Quantifying the Impact of Factor Dispersion on Mexican Aggregate Productivity

Andrea Vallebueno

October 2019

Executive Summary

The inefficient allocation of factors of production in Mexico at the sub-industry level distorts the country's total factor productivity (TFP). The mechanism by which this effect takes place, as described by Hsieh-Klenow (2009) in their model of monopolistic competition, involves differences in marginal products of labor and capital across businesses within the same sub-industry. Hsieh and Klenow quantified this impact using microdata on manufacturing plants to explore the effects of this misallocation in China, India and the United States.

The following analysis employs the same methodology on Mexican businesses and plants in the manufacturing, services and commercial industries using microdata from the Mexican Statistics Insitute (INEGI). Due to the confidential nature of the data, this report develops the code used to conduct such an analysis on a simulated dataset, which is by no means illustrative of the actual data on these businesses. The simulated dataset contains the same variables as the INEGI dataset, including annual revenues, labor (hours and wages), capital, location (state), NAIC classification, for approximately 4 million businesses. The actual data will be sourced from Mexico's Economic Census (2015). It is important to note that NAIC classification is used at the four-digit level to create sub-industries in line with Hsieh-Klenow (2009).

The actual, complete analysis will be published in December 2019 as the thesis project to complete my bachelor's in Economics.

1. Read data and set variables

We begin by accessing the required libraries.

```
library(DescTools)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:DescTools':
##
##     %like%
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggplot2)
library(mxmaps)
library(utils)
library(viridis)
```

```
## Loading required package: viridisLite
```

We now read in the simulated dataset and five additional files which include the following information: * naicscodes: Contains a dictionary of the alpha level (labor shares) per NAIC industry for industries available at the 4-digit level * shares: Contains a dictionary of labor shares per NAIC indutry at the 2, 3 and 4 digit levels for industries not available in naicscodes * indicadores OECD: Contains indicators that measure the quality of housing, income, education, security and other wellbeing measures for each Mexican states as consolidated by the OECD. These can be found here.

- sigmas: Contains a dictionary on the average sigma (elasticity of usbstitution between plant value added) per NAIC subindustry within the manufacturing industry

```r
### Read simulated database
data <- read.csv("Muestra2014.csv")
data$ID <- seq.int(nrow(data)) # Add observation ID

### Read NAIC information
naicscodes <- read.csv("naic_alpha.csv")
shares <- read.csv("valueadded.csv")

## Read indexes
indicadoresOECD <- read.csv('indicesOECDporestado.csv')

## Read manufacturing sigmas
sigmas<- read.csv("sigmasbynaic.csv")
```

Below is a first glance of the dataset we will be working with. Variables are coded as per INEGI's variable names.

```r
head(data)
```

```
##   e03 e04 e05 e06 clase sector g111a a111a a121a A131A a211a a221a h000a
## 1   4  15  15 256  6130     32  1997 16681 13069  9168  -430     0    42
## 2  11  15  15 256  7129     32  1997 16681 13069  4539  -430     0    42
## 3   4  15  15 256  6105     32  1997 16681 13069  3041  -430     0    42
## 4   5  15  15 256  4303     32  1997 16681 13069  6918  -430     0    42
## 5  20  15  15 256  6113     32  1997 16681 13069  4532  -430     0    42
## 6  27  15  15 256  6216     32  1997 16681 13069  6456  -430     0    42
##   h000d h001a H001D h010a h010d h020a h020d h101a h101d h203a h203d i000a
## 1    81    42  9923    41    79     1     2    33    60     8    19     0
## 2    81    42  7794    41    79     1     2    33    60     8    19     0
## 3    81    42  6633    41    79     1     2    33    60     8    19     0
## 4    81    42  5417    41    79     1     2    33    60     8    19     0
## 5    81    42  4721    41    79     1     2    33    60     8    19     0
## 6    81    42    71    41    79     1     2    33    60     8    19     0
##   i000d i100a i100d i200a i200d J000A j010a j011a j020a j100a j200a j203a
## 1     0     0     0     0     0  5941  1240  2314   502     0  1074  1074
## 2     0     0     0     0     0  3463  1240  2314   502     0  1074  1074
## 3     0     0     0     0     0  2076  1240  2314   502     0  1074  1074
## 4     0     0     0     0     0  4849  1240  2314   502     0  1074  1074
## 5     0     0     0     0     0  9458  1240  2314   502     0  1074  1074
## 6     0     0     0     0     0  6890  1240  2314   502     0  1074  1074
##   J300A J400A j500a j600a k000a K610A K620A m000a Q000A  Q000B q000c q000d
```

```
## 1  7623  2382   107     0 13069  2277  3965 17436  5929 1185.8     0     0
## 2  2587  1732   107     0 13069  8715  2984 17436  7067 1413.4     0     0
## 3  7967  8538   107     0 13069  6552  7441 17436    84   16.8     0     0
## 4  5046  4659   107     0 13069  8934  1578 17436   671  134.2     0     0
## 5   286  1183   107     0 13069   199  1276 17436  7794 1558.8     0     0
## 6  1812   794   107     0 13069  6918  3788 17436   403   80.6     0     0
##   ID
## 1  1
## 2  2
## 3  3
## 4  4
## 5  5
## 6  6
```

```r
print('Dataset dimensions:')
```

```
## [1] "Dataset dimensions:"
```

```r
dim(data)
```

```
## [1] 5000000      50
```

```r
str(data, strict.width = 'wrap')
```

```
## 'data.frame':    5000000 obs. of  50 variables:
## $ e03 : int 4 11 4 5 20 27 17 11 16 13 ...
## $ e04 : int 15 15 15 15 15 15 15 15 15 15 ...
## $ e05 : int 15 15 15 15 15 15 15 15 15 15 ...
## $ e06 : int 256 256 256 256 256 256 256 256 256 256 ...
## $ clase : int 6130 7129 6105 4303 6113 6216 4310 7106 6205 3245 ...
## $ sector: int 32 32 32 32 32 32 32 32 32 32 ...
## $ g111a : int 1997 1997 1997 1997 1997 1997 1997 1997 1997 1997 ...
## $ a111a : int 16681 16681 16681 16681 16681 16681 16681 16681 16681 16681
##    ...
## $ a121a : int 13069 13069 13069 13069 13069 13069 13069 13069 13069 13069
##    ...
## $ A131A : int 9168 4539 3041 6918 4532 6456 1014 9648 9398 633 ...
## $ a211a : int -430 -430 -430 -430 -430 -430 -430 -430 -430 -430 ...
## $ a221a : int 0 0 0 0 0 0 0 0 0 0 ...
## $ h000a : int 42 42 42 42 42 42 42 42 42 42 ...
## $ h000d : int 81 81 81 81 81 81 81 81 81 81 ...
## $ h001a : int 42 42 42 42 42 42 42 42 42 42 ...
## $ H001D : int 9923 7794 6633 5417 4721 71 5426 5525 5488 4310 ...
## $ h010a : int 41 41 41 41 41 41 41 41 41 41 ...
## $ h010d : int 79 79 79 79 79 79 79 79 79 79 ...
## $ h020a : int 1 1 1 1 1 1 1 1 1 1 ...
## $ h020d : int 2 2 2 2 2 2 2 2 2 2 ...
## $ h101a : int 33 33 33 33 33 33 33 33 33 33 ...
## $ h101d : int 60 60 60 60 60 60 60 60 60 60 ...
## $ h203a : int 8 8 8 8 8 8 8 8 8 8 ...
## $ h203d : int 19 19 19 19 19 19 19 19 19 19 ...
## $ i000a : int 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ i000d : int 0 0 0 0 0 0 0 0 0 0 ...
## $ i100a : int 0 0 0 0 0 0 0 0 0 0 ...
## $ i100d : int 0 0 0 0 0 0 0 0 0 0 ...
## $ i200a : int 0 0 0 0 0 0 0 0 0 0 ...
## $ i200d : int 0 0 0 0 0 0 0 0 0 0 ...
## $ J000A : int 5941 3463 2076 4849 9458 6890 9724 5683 9346 8860 ...
## $ j010a : int 1240 1240 1240 1240 1240 1240 1240 1240 1240 1240 ...
## $ j011a : int 2314 2314 2314 2314 2314 2314 2314 2314 2314 2314 ...
## $ j020a : int 502 502 502 502 502 502 502 502 502 502 ...
## $ j100a : int 0 0 0 0 0 0 0 0 0 0 ...
## $ j200a : int 1074 1074 1074 1074 1074 1074 1074 1074 1074 1074 ...
## $ j203a : int 1074 1074 1074 1074 1074 1074 1074 1074 1074 1074 ...
## $ J300A : int 7623 2587 7967 5046 286 1812 4856 2894 9555 5830 ...
## $ J400A : int 2382 1732 8538 4659 1183 794 661 7925 845 1119 ...
## $ j500a : int 107 107 107 107 107 107 107 107 107 107 ...
## $ j600a : int 0 0 0 0 0 0 0 0 0 0 ...
## $ k000a : int 13069 13069 13069 13069 13069 13069 13069 13069 13069 13069
##    ...
## $ K610A : int 2277 8715 6552 8934 199 6918 6105 1260 9644 8926 ...
## $ K620A : int 3965 2984 7441 1578 1276 3788 3637 6553 1022 9148 ...
## $ m000a : int 17436 17436 17436 17436 17436 17436 17436 17436 17436 17436
##    ...
## $ Q000A : int 5929 7067 84 671 7794 403 6886 4912 63 4425 ...
## $ Q000B : num 1185.8 1413.4 16.8 134.2 1558.8 ...
## $ q000c : int 0 0 0 0 0 0 0 0 0 0 ...
## $ q000d : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...
```

Define variables

Below we instantiate relevant variables for the model, including a baseline sigma of 3, and an interest rate of 10% as per Hsieh-Klenow.

```
data$sigma<-3
r<- 0.10
sigma<-3
```

2. Clean data

We now begin to clean the data by removing missing values, checking the column types, eliminating rows for businesses that do not belong to the manufactuing, commercial or services sectors, and select only the variables that we will be using for our analysis.

```
### Remove missing values and set column types
data <- na.omit(data)

## Eliminate rows where naics doesnt fit our description
NAICsPermitidos<- c("31", "32", "33", "43", "46",
                    "51", "52", "53", "54", "55", "56",
                    "61", "62", "71", "72", "81")
data <- data[substr(as.character(data$clase), 1,2) %in% NAICsPermitidos, ]
rm(NAICsPermitidos)
```

```r
## All cols as numeric
columns <- 1:length(names(data))
for (i in columns) {
  as.numeric(data[, i])
}

## Select only variables to be used
variables <- c('ID', 'clase', 'e03', 'A131A', 'H001D', 'Q000A', 'Q000B',  'J000A',
               'K610A' , 'K620A', 'J300A', 'J400A', 'sigma')
data <- data[, variables]
rm(variables)
```

As a next step, we load the dictionary of alphas per 4-digit NAIC and add it to our datatable. We also rename our most important variables, primarily PsiYsi (annual value added) and Lsi (annual wages), and create the Ksi (capital) variable by subtracting depreciation from gross capital. In line with Hsieh-Klenow, Busso, Fazio and Levy (2012) and Mayorga (2017), we exclude industries with fewer than 10 establishments, and remove firms with zero or negative capital, value added or labor.

```r
## Add capital shares for naics codes from 2012 list
data <- merge(data, naicscodes, by.x = "clase", by.y = "X2012.CODE", all.x = TRUE)

data$substring <- as.integer(substr(data$clase, 1, 2))
shares <- shares[, c("X", "Capital.share")]
data <- merge(data, shares, by.x = 'substring', by.y = 'X', all.x = TRUE)
data$alphas <- ifelse(is.na(data$alphas), data$Capital.share, data$alphas)

### Rename used variables (Labor, wage bill, revenues, capital per plant i)
data.table::setnames(data, old = c('A131A', 'H001D'), new = c('PsiYsi', 'Lsi'))
data$Ksi <- data$Q000A - data$Q000B
data$wsiLsi <- data$J000A + data$K610A + data$K620A

### Adjust sample: Remove firms with zero or negative capital, value added or labor
data <- data[data$PsiYsi > 0,]
data <- data[data$wsiLsi > 0,]
data <- data[data$Ksi > 0,]

### Exclude industries with fewer than 10 establishments
allnaics <- data$clase
allnaics <- unique(allnaics)

data <- data.table(data)
naic_counts_table <- data[, list(naic_counts = length(clase)),
                          by = list(unique_clase = clase)]
data <- merge(data, naic_counts_table, by.x = 'clase', by.y = 'unique_clase')
data <- data[naic_counts > 10]
data <- data[, clase:wsiLsi]
```

```r
head(data)
```

```
##    clase substring      ID e03 PsiYsi  Lsi Q000A  Q000B J000A K610A K620A
## 1:  3111        31 2000723  16    681 2748  8542 1708.4  7960  3926  6114
## 2:  3111        31 1388610  11   1143 5548  2562  512.4  5383  5656  3754
```

```
## 3:   3111        31 3508083  13    2959 9759  3223  644.6  8337  5619  5467
## 4:   3111        31 4117371  21    2081 2158  2433  486.6  5360  2649  9316
## 5:   3111        31 3309643  22    3400 2999  4387  877.4  3254  5181  3955
## 6:   3111        31 2871394   1    5391 8582  3473  694.6  2061  6825  1716
##     J300A J400A sigma alphas Capital.share    Ksi wsiLsi
## 1:   6627  6297     3   0.59          0.53 6833.6  18000
## 2:   2313  6611     3   0.59          0.53 2049.6  14793
## 3:   6364  7193     3   0.59          0.53 2578.4  19423
## 4:   2323  6656     3   0.59          0.53 1946.4  17325
## 5:    435  5203     3   0.59          0.53 3509.6  12390
## 6:   9797  6349     3   0.59          0.53 2778.4  10602
```

3. Calculate Distortions and Productivity Measures

Next, we calculate the total factor productivity of revenue (TFPRsi), total factor productivity (TFPQsi), capital distortion (PlusTksi) and value-added distortion (MinusTYsi) per establishment. With this information, we proceed to calculating these same metrics at the subindustry level (TFPRs, TFPQs, PlusTks, MinusTYs).

```r
### For each plant
## Calculated as in Hsieh-Klenow
data$PlusTksi <- (data$alphas / (1- data$alphas))*(data$wsiLsi / (r * data$Ksi))
data$MinusTYsi <- (sigma/(sigma - 1))*(data$wsiLsi/((1 - data$alphas)*data$PsiYsi))
data$TFPRsi <- data$PsiYsi/((data$Ksi^data$alphas)*(data$wsiLsi^(1-data$alphas)))
data$TFPQsi <- ((data$PsiYsi)^(sigma/(sigma-1))) /
  ((data$Ksi^data$alphas)*(data$wsiLsi^(1-data$alphas)))

### Average
## Calculate TFPRs
data$wsiLsi <- as.numeric(data$wsiLsi)
data$Lsi <- as.numeric(data$Lsi)
data$PsiYsi <- as.numeric(data$PsiYsi)

sector_table <- data[, list(wsLs = sum(wsiLsi), Ls = sum(Lsi), Ks = sum(Ksi),
                            PsYs = sum(PsiYsi),
                            alphas = mean(alphas), sigma = mean(sigma)),
                            by = list(clase = clase)]
sector_table$PlusTks <- ((sector_table$alphas/(1- sector_table$alphas)) *
                           (sector_table$wsLs/(r*sector_table$Ks)))
sector_table$MinusTYs <- ((sigma/(sigma - 1))*(sector_table$wsLs /
                           ((1 - sector_table$alphas)*sector_table$PsYs)))
sector_table$TFPRs <- sector_table$PsYs/((sector_table$Ks^sector_table$alphas) *
                           (sector_table$wsLs^(1- sector_table$alphas)))
sector_table <- sector_table[, c(1:5, 8:10)]

data <- merge(data, sector_table, by.x = 'clase', by.y = 'clase', all.x = TRUE)
rm(sector_table)

## Calculate TFPQs
data$TFPQsitrans <- (data$TFPQsi*data$TFPRs/data$TFPRsi)^(sigma - 1)
sector_table <- data[, list(TFPQssum = sum(TFPQsitrans), sigma = mean(sigma)),
                     by = list(clase = clase)]
sector_table$TFPQs <-  sector_table$TFPQssum^(1/(sigma - 1))
```

```r
sector_table <- sector_table[, c(1, 2, 4)]

data <- merge(data, sector_table, by.x = 'clase', by.y = 'clase', all.x = TRUE)
rm(sector_table)

## Trim the 1% tails of log(TFPRsi/TFPRs) and log(Asi/As) across subindustries
data$logTFPRsi <- log(data$TFPRsi/data$TFPRs)
data$logTFPQsi <- log(data$TFPQsi/data$TFPQs)

logTFPRsi<- data$logTFPRsi
logTFPQsi<- data$logTFPQsi
logTFPRsiTrim<-DescTools::Trim(logTFPRsi, trim = 0.01)
logTFPQsiTrim<- DescTools::Trim(logTFPQsi, trim = 0.01)
TrimTFPRsi<- attr(logTFPRsiTrim, "trim")
TrimTFPQsi<- attr(logTFPQsiTrim, "trim")
toTrim<-unique(TrimTFPQsi, TrimTFPRsi)

data <- data[-toTrim,]
rm(logTFPRsiTrim, logTFPQsiTrim, TrimTFPRsi,TrimTFPQsi, toTrim)
```

