The Granular Origins of Aggregate Fluctuations: Empirical Work

Load and Format Data

Alex Chinco

October 12, 2010

This code loads and formats the data used in Xavier Gabaix's 2010 Econometrica article, The Granular Origins of Aggregate Fluctuations.

The code is organized as follows. In the first section below I load all of the raw data used in the analysis. Next, in section 2 I format and save the macroeconomic data. Finally, in section 3 I format and save the COMPUSTAT data.

I use the `R` packages listed below.

```
> rm(list = ls())
> library(foreign)
> library(matlab)
> library(xtable)
> library(reshape)
> library(ggplot2)
> library(plyr)
> library(plm)
> library(graphics)
```

These are the directories I use.

```
> HOME_DIRECTORY <- "~/Dropbox/raWork/granularOrigins/Granular-FinalMaterials/empiricalResults/"
> FIGURES_DIRECTORY <- "~/Dropbox/raWork/granularOrigins/Granular-FinalMaterials/empiricalResults/figures/"
> DATA_DIRECTORY <- "~/Dropbox/raWork/granularOrigins/Granular-FinalMaterials/data/"
```

1 Load Raw Data

In this section I load each of the raw datasets used in the later analysis. The data come from 3 general sources. First, I get information on firm sales and number of employees from the COMPUSTAT database. Next, I use macroeconomic data put together by Fernando Duarte for an earlier version of the paper. Finally, I use raw data from each of the sources used by Fernando in order to update the data series through 2008.

Load Raw COMPUSTAT Data

The COMPUSTAT data come from the Fundamentals Annual section of the COMPUSTAT, North America database on WRDS. The consist of year by firm observations from 1950 to 2008 of the following variables: SIC code (SIC), net sales in \$MM (DATA 12), employees in M (DATA 29). I exclude foreign firms based in the US restricting the dataset to firms whose 'fic' and 'loc' codes are equal to 'USA'.

An important caveat is in order for U.S. firms. With Compustat, the sales of Ford, say, represent the worldwide sales of Ford, not directly the output produced by Ford in the U.S. There is no simple solution to this problem, especially if one wants a long time series. An important task of future research is to provide a version of Compustat that corrects for multinationals.

After I load the data, I plot the number of firms in the database over time split by whether or not the firm year observations are missing sales or employee count data.

```
> CompustatData <- read.csv(paste(DATA_DIRECTORY, "compustatData.csv", sep = ""), stringsAsFactors = FALSE)
> CompustatData <- CompustatData[, c("gvkey", "fyear", "tic", "conm", "emp", "sale", "sic")]
> names(CompustatData) <- c("id", "year", "ticker", "companyName", "numberOfEmployees", "sales", "sicCode")
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_NumberOfFirmsInRawCompustatDataWithValidSalesVolumeAndNumberOfEmployeesData.pdf", sep = ""),
      height = 5, width = 9)
> PlotData <- ddply(CompustatData, c("year"), function(X) c(dim(X)[1], dim(X[!is.na(X$sales), ])[1], dim(X[!is.na(X$numberOfEmployees),
      ])[1], dim(X[!is.na(X$numberOfEmployees) & !is.na(X$sales), ])[1]))
> names(PlotData) <- c("year", "All", "Valid Sales", "Valid Employees", "No Valid Info")
> PlotData <- PlotData[!is.na(PlotData$year), ]
> PlotData <- melt(PlotData, id = c("year"))
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable, colour = variable, linetype = variable))
> p <- p + geom_path()
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> save(CompustatData, file = paste(DATA_DIRECTORY, "RawCompustatData_10Oct2010.Rdata", sep = ""))
```

