1042417	3.3155112	1531460475	0.1954
1976	5585649671	5585516520	5585524140	-0.1382412	0.2335701	2.8874409	1934420235	0.2468
1977	5371655443	5371561085	5371569138	-0.1329317	0.0939162	2.5280264	2124807394	0.2711

	REMOVED_V_Total	VV_Total	V_Total	PC_Total	QC_Total	PI_Total	Q_Total	QI_To
1978	6158425762	6156919665	6157094183	-0.0769244	0.2133679	2.3408509	2630280375	0.3356
1979	6468717520	6467758155	6468442787	-0.1187386	0.1680596	2.0787691	3111669669	0.3970
1980	6558742947	6558397246	6558543970	-0.0028588	0.0166571	2.0728348	3164045735	0.4037
1981	6021841354	6020499500	6021797283	-0.0050728	-0.0809418	2.0623464	2919876732	0.3725
1982	6438791754	6437520342	6438787721	-0.0042461	0.0710469	2.0536081	3135353644	0.4000
1983	6394874948	6393035509	6394772190	-0.0170187	0.0097773	2.0189541	3167368840	0.4041
1984	6400733217	6399410691	6400720110	-0.0210457	0.0225032	1.9769077	3237743467	0.4131
1985	6327817404	6326883481	6327760091	0.0248742	-0.0361205	2.0266984	3122201132	0.3983
1986	6087762826	6086176550	6087575242	-0.0879258	0.0492370	1.8561088	3279751336	0.4184
1987	6970940970	6966436792	6970738599	-0.0810357	0.2183057	1.7116307	4072571592	0.5196
1988	7345259874	7339937458	7345092864	-0.1320955	0.1867979	1.4998290	4897287018	0.6248
1989	8703012869	8696350662	8700805819	0.0934084	0.0770196	1.6466772	5283856218	0.6742
1990	9763270615	9755987113	9759236532	-0.0336573	0.1485775	1.5921769	6129492601	0.7821
1991	9591954950	9587466341	9589284690	-0.0070791	-0.0104712	1.5809456	6065537546	0.7739
1992	9910321012	9898636780	9904847441	-0.1427526	0.1751201	1.3706301	7226492173	0.9220
1993	9931205231	9920681832	9927604931	0.1920875	-0.1897894	1.6608974	5977253663	0.7626
1994	10054778439	10048409338	10050347153	-0.0727858	0.0850619	1.5443023	6508017866	0.8304
1995	9630008417	9622004296	9626118805	-0.1181067	0.0747995	1.3722690	7014746441	0.8950
1996	9340724555	9332719998	9336940076	0.0848559	-0.1152719	1.4937974	6250472860	0.7975
1997	9583115360	9571439097	9579311407	-0.0399879	0.0655933	1.4352421	6674352397	0.8516
1998	8960217365	8948494095	8955981563	0.1754536	-0.2426518	1.7105024	5235877681	0.6680
1999	9062894698	9057085694	9059086055	0.0081807	0.0032479	1.7245529	5253005657	0.6702
2000	8833974470	8826483086	8829301770	-0.0450822	0.0193419	1.6485327	5355854939	0.6834
2001	9253069890	9238541235	9249195256	0.0240899	0.0216169	1.6887278	5477019623	0.6988
2002	9208948179	9192150175	9201271724	0.1450012	-0.1501947	1.9522385	4713190355	0.6014
2003	9291777375	9277170631	9284304076	-0.0071681	0.0163434	1.9382947	4789934150	0.6111
2004	9182300008	9105649566	9139955573	-0.2030562	0.1843708	1.5820988	5777107854	0.7371
2005	9109639858	9002339046	9043527465	-0.1134684	0.1018758	1.4123909	6402991665	0.8170
2006	9138912762	8894984841	9068378806	-0.0917597	0.0801827	1.2885587	7037614109	0.8979
2007	8885395234	8658686078	8793721529	-0.0319107	0.0033075	1.2480890	7045748823	0.8990
2008	7882138532	7525694781	7752010705	-0.2427843	0.1013045	0.9790518	7917876230	1.0103
2009	7788358546	7438874724	7617582061	0.1555141	-0.1690647	1.1437854	6659974805	0.8498
2010	8046768286	7637051682	7836994486	-0.1343433	0.1611265	1.0000000	7836994486	1.0000
2011	9426480100	9063426240	9206821949	-0.0776609	0.2506753	0.9252781	9950328996	1.2696
2012	9316801850	8965908496	9101820510	-0.0265596	0.0158965	0.9010266	10101611121	1.2889
2013	9387237694	9087008167	9221764369	-0.0265010	0.0398022	0.8774621	10509587156	1.3410
2014	9101596982	8783280740	8928117767	0.0682839	-0.1012128	0.9394717	9503338786	1.2126
2015	9284445654	9048002470	9163838776	0.0183112	0.0104446	0.9568330	9577260153	1.2220
2016	9311345804	9032205329	9189639557	-0.0468074	0.0453727	0.9130781	10064461282	1.2842
2017	9565233796	9286198101	9430258606	0.1035049	-0.0760488	1.0126505	9312451861	1.1882