Once we have the aggregate metrics per subindustry, we trim the 1% tails of log(TFPRsi/TFPRs) and log(Asi/As) across subindustries to remove outliers in line with the previously mentioned authors.

```r
## Trim across each industry
naics <- unique(data$clase)

trimmed_data<-NULL

for (i in naics) {
  ## Slice activity
  naics_selection <- data[clase == i,]

  ## Count number of firms with this activity
  count <- nrow(naics_selection)

  if (count < 100) {
    add_list <- list(trimmed_data, naics_selection)
    trimmed_data <- rbindlist(add_list, use.names = TRUE, fill = TRUE, idcol = FALSE)

  } else {
    ## Trim
    logTFPRsi<- naics_selection$logTFPRsi
    logTFPRsiTrim <- DescTools::Trim(logTFPRsi, trim = 0.01)

    logTFPQsi<- naics_selection$logTFPQsi
    logTFPQsiTrim <- DescTools::Trim(logTFPQsi, trim = 0.01)

    TrimTFPRsi <- attr(logTFPRsiTrim, "trim")
    TrimTFPQsi <- attr(logTFPQsiTrim, "trim")
    toTrim <- unique(TrimTFPQsi, TrimTFPRsi)

    naics_selection <- naics_selection[-toTrim,]
```

```r
    ## Bind
    add_list <- list(trimmed_data, naics_selection)
    trimmed_data <- rbindlist(add_list, use.names = TRUE, fill = TRUE, idcol = FALSE)
  }
}

rm(TrimTFPRsi, TrimTFPQsi, naics_selection, add_list, logTFPRsi, logTFPRsiTrim,
   logTFPQsi, logTFPQsiTrim)

data <- trimmed_data[, c(1:22)]
rm(trimmed_data)

## For sensitivities
# set aside a copy of this dataset to be used later in the analysis
data_sensitivities <- data[, -14]
```

4. Re-calculate Aggregate Measures and Calculate Subindustry Shares

Now that we have individual productivity and distortion metrics, and have removed outliers, we need to recalculate subindutry aggregate productivity and distortion metrics to reflect this reduced set of establishments that we will perform our analysis on.

```r
## Calculate TFPRs
sector_table <- data[, list(wsLs = sum(wsiLsi), Ls = sum(Lsi), Ks = sum(Ksi),
                            PsYs = sum(PsiYsi), alphas = mean(alphas),
                            sigma = mean(sigma)), by = list(clase = clase)]
sector_table$PlusTks <- ((sector_table$alphas/(1- sector_table$alphas)) *
                           (sector_table$wsLs/(r*sector_table$Ks)))
sector_table$MinusTYs <- ((sigma/(sigma - 1))*
                            (sector_table$wsLs /
                               ((1 - sector_table$alphas)*sector_table$PsYs)))
sector_table$TFPRs <- sector_table$PsYs/((sector_table$Ks^sector_table$alphas) *
                           (sector_table$wsLs^(1- sector_table$alphas)))
sector_table <- sector_table[, c(1:5, 8:10)]
data <- merge(data, sector_table, by.x = 'clase', by.y = 'clase', all.x = TRUE)

## Calculate TFPQs and As
data$TFPQsitrans <- (data$TFPQsi * data$TFPRs / data$TFPRsi)^(sigma - 1)
sector_table_2 <- data[, list(TFPQssum = sum(TFPQsitrans),
                              Assum = sum(TFPQsi^(sigma - 1)), sigma = mean(sigma)),
                              by = list(clase = clase)]
sector_table_2$TFPQs <- sector_table_2$TFPQssum^(1/(sigma - 1))
sector_table_2$As <- (sector_table_2$Assum)^(1/(sigma - 1))
sector_table_2 <- sector_table_2[, c(1, 2, 3, 5, 6)]

data <- merge(data, sector_table_2, by.x = 'clase', by.y = 'clase', all.x = TRUE)
rm(sector_table_2)
```

At this moment, we are ready to calculate subindustry shares (denoted by theta as per Hsieh-Klenow).

```r
PsYs_values <- sector_table$PsYs
PY <- sum(PsYs_values)
```

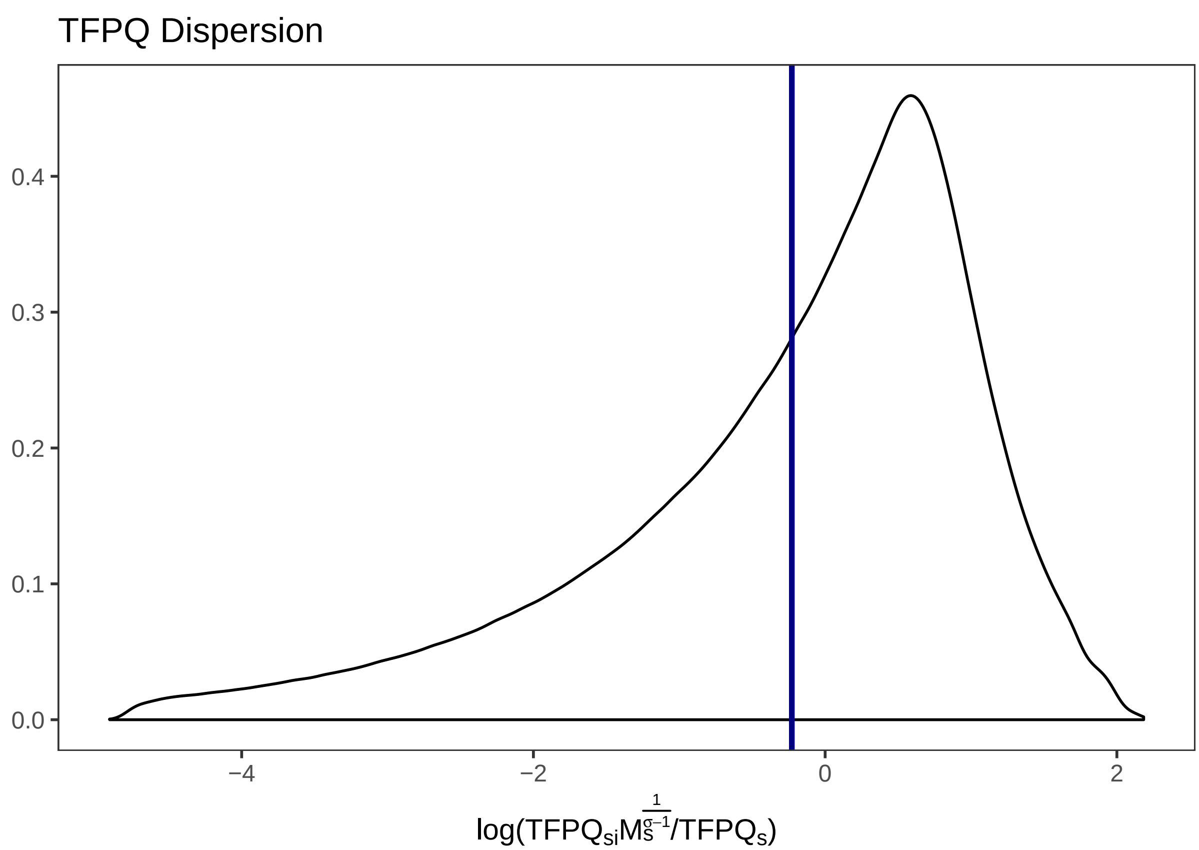
```r
sector_table$thetas <- sector_table$PsYs / PY

data <- merge(data, sector_table[, c("clase", "thetas")], by.x = 'clase',
              by.y = 'clase', all.x = TRUE)
rm(sector_table)
```

5. Measure dispersion and distribution of TFPQ and TFPR, and scaled capital and output distortions

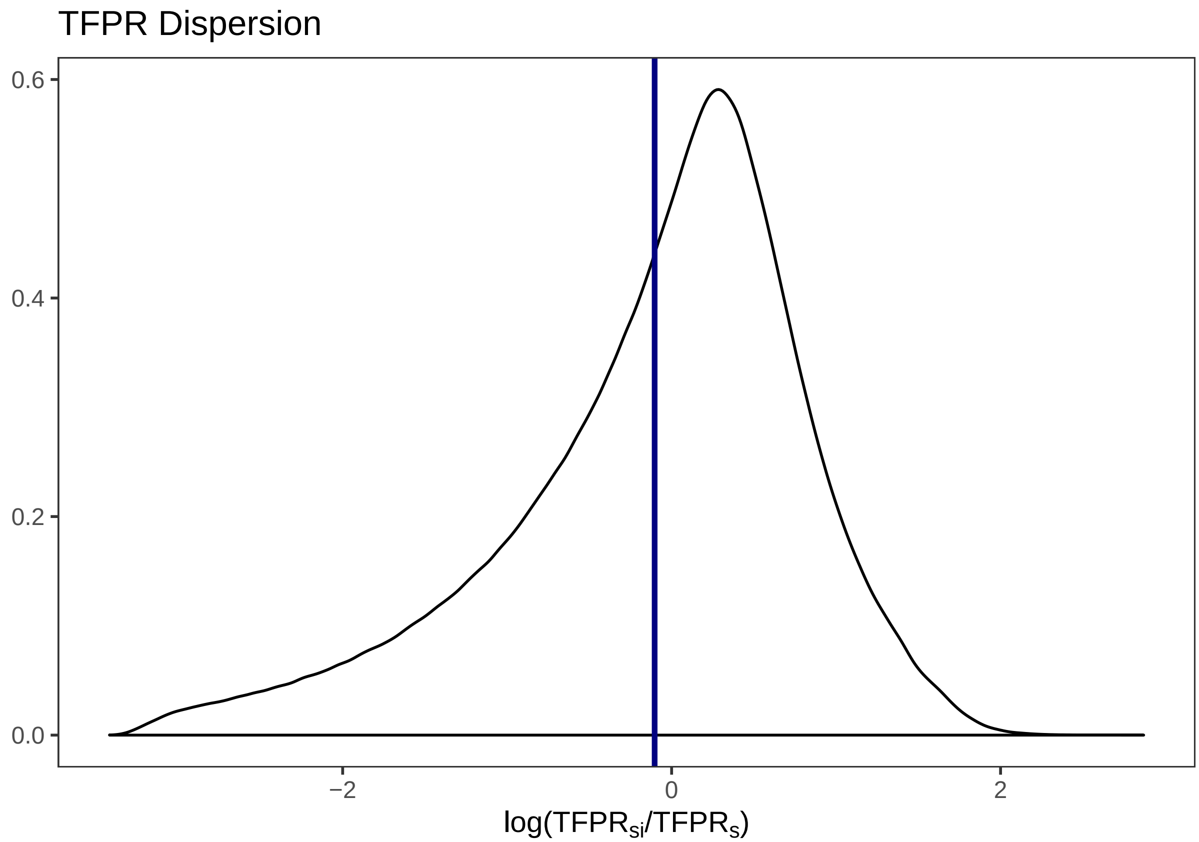
The following code adds an Ms_totals variable, which counts the number of establishments per subindustry. With this last variable we can now calculate and graph the dispersion of TFPQ, TFPR and the scaled capital and value-added distortions.

```r
## Calculate Ms
Ms_totals <- data
Ms_totals$count_naic <- 1
Ms_totals <- Ms_totals[, list(Ms = sum(count_naic)), by = list(clase = clase)]

data <- merge(data, Ms_totals, by.x ='clase', by.y = 'clase', all.x = TRUE)

data$TFPQdispersion <- log(data$TFPQsi*(data$Ms^(1/(sigma - 1)))/data$TFPQs)
data$TFPRdispersion <- log(data$TFPRsi/data$TFPRs)
data$scaledKdistortion <- log(data$PlusTksi/data$PlusTks)
data$scaledYdistortion <- log(data$MinusTYsi/data$MinusTYs)

data <- data[, c('clase', 'e03', 'PsiYsi', 'Lsi', 'ID', 'alphas', 'Ksi', 'wsiLsi',
                 'PlusTksi', 'MinusTYsi', 'TFPQsi', 'TFPRsi', 'wsLs', 'Ls', 'Ks',
                 'PsYs', 'PlusTks', 'MinusTYs',  'TFPRs', 'TFPQs', 'thetas','Ms',
                 'TFPQdispersion', 'TFPRdispersion', 'scaledKdistortion',
                 'scaledYdistortion', 'As', 'sigma', 'J300A', 'J400A', 'J000A',
                 'K610A', 'K620A')]

#### Graph dispersion
graphdispersionTFPQ <- ggplot(data, aes(x = TFPQdispersion)) + geom_density() +
  geom_vline(aes(xintercept = mean(TFPQdispersion)), color="navy", size=1) +
  ggtitle("TFPQ Dispersion") +
  labs(x = expression(paste("log(",TFPQ[si]*M[s]^{frac(1, sigma - 1)},"/",TFPQ[s],")")) ,
       y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

graphdispersionTFPR <- ggplot(data, aes(x = TFPRdispersion)) + geom_density() +
  geom_vline(aes(xintercept = mean(TFPRdispersion)), color="navy", size=1) +
  ggtitle("TFPR Dispersion") +
  labs(x = expression(paste("log(",TFPR[si],"/",TFPR[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

graphdispersionSKD <- ggplot(data, aes(x = scaledKdistortion)) + geom_density() +
  geom_vline(aes(xintercept = mean(scaledKdistortion)), color="navy", size=1) +
  ggtitle("Capital Distortion Dispersion") +
  labs(x =  expression(paste("log((",1-tau[ksi],")/(",1-tau[ks],"))")), y = "") +
```

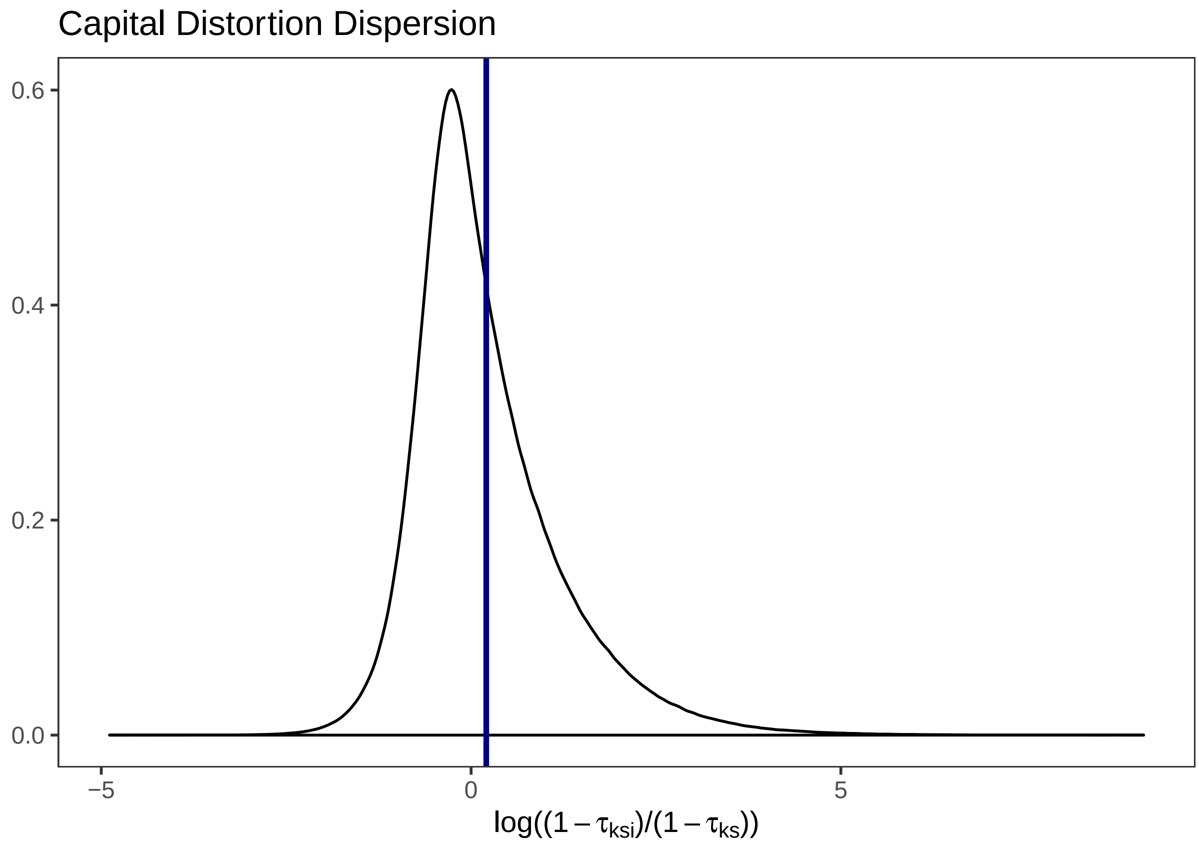
```
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

graphdispersionSYD <- ggplot(data, aes(x = scaledYdistortion)) + geom_density() +
  geom_vline(aes(xintercept = mean(scaledYdistortion)), color="navy", size=1) +
  ggtitle("Value-added Distortion Dispersion") +
  labs(x = expression(paste("log((",1-tau[Ysi],")/(",1-tau[Ys],"))")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

graphdispersionTFPQ
```