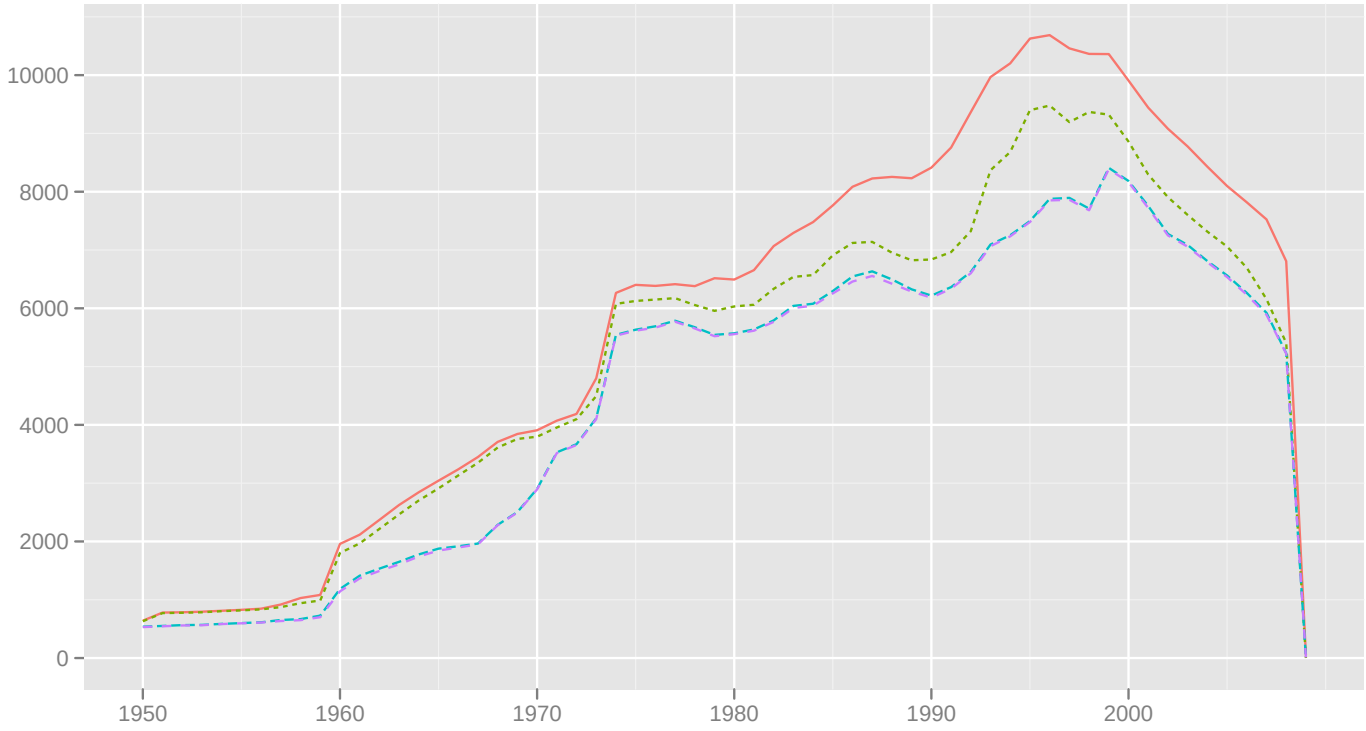


Figure 1: ***Number of Firms in the Raw COMPUSTAT Database with Valid Observations****: This figure shows the number of firms listed in the COMPUSTAT database over time broken into groups of all firms (red, solid), firms with valid sales data (green, dotted), firms with valid employees data (dashed, blue) and firms with valid sales and employees data (dashed, purple).*

Load Macroeconomic Data from Fernando Duarte

Next, I load up the macroeconomic data created by Fernando Duarte. The real GDP, GDP per capita and inflation index data series all come from the Bureau of Economic Analysis. The Solow residual is the multifactor productivity of the private business sector reported by the Bureau of Labor Studies. The data for the Romer and Romer (2004) monetary policy shocks come from David Romer's web page. Their original series (RESID) is monthly, from 1969 to 1996. Here the yearly Romer-Romer shock is the sum of the 12 monthly shocks in that year. The data for the Hamilton (2003) oil shocks primarily come from James Hamilton's web page. This series is quarterly and runs until 2001. It is defined as the amount by which the current oil price exceeds the maximum value over the past year. This paper's yearly shock is the sum of the quarterly Hamilton shocks.