Did all of the analyses work as intended?

```
print(warnings.list0)
```

```
## [[1]]
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
##
## [[2]]
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
##
## [[3]]
```

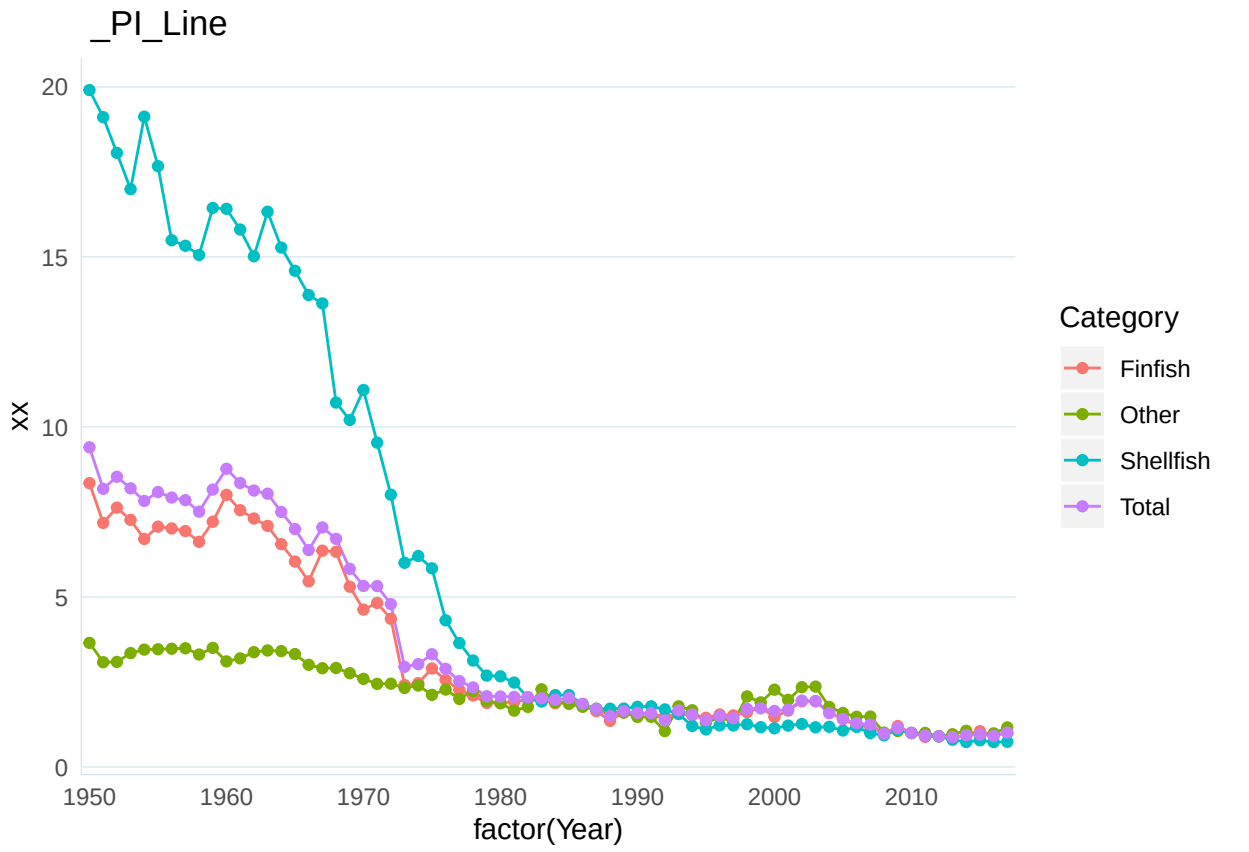
```
## [1] "When back calculated, Q_{i,t} did not equal V_{i,t}/PI_{i,t}"
##
## [[4]]
## [1] "When back calculated, V_t did not equal PI_t * Q_t"
##
## [[5]]
## [1] "When back calculated, Q_t did not equal V_t/PI_t"
##
## [[6]]
## [1] "When back calculated, ln(Q_t/Q_{t-1}) = did not equal sum( ( frac{R_{i, t} - R_{i, t-1}}{2})  *
##
## [[7]]
## [1] "Out of 30 columns, 32 of species V columns are completely empty, 34 of species Q columns are co
```