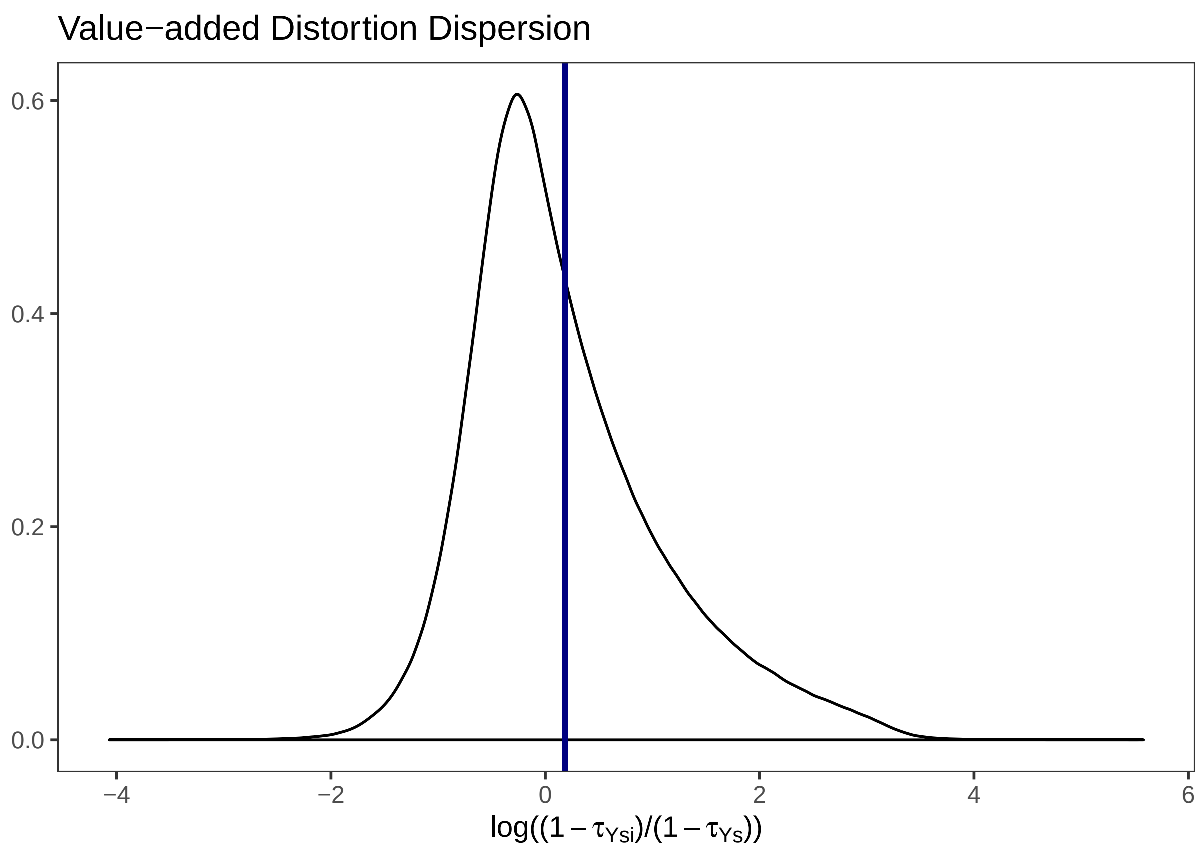

```
graphdispersionTFPR
```




```
graphdispersionSKD
```



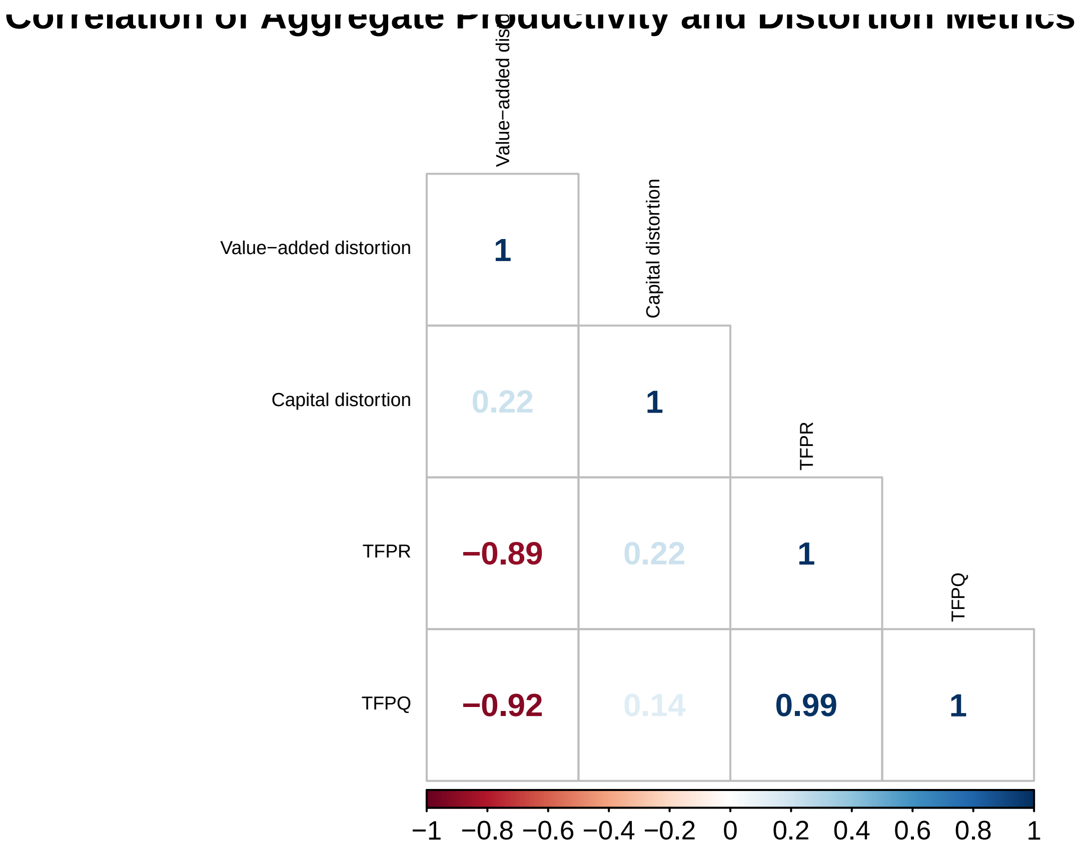

```
graphdispersionSYD
```




```
rm(graphdispersionTFPQ, graphdispersionTFPR, graphdispersionSKD, graphdispersionSYD)
```

6. Correlation between measures of productivity and distortions

```
productivity_distorsions <- data[, c('TFPQdispersion', 'TFPRdispersion',
                                     'scaledKdistortion', 'scaledYdistortion')]

corr_matrix0<-cor(productivity_distorsions, use = "complete.obs")

colnames(corr_matrix0) <- c("TFPQ", "TFPR", "Capital distortion",
                            "Value-added distortion")
rownames(corr_matrix0) <- c("TFPQ", "TFPR", "Capital distortion",
                            "Value-added distortion")

corrplot::corrplot(corr_matrix0, order = "FPC", method = "number",
                   type = "lower", tl.cex = 0.6,
                   tl.col = rgb(0,0,0),
                   title = "Correlation of Aggregate Productivity and Distortion Metrics")
```




```r
rm(productivity_distorsions)
```

7. Calculate efficient output

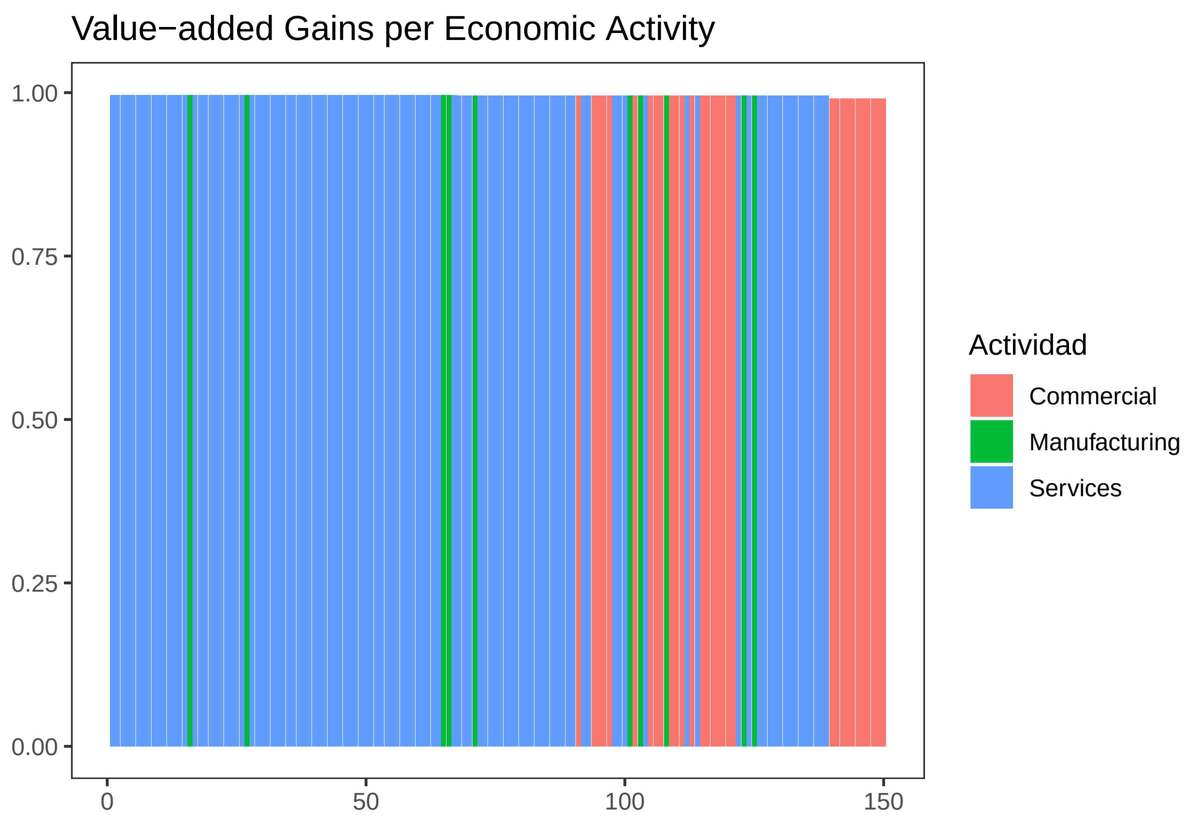
We now proceed to quantify the impact on the aggregate economy and per state of these distortions. ### Mexico

```r
### 1. Efficient output for aggregate economy
gains <- data[, list(TFPQs = mean(TFPQs), As = mean(As), thetas = mean(thetas),
                     PsYs = sum(PsiYsi)), by = list(clase = clase)]
gains$k <- (gains$TFPQs / gains$As)^gains$thetas

ks <- prod(gains$k)
productivitygain <- 1/ks - 1

### Economic activity analysis of top contributors to growth
## Add activity labels
activity_index <- data.frame(EconomicActivityIndex = c("31", "32", "33", "43", "46",
                                                     "51", "52", "53", "54", "55", "56",
                                                     "61", "62", "71", "72", "81"),
                          Actividad = c("Manufacturing","Manufacturing","Manufacturing",
                                        "Commercial", "Comercial", "Services", "Services",
                                        "Services", "Services", "Services", "Services",
                                        "Services", "Services", "Services", "Services",
                                        "Services" ))
```

```r
gains$clase <- as.character(gains$clase)
gains$EconomicActivityIndex <- substr(gains$clase, 1, 2)

gains <- merge(gains, activity_index, by.x = 'EconomicActivityIndex',
               by.y = 'EconomicActivityIndex', all.x = TRUE)
gains <- data.table::setorder(gains, -k)
gains$order <- seq.int(nrow(gains))

head(gains)
```

```
##    EconomicActivityIndex clase    TFPQs       As      thetas      PsYs
## 1:                    62  6223 5191.206 8741.150 0.006283022 152074824
## 2:                    62  6226 5253.172 8941.288 0.006204798 150181483
## 3:                    62  6221 5204.867 8769.106 0.006328093 153165724
## 4:                    62  6222 5155.421 8748.129 0.006245942 151177349
## 5:                    62  6227 5309.889 8986.798 0.006285974 152146270
## 6:                    62  6206 5312.477 9014.808 0.006261590 151556087
##            k Actividad order
## 1: 0.9967314  Services     1
## 2: 0.9967054  Services     2
## 3: 0.9967045  Services     3
## 4: 0.9967026  Services     4
## 5: 0.9966979  Services     5
## 6: 0.9966943  Services     6
```

```r
gainsanalysisgraph <- ggplot(gains, aes(x = order, y = k, fill = Actividad)) +
  geom_bar(stat="identity") +
  ggtitle("Value-added Gains per Economic Activity") +
  labs(x = "", y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())
gainsanalysisgraph
```



Mexican states

```r
### 2. Efficient output per state, dispersion in TFPR and in scaled capital distortions
## Calculate wsLse, Kse, PsYse, TFPRse and TFPQse, where e indicates a variable is an
## aggregate at the state level
state_gains <- data
state_gains$e03 <- as.character(state_gains$e03)

## Summarize figures by state
## TFPRSe
state_gains_0 <- state_gains
state_gains_0$count <- 1
state_gains_0 <- state_gains_0[, list(wsLse = sum(wsiLsi), Kse = sum(Ksi),
                                      PsYse = sum(PsiYsi), alphas = mean(alphas),
                                      Mse = sum(count), sigma= mean(sigma)),
                                by = list(e03 = e03, clase = clase)]

state_gains_0$TFPRse <- state_gains_0$PsYse / ((state_gains_0$Kse^state_gains_0$alphas)*
                                          (state_gains_0$wsLse^(1- state_gains_0$alphas)))
state_gains_0$PlusTkse <- ((state_gains_0$alphas / (1- state_gains_0$alphas)) *
                              (state_gains_0$wsLse/(r*state_gains_0$Kse)))
state_gains_0$MinusTYse <- ((sigma/(sigma - 1)) *
                               (state_gains_0$wsLse/((1 - state_gains_0$alphas) *
                                                        state_gains_0$PsYse)))
```

```r
state_gains_0$indicatorvariable <- paste(as.character(state_gains_0$e03),
                                         as.character(state_gains_0$clase), sep ="")
state_gains_0 <- state_gains_0[, c('e03', 'wsLse', 'Kse', 'PsYse', 'TFPRse',
                                   'indicatorvariable', 'PlusTkse', 'MinusTYse', 'Mse')]

state_gains$indicatorvariable <- paste(as.character(state_gains$e03),
                                       as.character(state_gains$clase), sep ="")
state_gains <- state_gains[, -'e03']
state_gains <- merge(state_gains, state_gains_0, by.x = 'indicatorvariable',
                     by.y = 'indicatorvariable', all.x = TRUE)
rm(state_gains_0)

## TFPQse and Ase
state_gains$TFPQsitrans <- (state_gains$TFPQsi * state_gains$TFPRse /
                              state_gains$TFPRsi)^(sigma - 1)
state_gains_1 <- state_gains
state_gains_1 <- state_gains_1[, list(TFPQssum = sum(TFPQsitrans),
                                      Assum = sum(TFPQsi^(sigma - 1)), sigma = mean(sigma)),
                               by = list(e03 = e03, clase = clase)]
state_gains_1$TFPQse <- state_gains_1$TFPQssum^(1/(sigma - 1))
state_gains_1$Ase <- (state_gains_1$Assum)^(1/(sigma - 1))
state_gains_1$indicatorvariable <- paste(as.character(state_gains_1$e03),
                                         as.character(state_gains_1$clase), sep ="")
state_gains_1 <- state_gains_1[, c('TFPQse', 'Ase', 'indicatorvariable', 'e03')]

state_gains <- merge(state_gains, state_gains_1[,-'e03'], by.x = 'indicatorvariable',
                     by.y = 'indicatorvariable', all.x = TRUE)
rm(state_gains_1)

state_gains$TFPQdispersione <- log(state_gains$TFPQsi *
                                     (state_gains$Mse^(1/(sigma - 1)))/state_gains$TFPQse)
state_gains$TFPRdispersione <- log(state_gains$TFPRsi/state_gains$TFPRse)
state_gains$scaledKdistortione <- log(state_gains$PlusTksi/state_gains$PlusTkse)
state_gains$scaledYdistortione <- log(state_gains$MinusTYsi/state_gains$MinusTYse)

## Add thetase
state_gains_2 <- state_gains[, list(PYe = sum(PsiYsi)), by = list(e03 = e03)]

state_gains <- state_gains[, list(TFPQse = mean(TFPQse), Ase = mean(Ase),
                                  averageTFPRdispersion = mean(TFPRdispersione),
                                  averageSKdistortion = mean(scaledKdistortione),
                                  sdTFPRdispersion = sd(TFPRdispersione),
                                  sdSKD = sd(scaledKdistortione), PsYse = sum(PsiYsi)),
                           by = list(e03 = e03, clase = clase)]
state_gains$indicatorvariable <- paste(as.character(state_gains$e03),
                                       as.character(state_gains$clase), sep ="")

state_gains <- merge(state_gains, state_gains_2, by = 'e03', all.x = TRUE)
state_gains$thetase <- state_gains$PsYse / state_gains$PYe
rm(state_gains_2)

## Produce final table
state_gains$k <- (state_gains$TFPQse / state_gains$Ase)^state_gains$thetase
```

```r
state_gains <- state_gains[, list(stategain = 1/prod(k)-1,
                                  averageTFPRdispersion = mean(averageTFPRdispersion),
                                  averageSKdistortion = mean(averageSKdistortion),
                                  PYe = sum(PsYse)),
                           by = list(e03 = e03)]
state_gains$averageTFPRdispersionsquared <- state_gains$averageTFPRdispersion^2
state_gains$averageSKdistortionsquared <- state_gains$averageSKdistortion^2

state_gains
```

```
##      e03 stategain averageTFPRdispersion averageSKdistortion       PYe
##  1:    1 0.8706309            -0.1038191           0.2062767 755271718
##  2:   10 0.8727474            -0.1042993           0.2068761 757906181
##  3:   11 0.8739574            -0.1061047           0.2060285 755795358
##  4:   12 0.8750313            -0.1051133           0.2052851 756776604
##  5:   13 0.8738720            -0.1071313           0.2046120 752388416
##  6:   14 0.8708344            -0.1042245           0.2059382 757756576
##  7:   15 0.8788899            -0.1066960           0.2045761 759478402
##  8:   16 0.8733899            -0.1055221           0.2054701 754907932
##  9:   17 0.8744046            -0.1064284           0.2048066 759691978
## 10:   18 0.8668681            -0.1033083           0.2065166 758780967
## 11:   19 0.8730424            -0.1054751           0.2086417 754604484
## 12:    2 0.8703825            -0.1069029           0.2045374 753614165
## 13:   20 0.8749738            -0.1053751           0.2064663 756620543
## 14:   21 0.8743562            -0.1057634           0.2067777 752476352
## 15:   22 0.8710565            -0.1050333           0.2069444 755566380
## 16:   23 0.8756933            -0.1063874           0.2069014 757049965
## 17:   24 0.8736333            -0.1059227           0.2043487 757269897
## 18:   25 0.8715711            -0.1053268           0.2076097 758354278
## 19:   26 0.8706836            -0.1045045           0.2061905 757871636
## 20:   27 0.8695122            -0.1077106           0.2035298 756867524
## 21:   28 0.8703728            -0.1053265           0.2065680 755684090
## 22:   29 0.8710551            -0.1058536           0.2041577 756694402
## 23:    3 0.8710048            -0.1060718           0.2049699 754637393
## 24:   30 0.8706556            -0.1043760           0.2051478 752547262
## 25:   31 0.8692816            -0.1062950           0.2043941 755265520
## 26:   32 0.8749791            -0.1050433           0.2046885 756179360
## 27:    4 0.8679365            -0.1064868           0.2044009 756821259
## 28:    5 0.8675666            -0.1045455           0.2049787 760258114
## 29:    6 0.8747551            -0.1073126           0.2055731 756646039
## 30:    7 0.8707280            -0.1028931           0.2060757 756328200
## 31:    8 0.8683568            -0.1041199           0.2044817 757404675
## 32:    9 0.8778985            -0.1039199           0.2071068 756574527
##      e03 stategain averageTFPRdispersion averageSKdistortion       PYe
##      averageTFPRdispersionsquared averageSKdistortionsquared
##  1:                    0.01077841                 0.04255009
##  2:                    0.01087834                 0.04279773
##  3:                    0.01125821                 0.04244776
##  4:                    0.01104881                 0.04214196
##  5:                    0.01147711                 0.04186607
##  6:                    0.01086274                 0.04241053
##  7:                    0.01138404                 0.04185136
##  8:                    0.01113492                 0.04221798
```

```
##  9:                  0.01132701                0.04194574
## 10:                  0.01067261                0.04264909
## 11:                  0.01112499                0.04353135
## 12:                  0.01142823                0.04183556
## 13:                  0.01110391                0.04262834
## 14:                  0.01118590                0.04275702
## 15:                  0.01103200                0.04282598
## 16:                  0.01131828                0.04280817
## 17:                  0.01121961                0.04175841
## 18:                  0.01109373                0.04310178
## 19:                  0.01092119                0.04251452
## 20:                  0.01160157                0.04142437
## 21:                  0.01109367                0.04267034
## 22:                  0.01120499                0.04168039
## 23:                  0.01125123                0.04201265
## 24:                  0.01089435                0.04208563
## 25:                  0.01129863                0.04177695
## 26:                  0.01103409                0.04189738
## 27:                  0.01133943                0.04177973
## 28:                  0.01092977                0.04201628
## 29:                  0.01151599                0.04226029
## 30:                  0.01058700                0.04246718
## 31:                  0.01084096                0.04181278
## 32:                  0.01079934                0.04289323
##     averageTFPRdispersionsquared averageSKdistortionsquared
```

```
## Plot gains per state
state_gains_plot <- data.frame(region = state_gains$e03, value = state_gains$stategain)
state_numbers <- 1:nrow(state_gains_plot)
state_gains_plot$region <- as.character(state_gains_plot$region)
toEdit <- c("1", "2", "3", "4", "5", "6", "7", "8", "9")

for (i in state_numbers) {
  if (state_gains_plot$region[i] %in% toEdit) {
    state_gains_plot$region[i] <- paste("0", state_gains_plot$region[i], sep = "")
  }
}

rm(state_gains_plot)

state_mean_scaled_TFPR <- data.frame(region = state_gains$e03,
                                     value = state_gains$averageTFPRdispersion)
state_mean_scaled_TFPR
```

```
##    region      value
## 1       1 -0.1038191
## 2      10 -0.1042993
## 3      11 -0.1061047
## 4      12 -0.1051133
## 5      13 -0.1071313
## 6      14 -0.1042245
## 7      15 -0.1066960
## 8      16 -0.1055221
## 9      17 -0.1064284
```

```
## 10      18 -0.1033083
## 11      19 -0.1054751
## 12       2 -0.1069029
## 13      20 -0.1053751
## 14      21 -0.1057634
## 15      22 -0.1050333
## 16      23 -0.1063874
## 17      24 -0.1059227
## 18      25 -0.1053268
## 19      26 -0.1045045
## 20      27 -0.1077106
## 21      28 -0.1053265
## 22      29 -0.1058536
## 23       3 -0.1060718
## 24      30 -0.1043760
## 25      31 -0.1062950
## 26      32 -0.1050433
## 27       4 -0.1064868
## 28       5 -0.1045455
## 29       6 -0.1073126
## 30       7 -0.1028931
## 31       8 -0.1041199
## 32       9 -0.1039199
```

```r
rm(state_mean_scaled_TFPR)
```