Below, I load the macroeconomic data from Fernando's STATA .dta file, rename the variables, and then plot and save the data as an `Rdata` file. I divide the `realGdp` data series by 1000 in order to put it into the same units as the `sales` data from COMPUSTAT. I also divide the inflation series by 100 to convert it to a fraction.

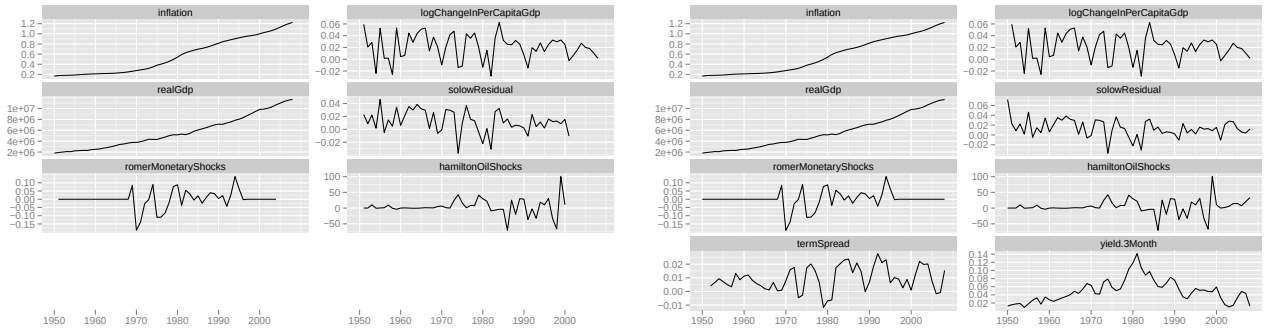
```
> MacroeconomicData <- read.dta(paste(DATA_DIRECTORY, "MacroVars.dta", sep = ""), convert.factors = FALSE)
> MacroeconomicData <- MacroeconomicData[, c("yeara", "deflator", "PerCapGDP_grw", "RealGDP", "solow", "rr_y", "oil_raw_y")]
> names(MacroeconomicData) <- c("year", "inflation", "logChangeInPerCapitaGdp", "realGdp", "solowResidual", "romerMonetaryShocks", "hamiltonOilShocks")
> MacroeconomicData$realGdp <- MacroeconomicData$realGdp * 1000
> MacroeconomicData$inflation <- MacroeconomicData$inflation/100
> pdf(paste(FIGURES_DIRECTORY, "TimeSeriesFacetWrap_MacroeconomicDataFromFernando.pdf", sep = ""), height = 5, width = 9)
> PlotData <- melt(MacroeconomicData, c("year"))
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable))
> p <- p + geom_path()
> p <- p + facet_wrap(~variable, ncol = 2, nrow = 4, scales = "free_y")
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> save(MacroeconomicData, file = paste(DATA_DIRECTORY, "RawMacroeconomicData_10Oct2010.Rdata", sep = ""))
```

Load Macroeconomic Data from Source

Next, since Fernando created the original data in 2006 or 2007, the data in his `dta` file only reaches 2005. As a result, I go to each one of his data sources and update the data series to 2008 to match the most recent vintage of the COMPUSTAT data.

In addition to updating the existing series, I also pull interest rate data from the WRDS database.