2.4.5 E. Graph

2.4.5.1 Graph 1: Price Index

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

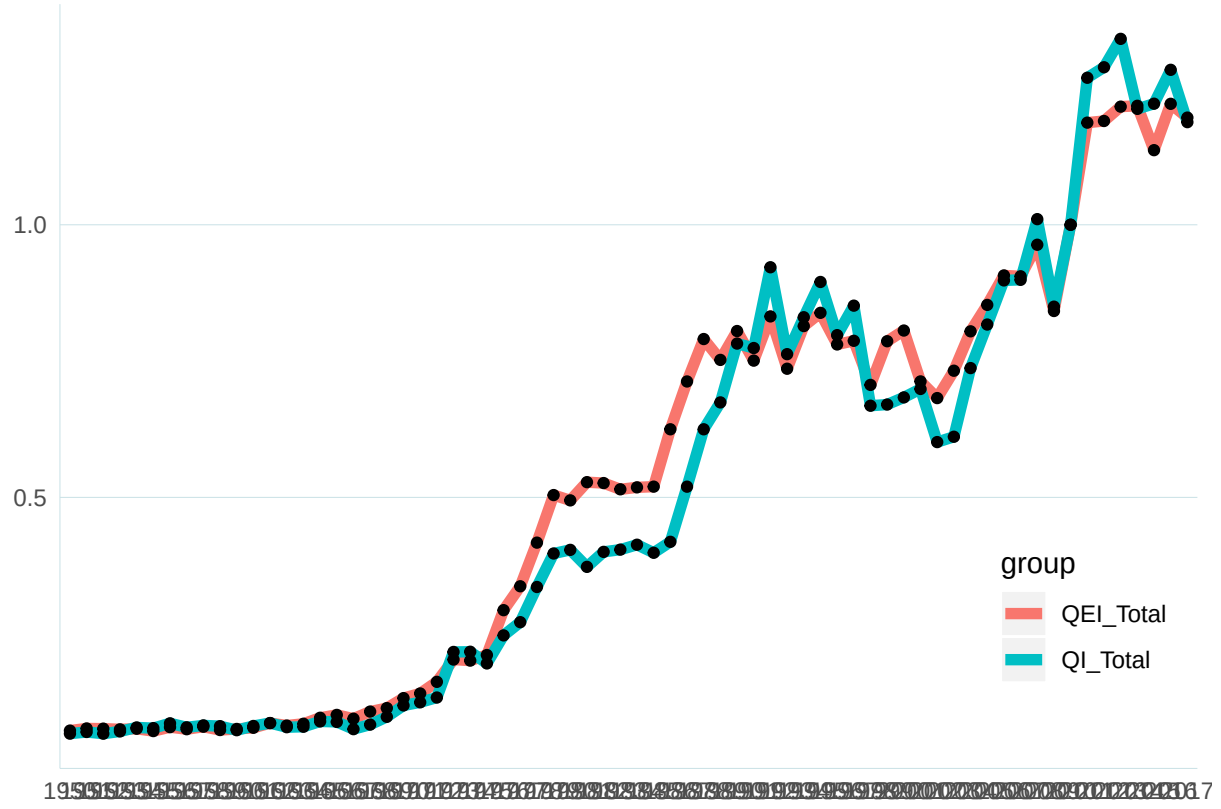
```
figures.list0$`_PI_Line`
```



2.4.5.2 Graph 2: Quantity Index Compare

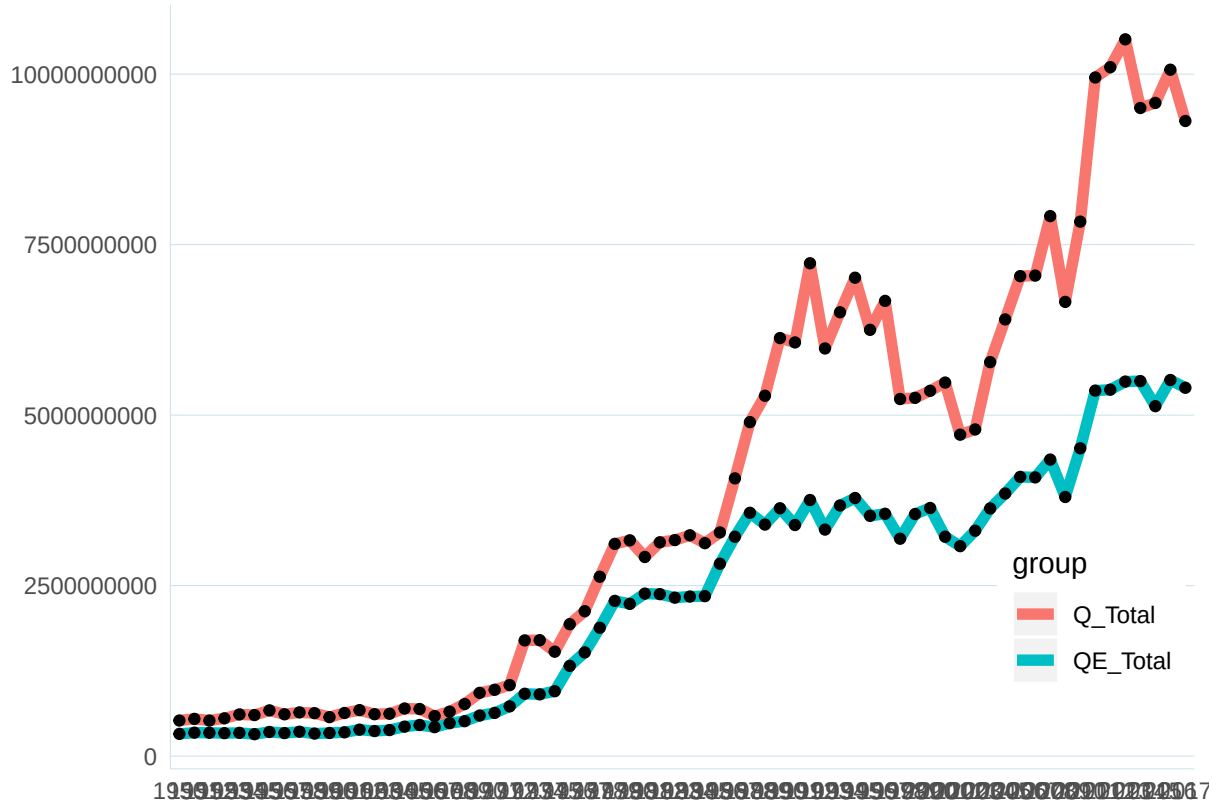
For comparison, let's recreate those graphs to make sure we are getting the same output:

```
figures.list0$`_QuantityIndexCompare`
```



2.4.5.3 Graph 3: Quantity Compare

```
figures.list0$`_QuantityCompare`
```


10000000000
7500000000
5000000000
2500000000
0
group
Q_Total
QE_Total