In the following figure, we map these gains per state to examine the pattern of potential gains geographically.

```r
state_gains_plot <- data.frame(region = state_gains$e03, value = state_gains$stategain)
state_gains_plot$region <- as.character(state_gains_plot$region)
toEdit <- c("1", "2", "3", "4", "5", "6", "7", "8", "9")

for (i in state_numbers) {
  if (state_gains_plot$region[i] %in% toEdit) {
    state_gains_plot$region[i] <- paste("0", state_gains_plot$region[i], sep = "")
  }
}

state_gains_plot$value <- state_gains_plot$value*100

ggstategains = MXStateChoropleth$new(state_gains_plot)
ggstategains$title <- "Ganancias en producto por estado"
ggstategains$set_num_colors(1)
ggstategains$ggplot_scale <- scale_fill_viridis('Porcentaje', option = "viridis")
ggstategains$render()
```

8. Correlation with other variables by state

We now aggregate by state the productivity and distortions we obtained and correlate them with wellbeing indicators consolidated by the OECD on housing, education, security and other aspects for each state.

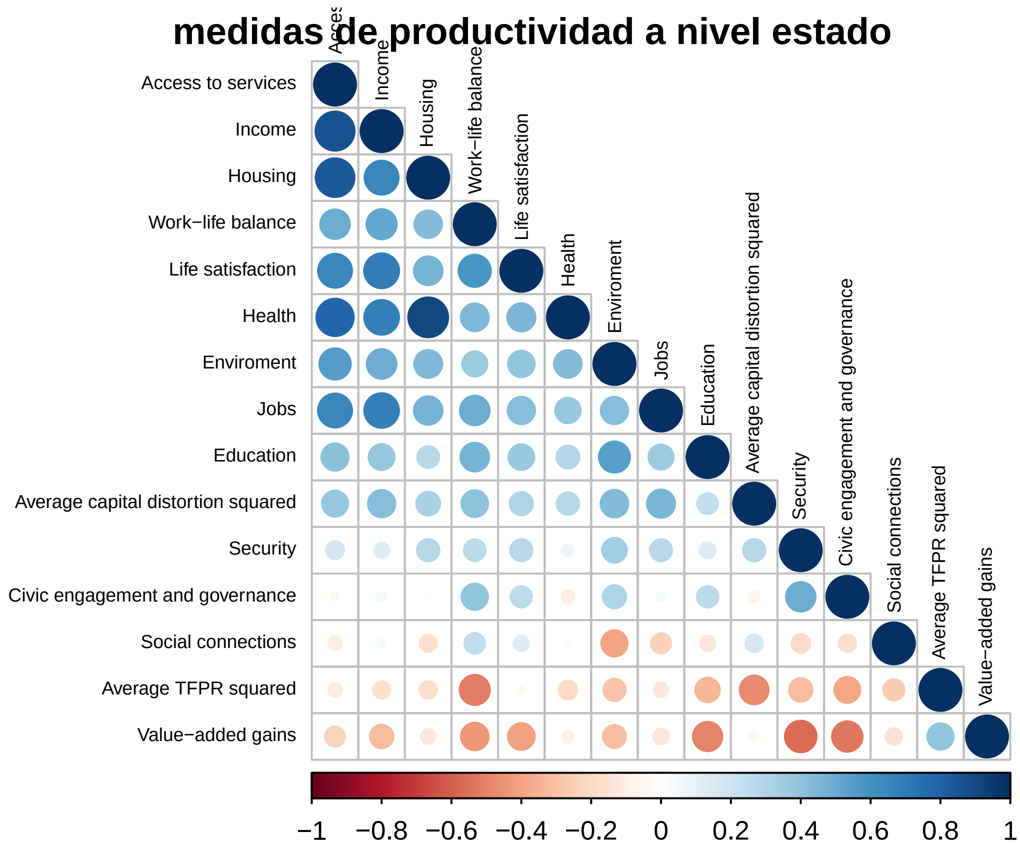
```
state_levels <- data.frame(region = as.character(state_gains$e03),
                           product.gain = state_gains$stategain,
                           averageTFPRsq = state_gains$averageTFPRdispersionsquared,
                           averageSKdistortionsq = state_gains$averageSKdistortionsquared)
print(state_levels)
```

```
##    region product.gain averageTFPRsq averageSKdistortionsq
## 1       1    0.8706309    0.01077841            0.04255009
## 2      10    0.8727474    0.01087834            0.04279773
## 3      11    0.8739574    0.01125821            0.04244776
## 4      12    0.8750313    0.01104881            0.04214196
## 5      13    0.8738720    0.01147711            0.04186607
## 6      14    0.8708344    0.01086274            0.04241053
## 7      15    0.8788899    0.01138404            0.04185136
## 8      16    0.8733899    0.01113492            0.04221798
## 9      17    0.8744046    0.01132701            0.04194574
## 10     18    0.8668681    0.01067261            0.04264909
## 11     19    0.8730424    0.01112499            0.04353135
## 12      2    0.8703825    0.01142823            0.04183556
```

```
## 13     20    0.8749738    0.01110391            0.04262834
## 14     21    0.8743562    0.01118590            0.04275702
## 15     22    0.8710565    0.01103200            0.04282598
## 16     23    0.8756933    0.01131828            0.04280817
## 17     24    0.8736333    0.01121961            0.04175841
## 18     25    0.8715711    0.01109373            0.04310178
## 19     26    0.8706836    0.01092119            0.04251452
## 20     27    0.8695122    0.01160157            0.04142437
## 21     28    0.8703728    0.01109367            0.04267034
## 22     29    0.8710551    0.01120499            0.04168039
## 23      3    0.8710048    0.01125123            0.04201265
## 24     30    0.8706556    0.01089435            0.04208563
## 25     31    0.8692816    0.01129863            0.04177695
## 26     32    0.8749791    0.01103409            0.04189738
## 27      4    0.8679365    0.01133943            0.04177973
## 28      5    0.8675666    0.01092977            0.04201628
## 29      6    0.8747551    0.01151599            0.04226029
## 30      7    0.8707280    0.01058700            0.04246718
## 31      8    0.8683568    0.01084096            0.04181278
## 32      9    0.8778985    0.01079934            0.04289323
```

```
state_levels$region <- as.character(state_levels$region)

filasindicadoresOECD <- 1:nrow(indicadoresOECD)
indicadoresOECD$region <- as.character(indicadoresOECD$region)
toEdit<-c("1", "2", "3", "4", "5", "6", "7", "8", "9")

for (i in filasindicadoresOECD) {
  if (indicadoresOECD$region[i] %in% toEdit) {
    indicadoresOECD$region[i]<-paste("0",indicadoresOECD$region[i], sep="")
  }
}

OECDindextablewithgains <- merge(indicadoresOECD, state_levels, by = "region",
                                 all.x = TRUE)

corr_matrix2 <- cor(OECDindextablewithgains[, c(-1,-2)], use = "complete.obs")

colnames(corr_matrix2)<- c("Housing", "Income", "Jobs", "Access to services",
                           "Security", "Education",  "Enviroment",
                           "Civic engagement and governance", "Health",
                           "Life satisfaction", "Work-life balance", "Social connections",
                           "Value-added gains", "Average TFPR squared",
                           "Average capital distortion squared")
rownames(corr_matrix2)<- c("Housing", "Income", "Jobs", "Access to services",
                           "Security", "Education",  "Enviroment",
                           "Civic engagement and governance", "Health",
                           "Life satisfaction", "Work-life balance", "Social connections",
                           "Value-added gains", "Average TFPR squared",
                           "Average capital distortion squared")

corrplot::corrplot(corr_matrix2, order = "FPC", method = "circle", type = "lower",
                   tl.cex = 0.6,
         tl.col = rgb(0,0,0),
```

```r
        title = "Correlacion entre indices de bienestar de OCDE y
        medidas de productividad a nivel estado")
```




```r
rm(corr_matrix1, corr_matrix2, indicadores, indicadoresOECD)
```

```
## Warning in rm(corr_matrix1, corr_matrix2, indicadores, indicadoresOECD):
## object 'corr_matrix1' not found
```

```
## Warning in rm(corr_matrix1, corr_matrix2, indicadores, indicadoresOECD):
## object 'indicadores' not found
```

9. Firm profiles and productivity

We now dive deeper to explore productivity measures according to different firm profiles: firm type and firm economic activity.

9.1 Firm type

In this section, we classify firms as legal/illegal and formal/informal as per a Legality and Formality Index as established by Busso, Levy and Levy (2012). The classification is as follows:

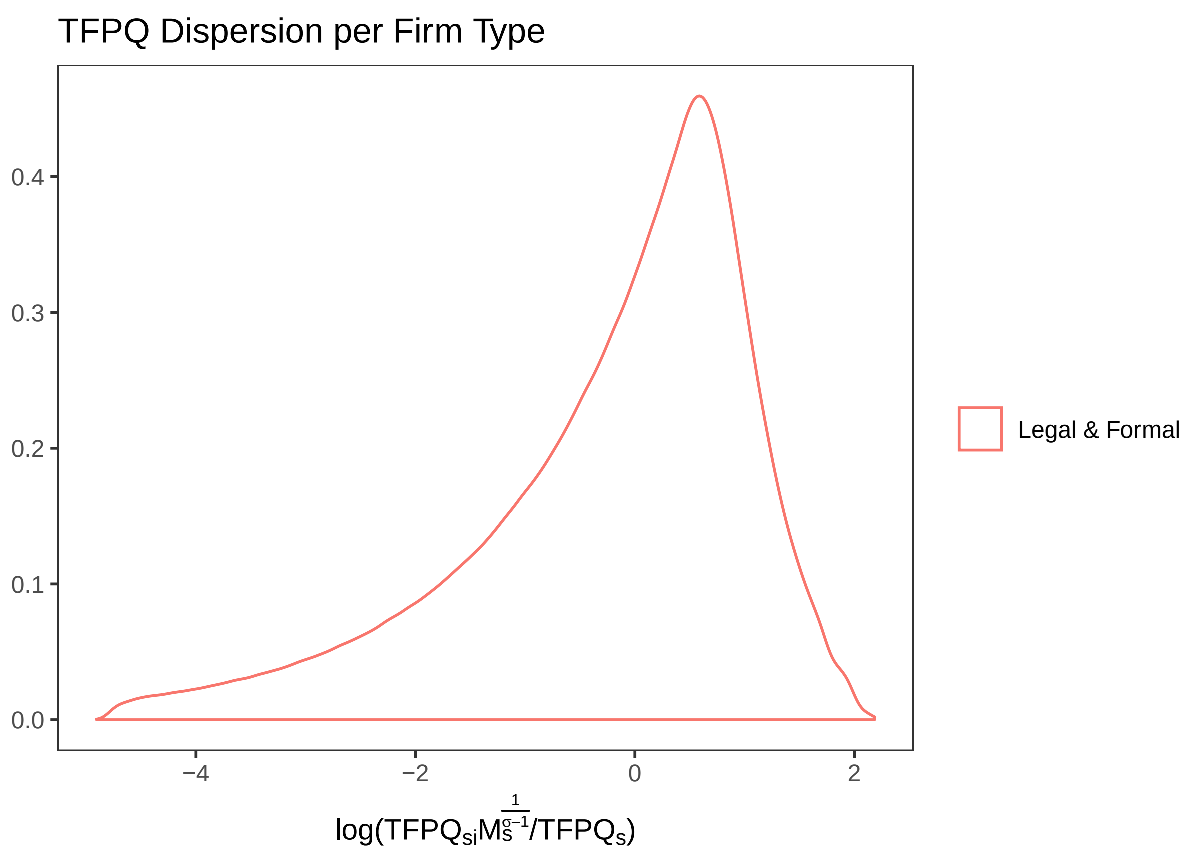
Table 1: Classification of firms as per legality and informality indexes

ï..	Index.of.Legality	Index.of.Formality
Legal and Formal	>18%	>18%
Legal and Informal	not defined	0%
Legal and Semi-formal	>18%	0% -18%
Semi-legal and Semi-formal	0% -18%	0% -18%
Illegal and Informal	0%	0%