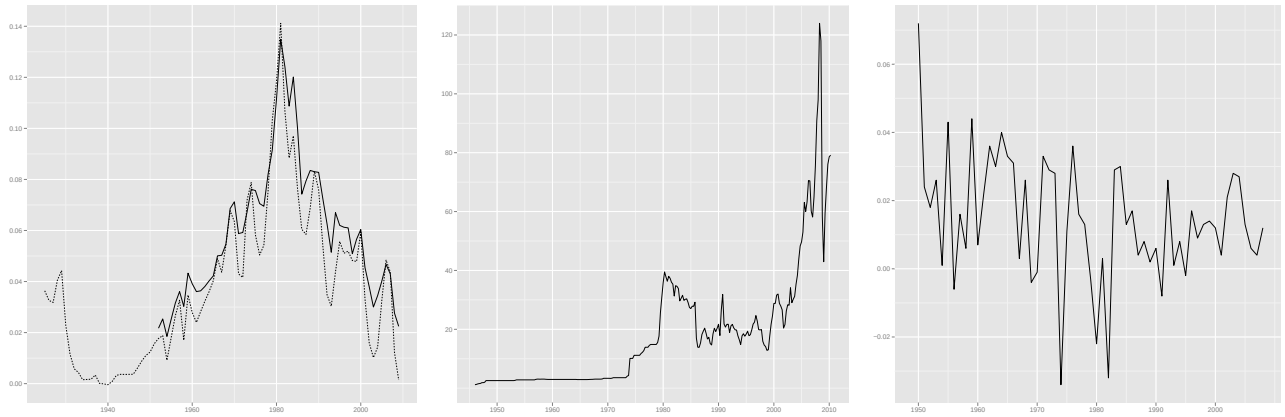
I use the spot price for oil reported by the St. Louis Federal Reserve to extend the Hamilton oil shock series to the present. I plot the oil price data using the function below.



(a) Raw

(b) Clean

Figure 2: ***Raw and Clean Macroeconomic Data****: This figure displays the raw macroeconomic data inherited from Fernando Duarte against the cleaned and updated series. Both datasets are at the yearly frequency. The original data from Fernando did not contain interest rate or term spread data.*



(a) Interest Rates

(b) Oil Price

(c) Solow Residual

Figure 3: ***Raw Data to Extend Macro Series****: Panel A. This panel houses the risk free rate (dotted line) and yield on a 5 year Fama-Bliss discount bond (solid). The data are yearly. The numbers represent the data series endpoints. Panel B. This panel houses the oil spot price at the quarterly frequency reported by the St. Louis Fed. Panel C. This panel houses the Solow Residual reported by the BLS. The data are yearly.*

```
> OilPriceData <- read.csv(paste(DATA_DIRECTORY, "oilPrice.csv", sep = ""), stringsAsFactors = FALSE)
> OilPriceData$year <- as.numeric(substring(OilPriceData$DATE, 1, 4))
> OilPriceData$month <- as.numeric(substring(OilPriceData$DATE, 6, 7))
> OilPriceData$quarter <- floor((OilPriceData$month - 1)/3) + 1
> OilPriceData <- ddply(OilPriceData, c("year", "quarter"), function(X) mean(X$VALUE, na.rm = TRUE))
> names(OilPriceData) <- c("year", "quarter", "price")
> OilPriceData$year <- as.numeric(as.character(OilPriceData$year))
> OilPriceData$quarter <- as.numeric(as.character(OilPriceData$quarter))
> OilPriceData$dt <- with(OilPriceData, year + (quarter - 1)/4)
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_OilPriceDataFromFred.pdf", sep = ""), height = 9, width = 9)
> PlotData <- OilPriceData[, c("dt", "price")]
> PlotData <- melt(PlotData, id = "dt")
> p <- ggplot(PlotData, aes(x = dt, y = value, group = variable))
> p <- p + geom_path()
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> save(OilPriceData, file = paste(DATA_DIRECTORY, "RawOilPriceData_10Oct2010.Rdata", sep = ""))
```