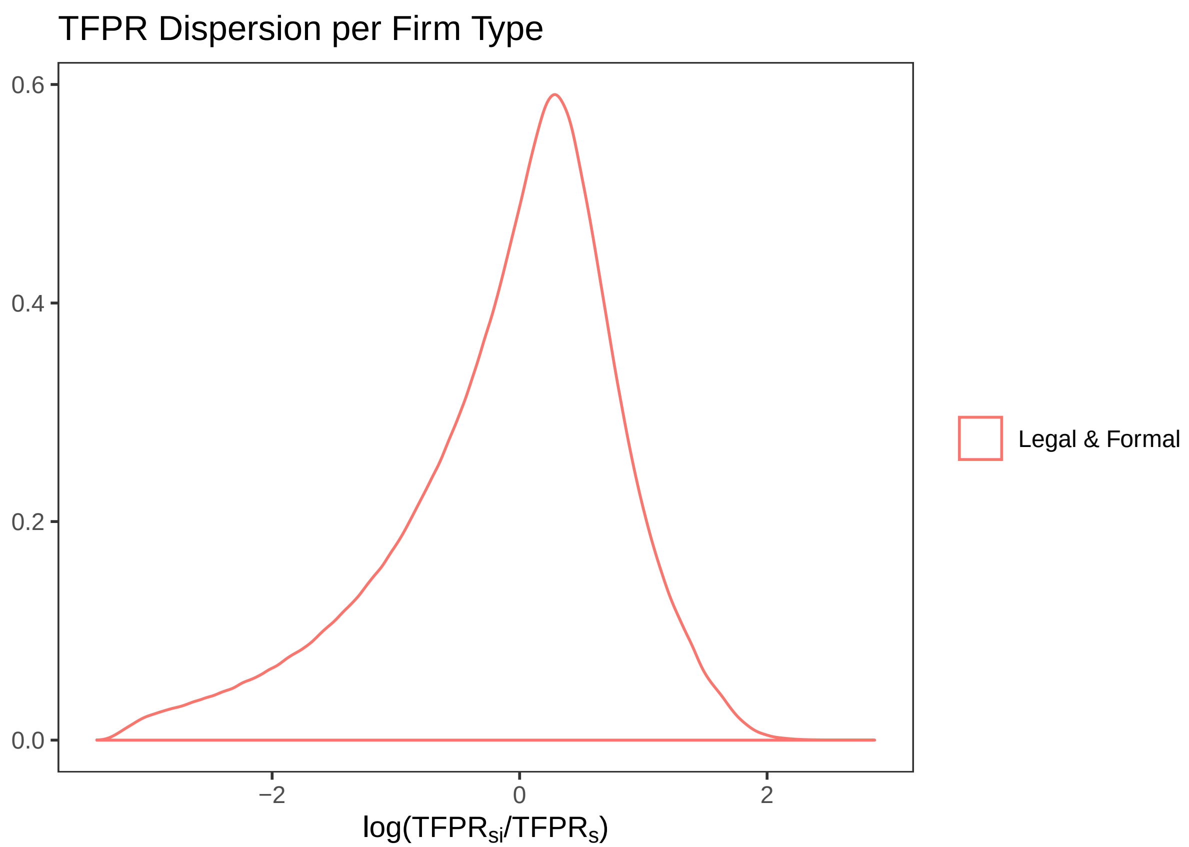
Source: Busso, Matías; Fazio, María Victoria and Levy, Santiago. 2012. "(In)Formal and (Un)Productive: The Productivity Costs of Excessive Informality in Mexico." IDB Working Paper Series No. IDB-WP-341. Washington, D.C., United States: Inter-American Development Bank.

```
firm_type <- data
firm_type$LegalityIndex <- (firm_type$J300A + firm_type$J400A)/ firm_type$J000A
firm_type$FormalityIndex <- (firm_type$J300A + firm_type$J400A)/(firm_type$J000A +
                                                                   firm_type$K610A +
                                                                   firm_type$K620A)

firm_type$FirmType <- ifelse((firm_type$LegalityIndex > .18) &&
                               (firm_type$FormalityIndex > 0.18 ), "Legal & Formal",
                             ifelse(is.na(firm_type$LegalityIndex) &&
                                      (firm_type$FormalityIndex == 0 ),
                                    "Legal & Informal",
                                    ifelse((firm_type$LegalityIndex > .18) &&
                                             (firm_type$FormalityIndex < .18 &&
                                                firm_type$FormalityIndex > 0),
                                           "Legal & Semi-formal",
                                           ifelse((firm_type$LegalityIndex < .18 &&
                                                     firm_type$LegalityIndex>0 ) &&
                                                    (firm_type$FormalityIndex< .18 &&
                                                       firm_type$FormalityIndex>0 ),
                                                  "Semi-legal & Semi-formal",
                                                  ifelse((firm_type$LegalityIndex == 0) &&
                                                           (firm_type$FormalityIndex == 0),
                                                         "Ilegal & Informal",
                                                         "Undefined")))))

firm_type_sub <- firm_type[FirmType != "Undefined", ]

## Graph densities
graphdispersionTFPQfirmtype <- ggplot(firm_type_sub, aes(x = TFPQdispersion)) +
  geom_density(aes(color = FirmType)) +
  ggtitle("TFPQ Dispersion per Firm Type") +
  labs(x = expression(paste("log(",TFPQ[si]*M[s]^{frac(1, sigma - 1)},"/",TFPQ[s],")")) ,
       y = "") +  theme(plot.title = element_text(hjust = 0.5)) + theme_bw()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionTFPRfirmtype <- ggplot(firm_type_sub, aes(x=TFPRdispersion)) +
  geom_density(aes(color = FirmType)) +
  ggtitle("TFPR Dispersion per Firm Type") +
```

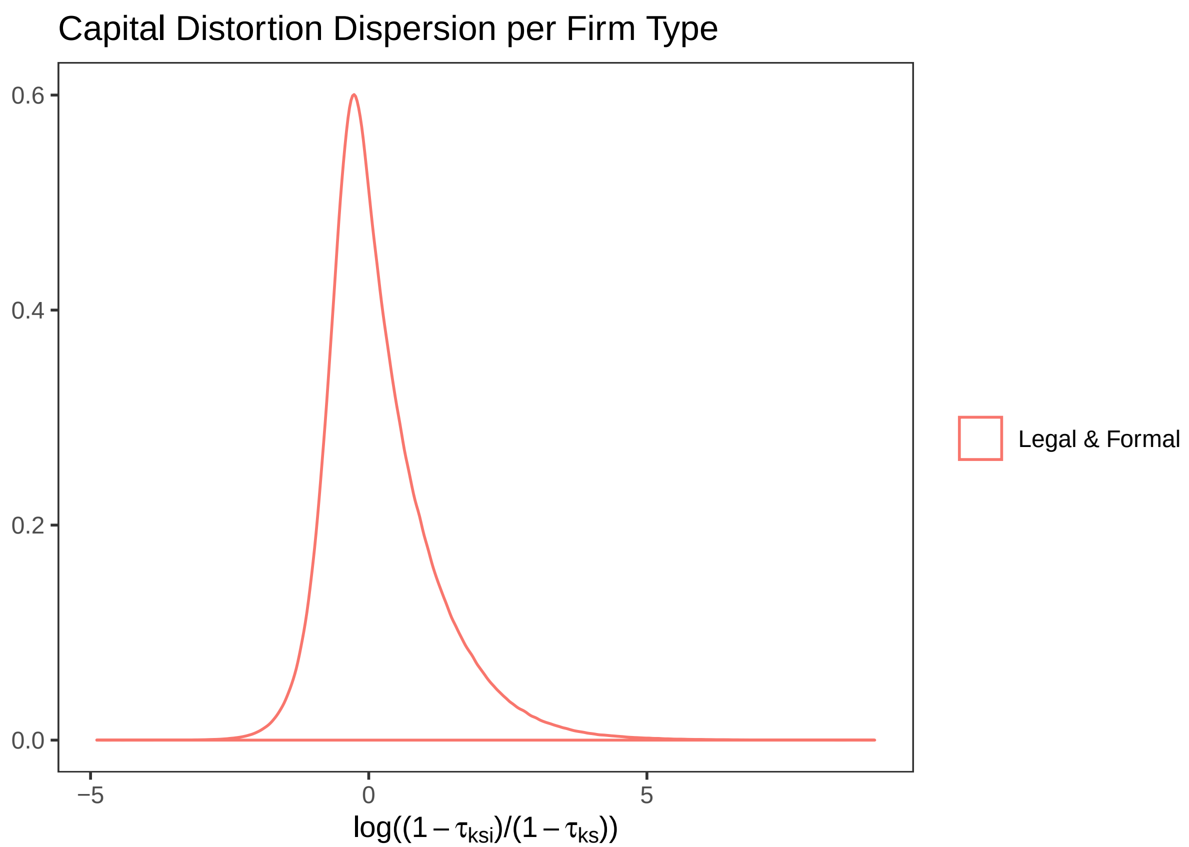
```
  labs(x = expression(paste("log(",TFPR[si],"/",TFPR[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionSKDfirmtype <- ggplot(firm_type_sub, aes(x= scaledKdistortion)) +
  geom_density(aes(color = FirmType)) +
  ggtitle("Capital Distortion Dispersion per Firm Type") +
  labs(x = expression(paste("log((",1-tau[ksi],")/(",1-tau[ks],"))")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionSYDfirmtype <- ggplot(firm_type_sub, aes(x=scaledYdistortion)) +
  geom_density(aes(color = FirmType)) +
  ggtitle("Value-added Distortion Dispersion per Firm Type") +
  labs(x = expression(paste("log((",1-tau[Ysi],")/(",1-tau[Ys],"))")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionTFPQfirmtype
```



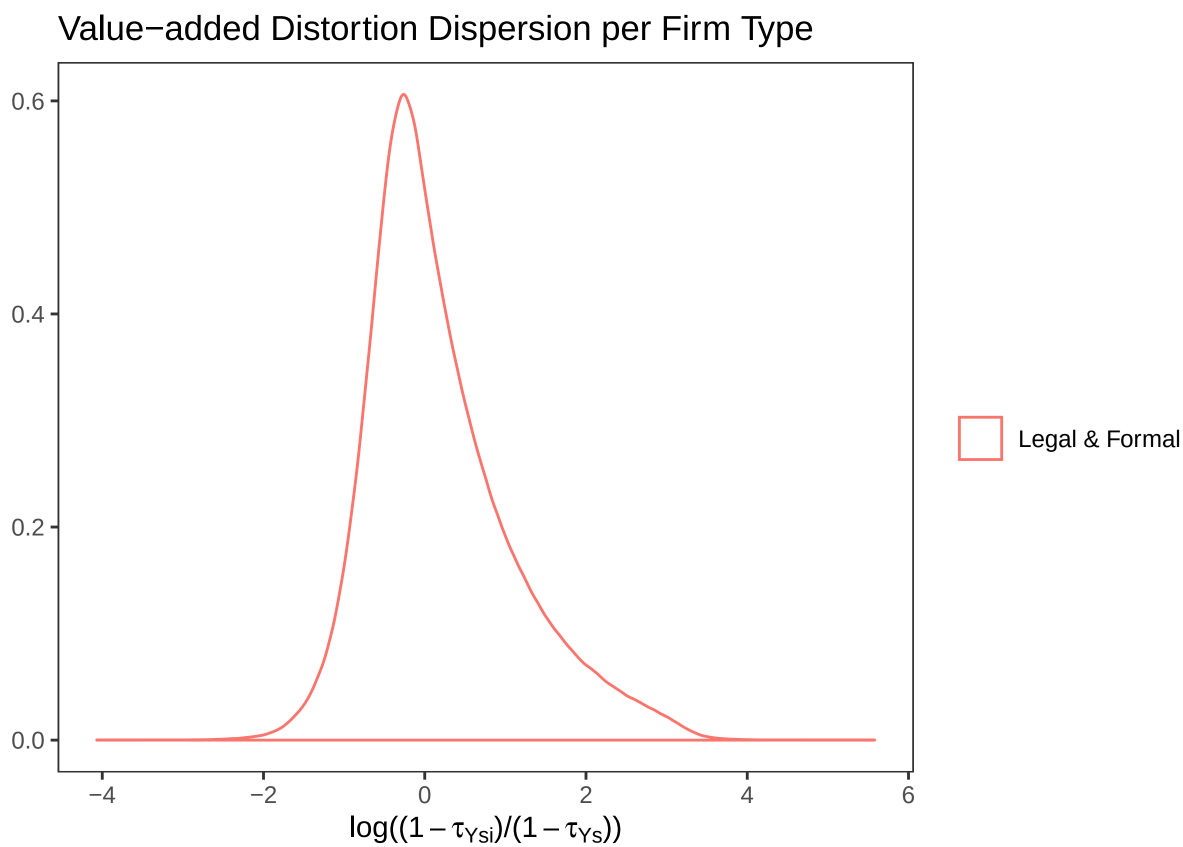

```
graphdispersionTFPRfirmtype
```




```
graphdispersionSKDfirmtype
```




```
graphdispersionSYDfirmtype
```



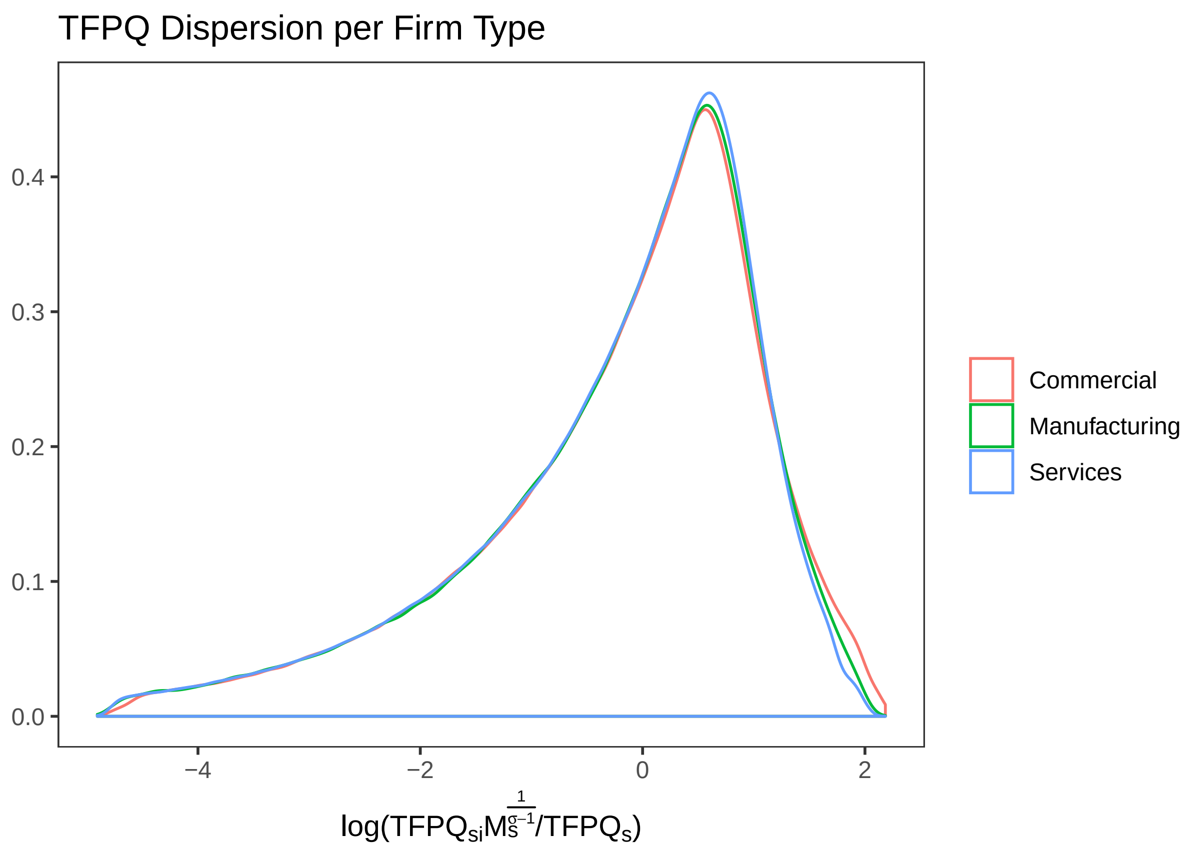

```
rm(graphdispersionTFPQfirmtype, graphdispersionTFPRfirmtype,
   graphdispersionSKDfirmtype, graphdispersionSYDfirmtype, firm_type_sub)
```

9.2 Firm economic activity

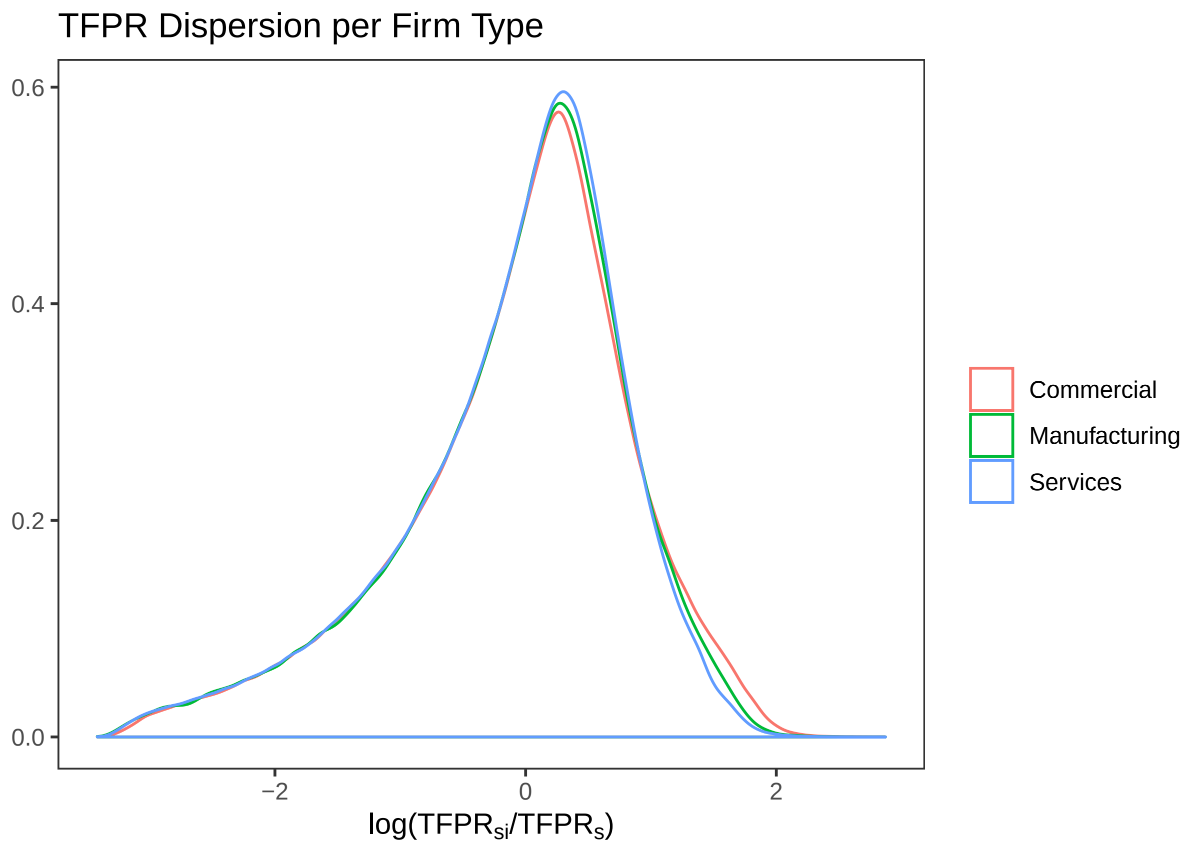
In this section, firms are classified according to their industry: Manufacturing, Services or Commercial.

```
## Add activity labels
econ_activity <- data
econ_activity$clase <- as.character(econ_activity$clase)
econ_activity$EconomicActivityIndex = substr(econ_activity$clase, 1, 2)
econ_activity <- merge(econ_activity, activity_index, by = "EconomicActivityIndex",
                       all.x = TRUE)

## Graph densities
graphdispersionTFPQactivity <- ggplot(econ_activity, aes(x = TFPQdispersion)) +
  geom_density(aes(color = Actividad)) +
  ggtitle("TFPQ Dispersion per Firm Type") +
  labs(x = expression(paste("log(",TFPQ[si]*M[s]^{frac(1, sigma - 1)},"/",TFPQ[s],")")),
       y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionTFPRactivity <- ggplot(econ_activity, aes(x = TFPRdispersion)) +
```

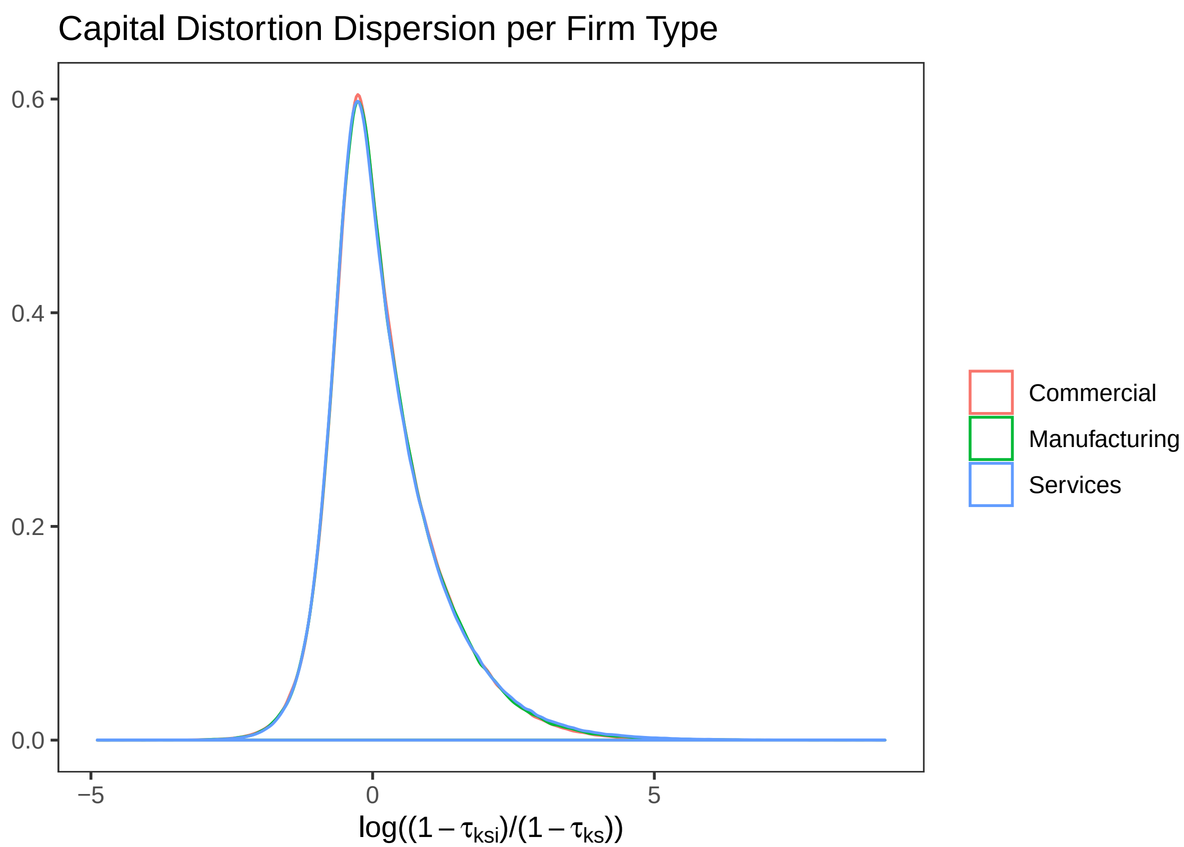
```r
  geom_density(aes(color = Actividad)) +
  ggtitle("TFPR Dispersion per Firm Type") +
  labs(x = expression(paste("log(",TFPR[si],"/",TFPR[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionSKDactivity <- ggplot(econ_activity, aes(x = scaledKdistortion)) +
  geom_density(aes(color = Actividad)) +
  ggtitle("Capital Distortion Dispersion per Firm Type") +
  labs(x = expression(paste("log((",1-tau[ksi],")/(",1-tau[ks],"))")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionSYDactivity <- ggplot(econ_activity, aes(x = scaledYdistortion)) +
  geom_density(aes(color = Actividad)) +
  ggtitle("Value-added Distortion Dispersion per Firm Type") +
  labs(x = expression(paste("log((",1-tau[Ysi],")/(",1-tau[Ys],"))")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='')

graphdispersionTFPQactivity
```



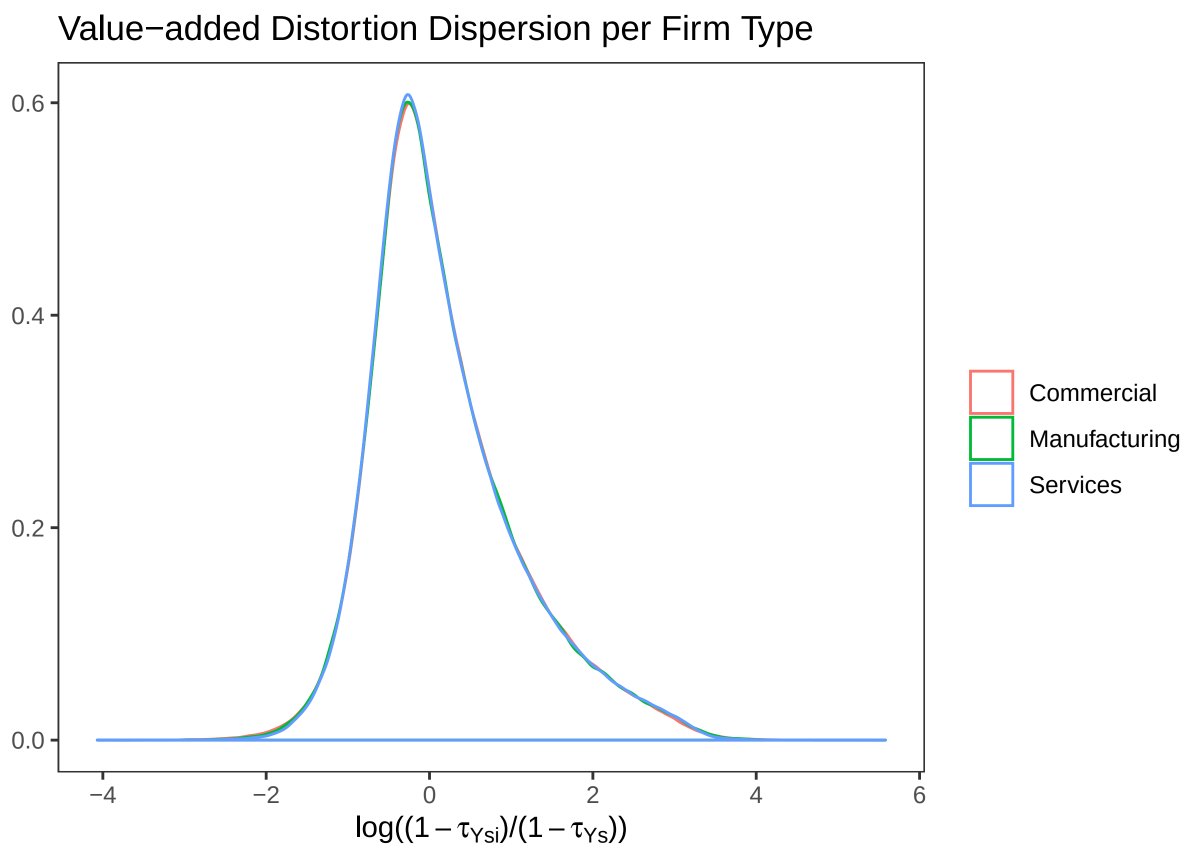
```
graphdispersionTFPRactivity
```




```
graphdispersionSKDactivity
```




```
graphdispersionSYDactivity
```




```
rm(graphdispersionTFPQactivity, graphdispersionTFPRactivity,
   graphdispersionSKDactivity, graphdispersionSYDactivity)
```

Gains by economic activity

Additionally, in this section we calculate the potential gains by economic activity, as we did previously at the state level.

```
## Calculate wsLsa, Ksa, PsYsa, TFPRsa and TFPQsa,
## where a indicates a variable is an aggregate at the activity level
gains_econ <- econ_activity
gains_econ$e03 <- as.character(gains_econ$e03)

## Summarize figures by state
## TFPRSa
gains_econ_0 <- gains_econ
gains_econ_0$count <- 1
gains_econ_0 <- gains_econ_0[, list(wsLsa = sum(wsiLsi), Ksa = sum(Ksi),
                                     PsYsa = sum(PsiYsi), alphas = mean(alphas),
                                     Msa = sum(count), sigma = mean(sigma)),
                               by = list(Actividad = Actividad, clase = clase)]
gains_econ_0$TFPRsa <- gains_econ_0$PsYsa /
  ((gains_econ_0$Ksa^gains_econ_0$alphas) *
     (gains_econ_0$wsLsa^(1 - gains_econ_0$alphas)))
gains_econ_0$PlusTksa <- ((gains_econ_0$alphas/(1 - gains_econ_0$alphas))*
```

```r
                            (gains_econ_0$wsLsa/(r*gains_econ_0$Ksa)))
gains_econ_0$MinusTYsa <- ((sigma/(sigma - 1))*
                             (gains_econ_0$wsLsa/
                                ((1 - gains_econ_0$alphas)*gains_econ_0$PsYsa)))
gains_econ_0$indicatorvariable <- paste(as.character(gains_econ_0$Actividad),
                                        as.character(gains_econ_0$clase), sep ="")
gains_econ_0 <- gains_econ_0[, c('Actividad', 'wsLsa', 'Ksa', 'PsYsa', 'TFPRsa',
                                 'indicatorvariable', 'PlusTksa', 'MinusTYsa', 'Msa')]

gains_econ$indicatorvariable <- paste(as.character(gains_econ$Actividad),
                                      as.character(gains_econ$clase), sep ="")
gains_econ <- gains_econ[, -'Actividad']

gains_econ <- merge(gains_econ, gains_econ_0, by = 'indicatorvariable',
                    all.x = TRUE)
rm(gains_econ_0)
gains_econ$TFPQsitrans <- (gains_econ$TFPQsi*gains_econ$TFPRsa /
                             gains_econ$TFPRsi)^(sigma - 1)

## TFPQsa and Asa
gains_econ_1 <- gains_econ
gains_econ_1 <- gains_econ_1[, list(TFPQssum = sum(TFPQsitrans),
                                    Assum = sum(TFPQsi^(sigma - 1)),
                                    sigma = mean(sigma)),
                             by = list(Actividad = Actividad, clase = clase)]
gains_econ_1$TFPQsa <- gains_econ_1$TFPQssum^(1/(sigma - 1))
gains_econ_1$Asa <- (gains_econ_1$Assum)^(1/(sigma - 1))
gains_econ_1$indicatorvariable <- paste(as.character(gains_econ_1$Actividad),
                                        as.character(gains_econ_1$clase), sep ="")
gains_econ_1 <- gains_econ_1[, c('TFPQsa', 'Asa', 'indicatorvariable', 'Actividad')]

gains_econ <- merge(gains_econ, gains_econ_1[, -'Actividad'],
                    by = 'indicatorvariable', all.x = TRUE)
gains_econ$TFPQdispersiona <- log(gains_econ$TFPQsi*
                                    (gains_econ$Msa^(1/(sigma - 1)))/gains_econ$TFPQsa)
gains_econ$TFPRdispersiona <- log(gains_econ$TFPRsi/gains_econ$TFPRsa)
gains_econ$scaledKdistortiona <- log(gains_econ$PlusTksi/gains_econ$PlusTksa)
gains_econ$scaledYdistortiona <- log(gains_econ$MinusTYsi/gains_econ$MinusTYsa)
rm(gains_econ_1)

## Add thetase
gains_econ_2 <- gains_econ[, list(PYa = sum(PsiYsi)), by = list(Actividad = Actividad)]

gains_econ <- gains_econ[, list(TFPQsa = mean(TFPQsa), Asa = mean(Asa),
                                averageTFPRdispersion = mean(TFPRdispersiona),
                                averageSKdistortion = mean(scaledKdistortiona),
                                sdTFPRdispersion = sd(TFPRdispersiona),
                                sdSKD = sd(scaledKdistortiona), PsYsa = sum(PsiYsi)),
                         by = list(Actividad = Actividad, clase = clase)]
gains_econ$indicatorvariable <- paste(as.character(gains_econ$Actividad),
                                      as.character(gains_econ$clase), sep ="")
gains_econ <- merge(gains_econ, gains_econ_2, by = 'Actividad', all.x = TRUE)
rm(gains_econ_2)
```

```
gains_econ$thetasa = gains_econ$PsYsa/gains_econ$PYa

## Produce final table
gains_econ$k <- (gains_econ$TFPQsa/gains_econ$Asa)^gains_econ$thetasa
gains_econ <- gains_econ[, list(activitygain = 1/prod(k)-1,
                                averageTFPRdispersion = mean(averageTFPRdispersion),
                                averageSKdistortion = mean(averageSKdistortion),
                                PYa = sum(PsYsa)), by = list(Actividad = Actividad)]

print(gains_econ)
```

```
##        Actividad activitygain averageTFPRdispersion averageSKdistortion
## 1:    Commercial    1.0180623           -0.07970483           0.1892126
## 2: Manufacturing    0.8815988           -0.09950634           0.1980532
## 3:      Services    0.8205829           -0.11335792           0.2111699
##            PYa
## 1:  6281098485
## 2:  1496760891
## 3: 16426230821
```

```
rm(gains_econ)
```

To summarize, the following table provides an overview of the distribution of establishments by economic activity and firm type.

```
### Sample description
sample <- firm_type
sample$clase <- as.character(sample$clase)
sample$EconomicActivityIndex <- substr(sample$clase, 1, 2)
rm(firm_type)

sample <- merge(sample, activity_index, by = "EconomicActivityIndex", all.x = TRUE)
K <- sum(sample$Ksi)
WL <- sum(sample$wsiLsi)
Y <- sum(sample$PsiYsi)
totalestablishments <- dim(sample)[1]

sample_description <- sample
sample_description$count <- 1
sample_description <- sample_description[, list(Firms = (sum(count)),
                                                 Firms.percent = (sum(count)/totalestablishments),
                                                 Value.Added = (sum(PsiYsi)),
                                                 Value.Added.Percent = (sum(PsiYsi)/PY),
                                                 Capital = (sum(Ksi)),
                                                 Capital.Percent = (sum(Ksi)/K),
                                                 Wage.Bill = (sum(wsiLsi)),
                                                 Wage.Bill.Percent = (sum(wsiLsi)/WL)),
                                          by = list( Actividad = Actividad,
                                                     FirmType = FirmType)]
print(sample_description)
```

```
##        Actividad       FirmType   Firms Firms.percent Value.Added
```

```
## 1: Manufacturing Legal & Formal  297282    0.06190594  1496760891
## 2:    Commercial Legal & Formal 1247300    0.25973750  6281098485
## 3:      Services Legal & Formal 3257574    0.67835655 16426230821
##    Value.Added.Percent     Capital Capital.Percent   Wage.Bill
## 1:          0.06183917  1210295966      0.06210953  4476928726
## 2:          0.25950566  5084186994      0.26090845 18741384331
## 3:          0.67865516 13191995648      0.67698202 49088574794
##    Wage.Bill.Percent
## 1:        0.06191566
## 2:        0.25919224
## 3:        0.67889210
```

```
### TFPR and TFPQ distribution measures
productivity_desc <- sample
productivity_desc <- productivity_desc[, list(TFPQMean = mean(TFPQdispersion),
                                              TFPQSD = sd(TFPQdispersion),
                                              TFPQPercentile7525 = (
                                                quantile(TFPQdispersion,
                                                         probs = 0.75) -
                                                  quantile(TFPQdispersion,
                                                           probs = 0.25)),
                                              TFPQPercentile9010 = (
                                                quantile(TFPQdispersion, probs = 0.90) -
                                                  quantile(TFPQdispersion, probs = 0.10)),
                                              TFPRMean = mean(TFPRdispersion),
                                              TFPRSD = sd(TFPRdispersion),
                                              TFPRPercentile7525 = (
                                                quantile(TFPRdispersion, probs = 0.75) -
                                                  quantile(TFPRdispersion, probs = 0.25)),
                                              TFPRPercentile9010 = (
                                                quantile(TFPRdispersion, probs = 0.90) -
                                                  quantile(TFPRdispersion, probs = 0.10))),
                                       by = list(Actividad = Actividad, FirmType = FirmType)]
## TFPR and TFPQ distribution measures table by economic activity and firm type
print(productivity_desc)
```

```
##        Actividad       FirmType   TFPQMean   TFPQSD TFPQPercentile7525
## 1: Manufacturing Legal & Formal -0.2250955 1.295535           1.579275
## 2:    Commercial Legal & Formal -0.2019967 1.301533           1.583741
## 3:      Services Legal & Formal -0.2386343 1.289153           1.578973
##    TFPQPercentile9010    TFPRMean    TFPRSD TFPRPercentile7525
## 1:           3.214014 -0.09969047 0.9172212           1.098060
## 2:           3.253287 -0.07736445 0.9331650           1.113517
## 3:           3.194720 -0.11355565 0.9065892           1.091578
##    TFPRPercentile9010
## 1:           2.306947
## 2:           2.366754
## 3:           2.272869
```

```
rm(sample)
```

10. Sensitivities

In this last section, we seek to calculate more accurate figures of productivity gains by replacing the baseline sigma (3) as per Hsieh-Klenow with data on the elasticity of substitution per subindustry. This data is only available for manufacturing subindustries, and can be accessed here. Therefore, sigma for establishments from the commercial and service sectors remains at 3. We note that, as productivity gains have a positive relationship with the elasticity of substitution in Hsieh-Klenow's model, and given that most of the elasticities available in this database are larger than 3, Hsieh-Klenow's baseline use of 3 was a conservative estimate of productivity gains. We should thus see an increase in potential gains compared to our baseline model.

```
data_sensitivities <- data_sensitivities[, c('clase', 'e03', 'PsiYsi', 'Lsi', 'ID',
                                             'alphas', 'Ksi', 'wsiLsi')]
data_sensitivities$sigma <- 3
data_sensitivities$clase <- as.character(data_sensitivities$clase)
data_sensitivities$econtype <- as.integer(substr(data_sensitivities$clase, 1, 2))
data_sensitivities$clase <- as.integer(data_sensitivities$clase)

data_sensitivities <- merge(data_sensitivities, sigmas, by.x = 'clase',
                            by.y = 'naics4', all.x = TRUE)
data_sensitivities <- merge(data_sensitivities, sigmas, by.x = 'econtype',
                            by.y = 'naics4', all.x = TRUE)
availablesigmas <- sigmas$naics4
manufacturing <- c(31, 32, 33)

data_sensitivities$sigma <- ifelse(data_sensitivities$econtype %in% manufacturing,
                                   ifelse(as.integer(data_sensitivities$clase) %in%
                                            availablesigmas, data_sensitivities$sigmaavg.x,
                                          data_sensitivities$sigmaavg.y), 3)

### Infer distortions and productivity
### For each plant
data_sensitivities$PlusTksi <- ((data_sensitivities$alphas/
                                   (1 - data_sensitivities$alphas))*
                                  (data_sensitivities$wsiLsi/
                                     (r*data_sensitivities$Ksi)))

data_sensitivities$MinusTYsi <- ((data_sensitivities$sigma/
                                    (data_sensitivities$sigma - 1))*
                                   (data_sensitivities$wsiLsi/
                                      ((1 - data_sensitivities$alphas)*
                                         data_sensitivities$PsiYsi)))

data_sensitivities$TFPRsi <- data_sensitivities$PsiYsi/
  ((data_sensitivities$Ksi^data_sensitivities$alphas)*
     (data_sensitivities$wsiLsi^(1-data_sensitivities$alphas)))
data_sensitivities$TFPQsi <- (((data_sensitivities$PsiYsi)^(data_sensitivities$sigma/
                                (data_sensitivities$sigma-1)))/
                                ((data_sensitivities$Ksi^data_sensitivities$alphas)*
                                   (data_sensitivities$wsiLsi^(1-data_sensitivities$alphas))))

### Re-calculate average productivity measures
## Calculate TFPRs
sector_table_sens <- data_sensitivities[, list(wsLs = sum(wsiLsi), Ls = sum(Lsi),
                                               Ks = sum(Ksi), PsYs = sum(PsiYsi),
```

```r
                                              alphas = mean(alphas),
                                              sigma = mean(sigma)),
                                       by = list(clase = clase)]

sector_table_sens$PlusTks <- ((sector_table_sens$alphas/(1 - sector_table_sens$alphas))*
                                (sector_table_sens$wsLs/(r*sector_table_sens$Ks)))
sector_table_sens$MinusTYs <- ((sector_table_sens$sigma/(sector_table_sens$sigma - 1))*
                                 (sector_table_sens$wsLs/
                                    ((1 - sector_table_sens$alphas)*
                                       sector_table_sens$PsYs)))
sector_table_sens$TFPRs <- sector_table_sens$PsYs/
  ((sector_table_sens$Ks^sector_table_sens$alphas)*
     (sector_table_sens$wsLs^(1 - sector_table_sens$alphas)))
sector_table_sens <- sector_table_sens[, c('clase', 'wsLs', 'Ls', 'Ks', 'PsYs', 'PlusTks',
                                           'MinusTYs', 'TFPRs')]

data_sensitivities <- merge(data_sensitivities, sector_table_sens, by = 'clase',
                            all.x = TRUE)

## Calculate TFPQs and As
data_sensitivities$TFPQsitrans <- (data_sensitivities$TFPQsi*
                                     data_sensitivities$TFPRs/
                                     data_sensitivities$TFPRsi)^(data_sensitivities$sigma - 1)
sector_table_sens_1 <- data_sensitivities[, list(TFPQssum = sum(TFPQsitrans),
                                                 Assum = sum(TFPQsi^(sigma - 1)),
                                                 sigma = mean(sigma)),
                                          by = list(clase = clase)]
sector_table_sens_1$TFPQs <- sector_table_sens_1$TFPQssum^(1/(sector_table_sens_1$sigma - 1))
sector_table_sens_1$As <- (sector_table_sens_1$Assum)^(1/(sector_table_sens_1$sigma - 1))
sector_table_sens_1 <- sector_table_sens_1[, -'sigma']

data_sensitivities <- merge(data_sensitivities, sector_table_sens_1, by = 'clase',
                            all.x = TRUE)
rm(sector_table_sens_1)

## Calculate industry shares thetas
PsYsvalues <- sector_table_sens$PsYs
PYcheck <- sum(PsYsvalues)

sector_table_sens$thetas <- sector_table_sens$PsYs/PYcheck
sector_table_sens <- sector_table_sens[, c("clase", "thetas")]

data_sensitivities <- merge(data_sensitivities, sector_table_sens, by = "clase",
                            all.x = TRUE)
rm(sector_table_sens)
```

Now that we have recalculated our average measures of productivity and distortions, we compute the dispersion of these measures. '### 10.5 Measure dispersion and distribution of TFPQ and TFPR, and scaled capital and output distortions

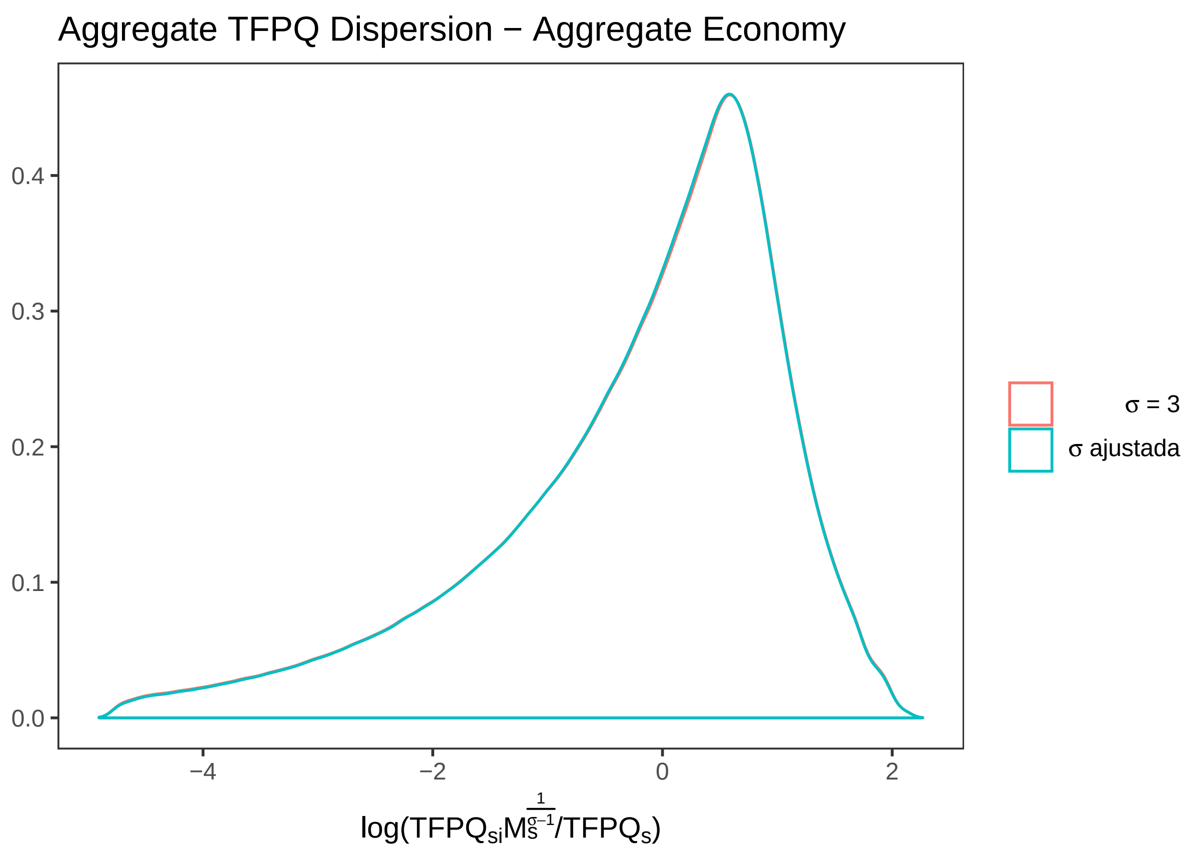
```r
## Calculate Ms
Ms_totals_sens <- data_sensitivities
Ms_totals_sens$count_naic <- 1
```

```
Ms_totals_sens <- Ms_totals_sens[, list(Ms = sum(count_naic)), by = list(clase = clase)]

data_sensitivities <- merge(data_sensitivities, Ms_totals_sens, by.x ='clase',
                            by.y = 'clase', all.x = TRUE)

data_sensitivities$TFPQdispersion <- log(data_sensitivities$TFPQsi*
                                          (data_sensitivities$Ms^(1/(data_sensitivities$sigma - 1)))
                                        /data_sensitivities$TFPQs)
data_sensitivities$TFPRdispersion <- log(data_sensitivities$TFPRsi/
                                          data_sensitivities$TFPRs)
data_sensitivities$scaledKdistortion <- log(data_sensitivities$PlusTksi/
                                             data_sensitivities$PlusTks)
data_sensitivities$scaledYdistortion <- log(data_sensitivities$MinusTYsi/
                                             data_sensitivities$MinusTYs)

data_sensitivities <- data_sensitivities[, c('clase', 'e03', 'PsiYsi', 'Lsi',
                                             'ID', 'alphas', 'Ksi', 'wsiLsi',
                                             'PlusTksi', 'MinusTYsi', 'TFPQsi',
                                             'TFPRsi', 'wsLs', 'Ls',
                                             'Ks', 'PsYs', 'PlusTks',
                                             'MinusTYs',  'TFPRs', 'TFPQs',
                                             'thetas','Ms',
                                             'TFPQdispersion', 'TFPRdispersion',
                                             'scaledKdistortion', 'scaledYdistortion',
                                             'As', 'sigma')]
```

Our next step is to re-calculate the efficient output level and potential productivity gains. We conduct this analysis for the economy as a whole, and for the manufacturing sector stand-alone.

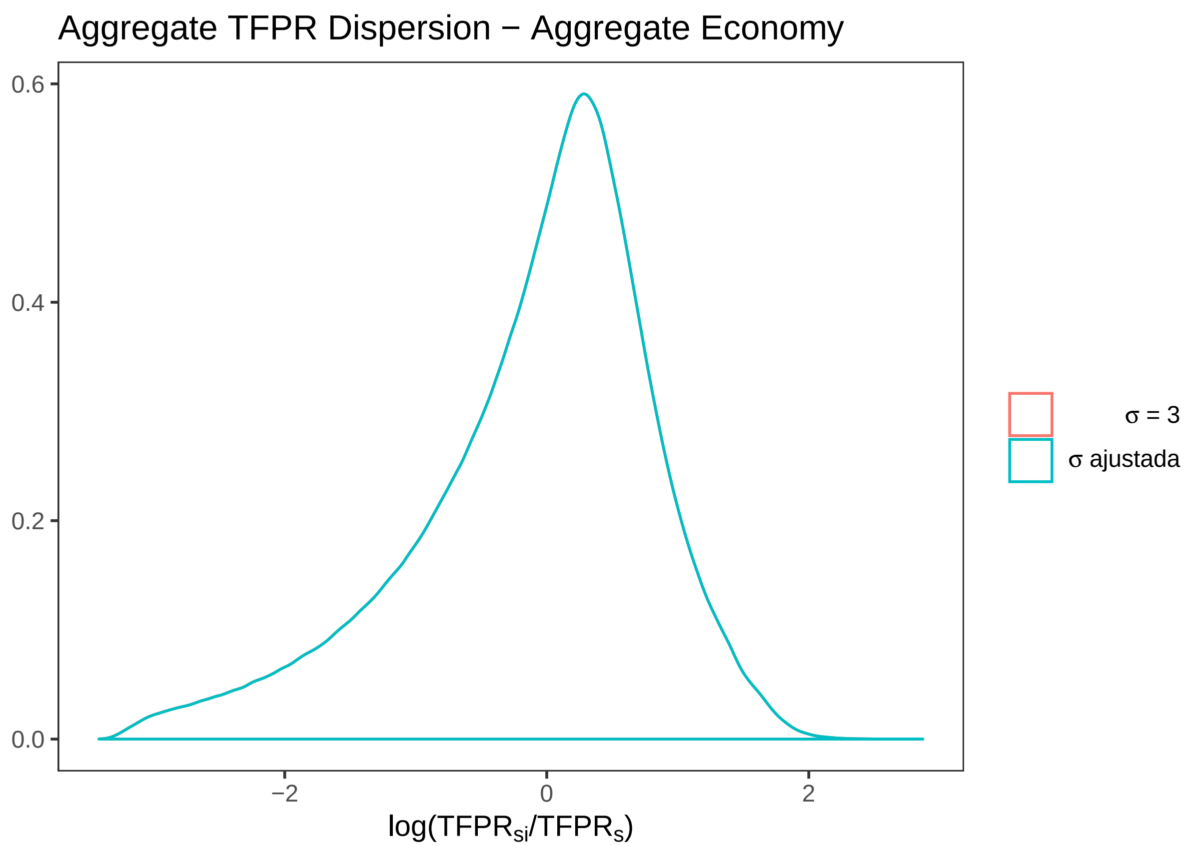
```
### Efficient output for aggregate economy
gains_sens <- data_sensitivities[, list(TFPQs = mean(TFPQs), As = mean(As),
                                        thetas = mean(thetas)),
                                 by = list(clase = clase)]
gains_sens$k <- (gains_sens$TFPQs / gains_sens$As)^gains_sens$thetas

ks_sens <- prod(gains_sens$k)
productivitygain_sens <- 1/ks_sens - 1

graphs_sens <- data_sensitivities[, c('TFPQdispersion', 'TFPRdispersion',
                                      'ID', 'clase')]
graphs_original <- data[, c('TFPQdispersion', 'TFPRdispersion', 'ID')]
joint <- merge(graphs_original, graphs_sens, by = 'ID', all.x = TRUE)
rm(graphs_original, graphs_sens)

## Aggregate economy
joint <- data.table::setnames(joint, old = c('TFPQdispersion.x', 'TFPQdispersion.y',
                                             'TFPRdispersion.x', 'TFPRdispersion.y'),
                              new = c('TFPQdispersion.Original',
                                      'TFPQdispersion.Adjusted',
                                      'TFPRdispersion.Original',
                                      'TFPRdispersion.Adjusted'))

jointTFPQ <- joint[, c('TFPQdispersion.Original', 'TFPQdispersion.Adjusted', 'ID')]
```

```
jointTFPQ <- data.table::melt(jointTFPQ, id.vars = "ID")

graphdispersionTFPQcases <- ggplot(jointTFPQ, aes(x = value)) +
  geom_density(aes(color = variable)) +
  ggtitle("Aggregate TFPQ Dispersion - Aggregate Economy") +
  labs(x = expression(paste("log(",TFPQ[si]*M[s]^{frac(1, sigma - 1)},"/",TFPQ[s],")")),
     y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='') +
  scale_color_hue(labels = c(expression(paste(sigma, " = 3") ),
                             expression(paste(sigma , " ajustada"))))

graphdispersionTFPQcases
```



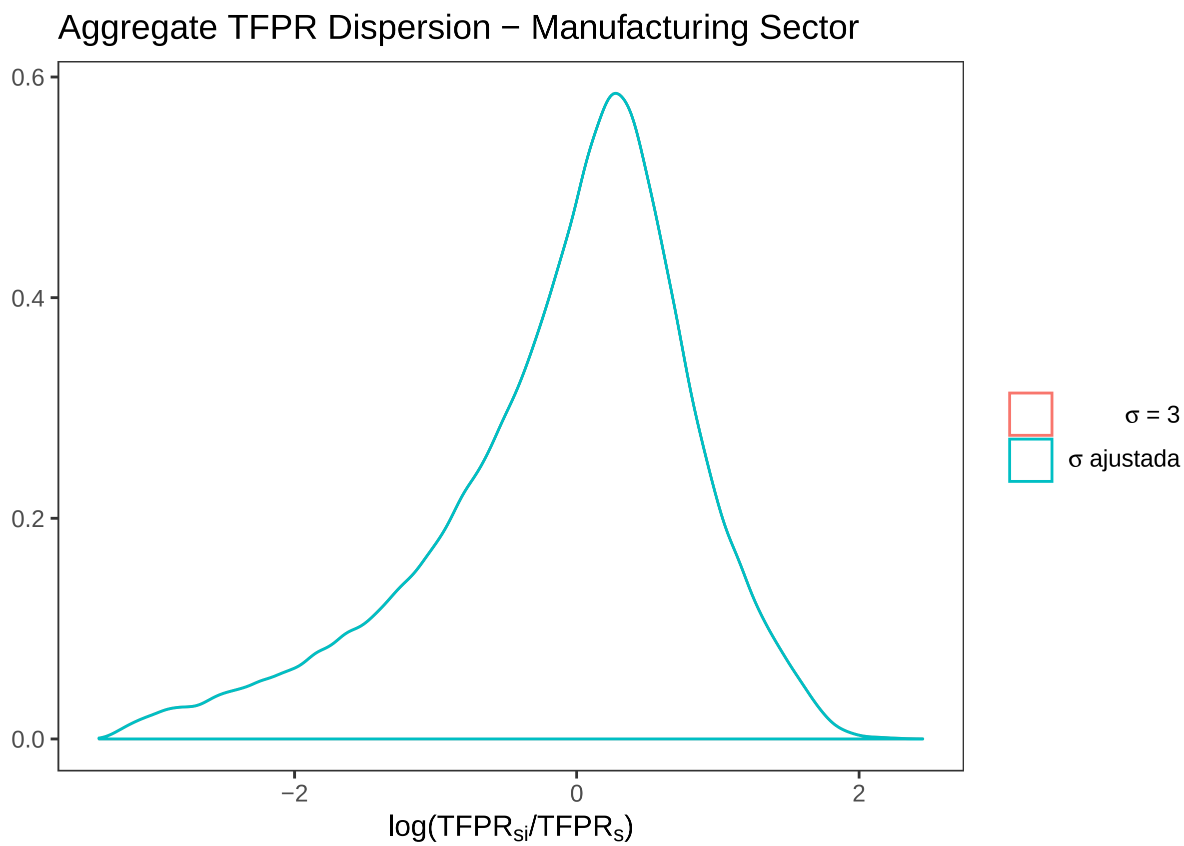

```
jointTFPR <- joint[, c('TFPRdispersion.Original', 'TFPRdispersion.Adjusted', 'ID')]
jointTFPR <- data.table::melt(jointTFPR, id.vars = "ID")

graphdispersionTFPRcases <- ggplot(jointTFPR, aes(x=value)) +
  geom_density(aes(color = variable)) +
  ggtitle("Aggregate TFPR Dispersion - Aggregate Economy") +
  labs(x =  expression(paste("log(",TFPR[si],"/",TFPR[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='') +
```

```
  scale_color_hue(labels = c(expression(paste(sigma, " = 3") ),
                             expression(paste(sigma , " ajustada"))))

graphdispersionTFPRcases
```

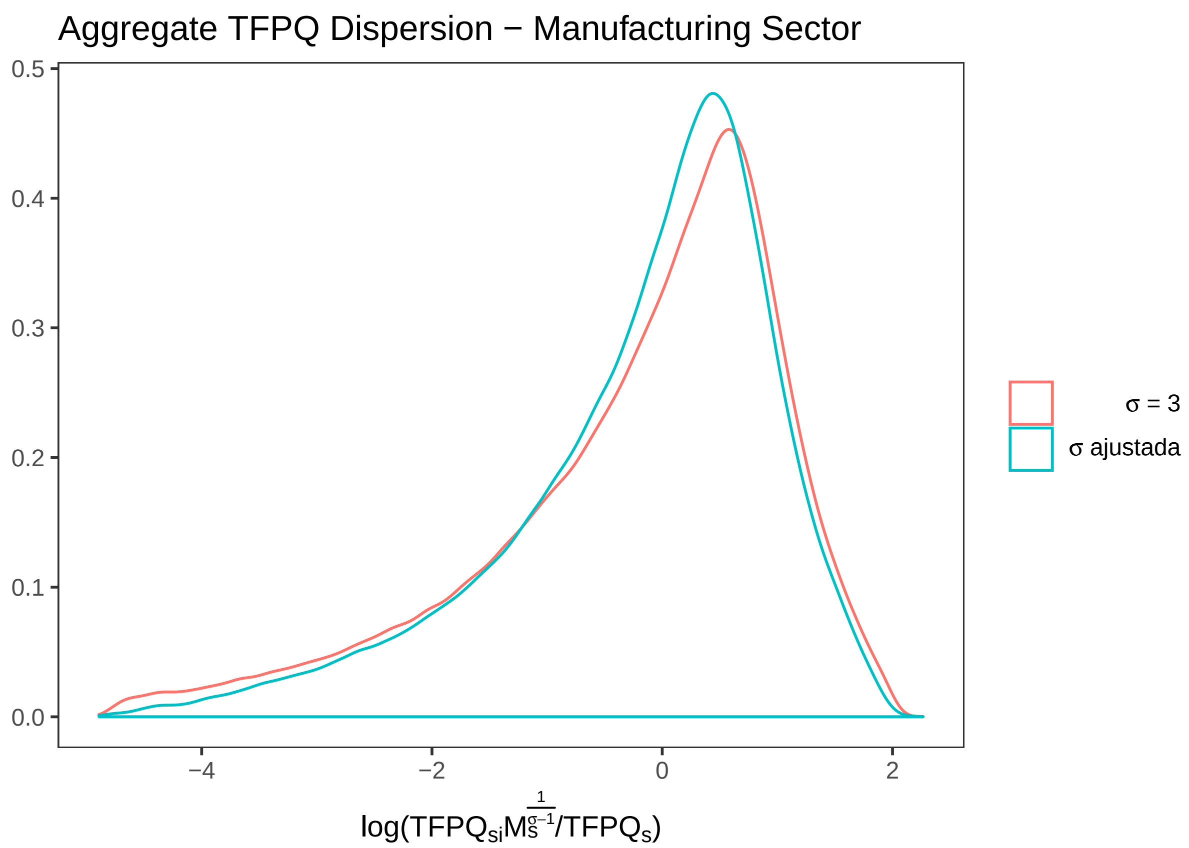



```
rm(graphdispersionTFPQcases, graphdispersionTFPRcases)

## Manufacturing only
jointm <- joint
jointm$indicator <- as.integer(substr(as.character(jointm$clase), 1, 2))
jointm <- jointm[indicator %in% c(31, 32, 33),]

jointTFPQm <- jointm[, c('TFPQdispersion.Original', 'TFPQdispersion.Adjusted', 'ID')]
jointTFPQm <- data.table::melt(jointTFPQm, id.vars = "ID")

graphdispersionTFPQcasesm <- ggplot(jointTFPQm, aes(x=value)) +
  geom_density(aes(color = variable)) +
  ggtitle("Aggregate TFPQ Dispersion - Manufacturing Sector") +
  labs(x = expression(paste("log(",TFPQ[si]*M[s]^{frac(1, sigma - 1)},"/",TFPQ[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='') +
  scale_color_hue(labels = c(expression(paste(sigma, " = 3") ),
                             expression(paste(sigma , " ajustada"))))
```

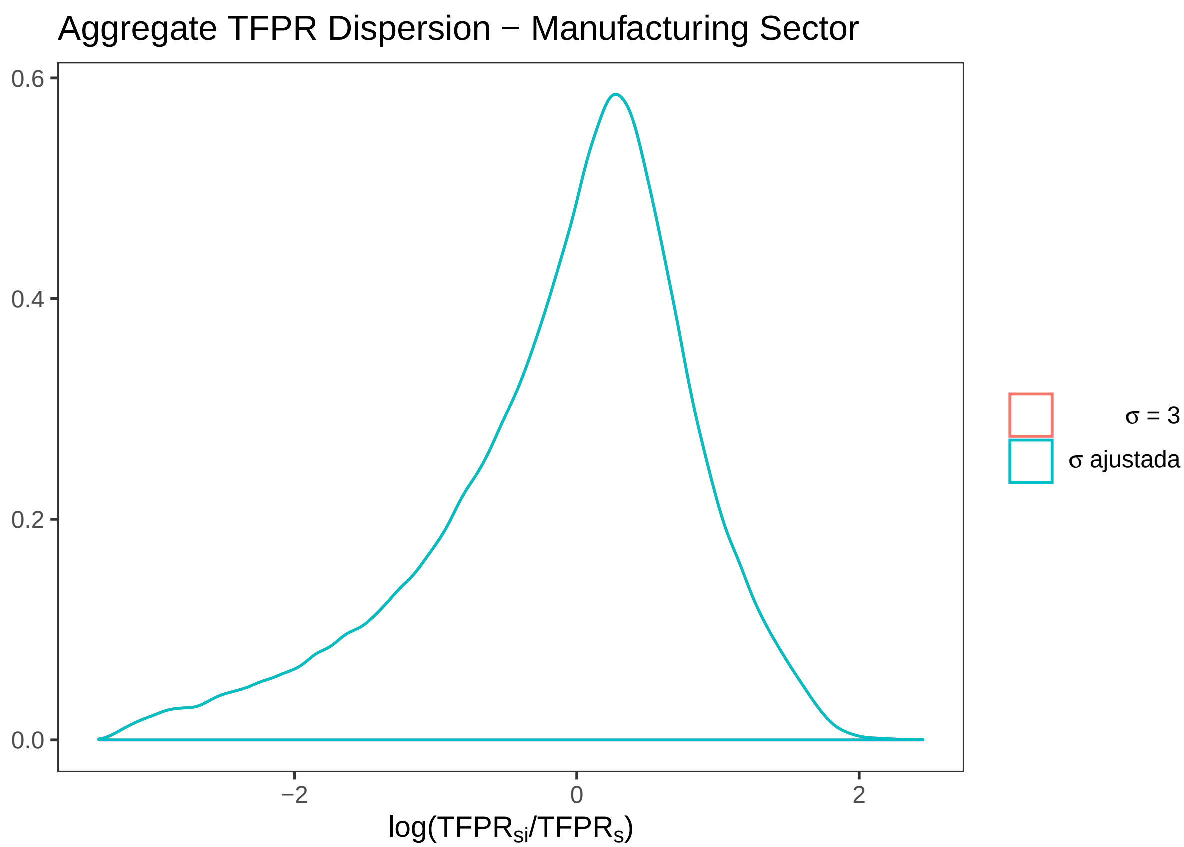
```r
jointTFPRm <- jointm[, c('TFPRdispersion.Original', 'TFPRdispersion.Adjusted', 'ID')]
jointTFPRm <- data.table::melt(jointTFPRm, id.vars = "ID")

graphdispersionTFPRcasesm <- ggplot(jointTFPRm, aes(x=value)) +
  geom_density(aes(color = variable)) +
  ggtitle("Aggregate TFPR Dispersion - Manufacturing Sector") +
  labs(x =  expression(paste("log(",TFPR[si],"/",TFPR[s],")")), y = "") +
  theme(plot.title = element_text(hjust = 0.5)) + theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(color='') +
  scale_color_hue(labels = c(expression(paste(sigma, " = 3") ),
                             expression(paste(sigma , " ajustada"))))
graphdispersionTFPRcasesm
```




```r
graphdispersionTFPQcasesm
```




```
graphdispersionTFPRcasesm
```




```
rm(graphdispersionTFPQcasesm, graphdispersionTFPRcasesm)
```

Below, we calculate and present a summary of the output gains for both the aggregate economy and the manufacturing industry.

```
print('Output gains - Aggregate economy')
```

```
## [1] "Output gains - Aggregate economy"
```

```
## Avg sigma economy
data_sensitivities$weight <- data_sensitivities$PsiYsi/PY
data_sensitivities$weightedsigma <- data_sensitivities$weight*data_sensitivities$sigma

sigmaactual <- data_sensitivities$weightedsigma
sigmapromedio <- sum(sigmaactual)

## Avg sigma manufacturing
data_sens_m <- data_sensitivities
data_sens_m$indicator <- as.integer(substr(as.character(data_sens_m$clase), 1, 2))
data_sens_m <- data_sens_m[indicator %in% c(31, 32, 33),]
PYm<- sum(data_sens_m$PsiYsi)

data_sens_m$weight <- data_sens_m$PsiYsi/PYm
data_sens_m$weightedsigma <- data_sens_m$weight*data_sens_m$sigma
```

```r
sigmaactualm <- data_sens_m$weightedsigma
sigmapromediomanufactura <- sum(sigmaactualm)

ganancias<- data.frame(Ganancias.producto = c(productivitygain, productivitygain_sens),
                       Average.Sigma.Economy = c(3, sigmapromedio),
                       Average.Sigma.Manufacturing = c(3, sigmapromediomanufactura),
                       row.names = c("sigma3", "sigmaajustada"))

print(ganancias)
```

```
##                Ganancias.producto Average.Sigma.Economy
## sigma3                  0.8737081              3.000000
## sigmaajustada           0.9032631              3.150404
##                Average.Sigma.Manufacturing
## sigma3                            3.000000
## sigmaajustada                     5.432178
```

Sources

Abramovitz, Moses. 1956. "Resource and Output Trends in the United States since 1870." NBER Occasional Paper 52. Cambridge, Massachussetts, United States: National Bureau of Economic Research.

Busso, Matías; Madrigal, Lucía y Pagés, Carmen. 2012. "Productivity and Resource Misallocation in Latin America." IDB Working Paper Series No. IDB-WP-306. Washington, D.C., United States: Banco Interamericano de Desarrollo.

Busso, Matías; Fazio, María Victoria y Levy, Santiago. 2012. "(In)Formal and (Un)Productive: The Productivity Costs of Excessive Informality in Mexico." IDB Working Paper Series No. IDB-WP-341. Washington, D.C., United States: Banco Interamericano de Desarrollo.

Chen, Edward. 1997. "The Total Factor Productivity Debate: Determinants of Economic Growth in East Asia". Asian-Pacific Economic Literature, Vol. 11, No. 1. Hong Kong, China: Lingnan College.

Frija, Aymen; Dhehibi, Boubaker; Aw-Hassan, Aden; Akroush, Samia e Ibrahim, Ali. 2015. "Approaches to Total Factor Productivity Measurements in the Agriculture Economy". Research Program on Dryland Systems. Montpellier, France: CGIAR.

Griliches, Zvi. 1995. "The Discovery of the Residual: An Historical Note". NBER Working Paper No. 5348. Cambridge, Massachussetts, United States: National Bureau of Economic Research.

Haltiwanger, John; Kulick, Robert y Syverson, Chad. 2018. "Misallocation Measures: The Distortion That Ate the Residual." NBER Working Paper No. 24199. Cambridge, Massachussetts, United States: National Bureau of Economic Research.

Hopenhayn, Hugo y Rogerson, Richard. 1993. "Job Turnover and Policy Evaluation: A General Equilibrium Analysis." Journal of Political Economy, Vol. 101, No. 5. Chicago, United States: University of Chicago.

Hsieh, Chang-Tai y Klenow, Peter J. 2009a. "Misallocation and Manufacturing TFP in China and India." The Quarterly Journal of Economics, Vol. 124, Issue 4, pp. 1403-1448.

Hsieh, Chang-Tai y Klenow, Peter J. 2009b. "Resource Misallocation in Mexico." Washington, D.C., United States: Inter-American Development Bank. Unpublished manuscript.

Hulten, Charles. 2009. "Growth Accounting". NBER Working Paper No. 15341. Cambridge, Massachussetts, United States: National Bureau of Economic Research.

Instituto Nacional de Estadística y Geografía. 2013. "Sistema de Clasificación Industrial de América del Norte, México. SCIAN 2013" Aguascalientes, Mexico: National Institute of Statistics and Geography.

Instituto Nacional de Estadística y Geografía. 2015. "Censos Económicos 2014: Metodología." Aguascalientes, Mexico: National Institute of Statistics and Geography.

Isaksson, Anders. 2007. "Determinants of total factor productivity: a literature review." Research and Statistics Branch Staff Working Paper 02/2007. Vienna, Austria: United Nations Industrial Development Organization.

Kendrick, John. 1956. "Productivity Trends: Capital and Labor". Occasional Paper 53. Cambridge, Massachussetts, United States: National Bureau of Economic Research.

Levy, Santiago. 2017. "Esfuerzos mal recompensados: La elusiva búsqueda de la prosperidad en México." Washington, D.C., United States: Inter-American Development Bank.

Mayorga Garrido Cortés, Joaquin. 2017. "Resource Misallocation in Mexico." Ciudad de México, Mexico: Centro de Investigación y Docencia Económicas.

Melitz, Marc J. 2003. "The Impact of Trade on Intra-Industry Reallocations and Aggregate Industry Productivity." Econometrica, Vol. 71, No. 6. Pp. 1695-1725.

Misch, Florian y Saborowski, Christian. 2018. "Resource Misallocation and Productivity: Evidence from Mexico." IMF Working Paper WP/18/112. Washington, D.C., United States: International Monetary Fund.

MIT OpenCourseWare. 2013. "Problem Set 3: Hsieh and Klenow (QJE 2009)." 14.772 Development Economics: Macroeconomics, Spring 2013. Cambridge, Massachussetts, Estados Unidos: MIT OpenCourseWare.

Organización de las Naciones Unidas. 2008. "International Standard Industrial Classification of All Economic Activities (ISIC), Rev. 4." Statistical Papers Series M No. 4, Rev. 4. Nueva York, United States: United Nations Organization.

Restuccia, Diego y Rogerson, Richard. 2008. "Policy Distortions and Aggregate Productivity with Heterogenous Plants." Review of Economic Dynamics, Vol. 11(4), pp. 707-720.

Restuccia, Diego y Rogerson, Richard. 2013. "Misallocation and Productivity." Review of Economic Dynamics, Vol. 16, Issue 1, pp. 1-230.

Restuccia, Diego y Rogerson, Richard. 2017. "The Causes and Costs of Misallocation." Journal of Economic Perspectives, Vol. 31, No. 3, pp. 151-174.

Schelkle, Thomas. 2016. "General Methods for Measuring Factor Misallocation." Working Paper Series in Economics No. 87. Colonia, Germany: Universidad of Cologne.

Solow, Robert. 1957. "Technical Change and the Aggregate Production Function." The Review of Economics and Statistics, Vol. 39, No. 3. Pp. 312-320.

Willis, Jack. 2015. "Hsieh and Klenow (2009): Misallocation and Manufacturing TFP in China and India." Ec 2390: Notes Section 3. Cambridge, Massachussetts, United States: Harvard Canvas. Unpublished.