After, this I load the Solow residual data. The BEA report these data. I divide the reported value by 100 to convert it to a fraction in order to match the rest of the data.

```
> SolowResidualData <- read.csv(paste(DATA_DIRECTORY, "solowResidualBLS.csv", sep = ""), stringsAsFactors = FALSE)
> names(SolowResidualData) <- c("year", "solowResidual")
> SolowResidualData$solowResidual <- SolowResidualData$solowResidual/100
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_SolowResidualDataFromTheBEA.pdf", sep = ""), height = 9, width = 9)
> PlotData <- SolowResidualData[, c("year", "solowResidual")]
> PlotData <- melt(PlotData, id = "year")
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable))
> p <- p + geom_path()
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> save(SolowResidualData, file = paste(DATA_DIRECTORY, "RawSolowResidualData_10Oct2010.Rdata", sep = ""))
```

The term spread and real interest rate data are created using the specification found in Ang, Piazzesi and Wei (2006). The 5 year yield is the "Yield for 5 Year Artificial Security" data series reported in the Fama-Bliss Discount Bond Yields section of the CRSP Monthly Treasury database on WRDS. The 3 month risk free rate is the "Yield Based on 3 Month Average" data series reported in the Fama Risk Free Rates section of the CRSP Monthly Treasury database on WRDS. The term spread is the difference between these two series. The short run interest rate only goes back to 1952, so any regression involving the term spread or interest rate will be capped in the time series at 1952.

```
> FamaBlissDiscountBondYield <- read.csv(paste(DATA_DIRECTORY, "famaBliss5Year.csv", sep = ""), stringsAsFactors = FALSE)
> FamaBlissDiscountBondYield$year <- floor(FamaBlissDiscountBondYield$qdate/10000)
> FamaBlissDiscountBondYield <- ddply(FamaBlissDiscountBondYield, .(year), function(X) mean(X$yield5/100, na.rm = TRUE))
> names(FamaBlissDiscountBondYield) <- c("year", "yield.5Year")
> FamaRiskFreeRate <- read.csv(paste(DATA_DIRECTORY, "famaRiskFree3Month.csv", sep = ""), stringsAsFactors = FALSE)
> FamaRiskFreeRate$year <- floor(FamaRiskFreeRate$qdate/10000)
> FamaRiskFreeRate <- ddply(FamaRiskFreeRate, .(year), function(X) mean(X$ave_3/100, na.rm = TRUE))
> names(FamaRiskFreeRate) <- c("year", "yield.3Month")
> InterestRateData <- merge(FamaBlissDiscountBondYield, FamaRiskFreeRate, by = "year", all = TRUE)
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_5YearAnd3MonthTreasuryYieldFromCrsp.pdf", sep = ""), height = 9, width = 9)
> PlotData <- melt(InterestRateData, c("year"))
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable, linetype = variable))
> p <- p + geom_path()
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> InterestRateData$termSpread <- with(InterestRateData, yield.5Year - yield.3Month)
> InterestRateData <- InterestRateData[, c("year", "termSpread", "yield.3Month")]
> save(InterestRateData, file = paste(DATA_DIRECTORY, "RawInterestRateData_10Oct2010.Rdata", sep = ""))
```

2 Clean Macroeconomic Data

In this section I format the raw macroeconomic data, update the values to 2008 and save a cleaned version.

Update Solow Residual Series

I march through all of the years in the macroeconomic data I got from Fernando for which I have updated Solow residual data from the BEA and look for empty observations in the data from Fernando. For these years I fill in the value from the BEA.

```
> ListOfYearsWithSolowResidualData <- unique(MacroeconomicData$year[MacroeconomicData$year %in% SolowResidualData$year])
> for (YEAR in ListOfYearsWithSolowResidualData) {
      if (is.na(MacroeconomicData[MacroeconomicData$year == YEAR, ]$solowResidual)) {
          if (!is.na(SolowResidualData[SolowResidualData$year == YEAR, ]$solowResidual)) {
              MacroeconomicData[MacroeconomicData$year == YEAR, ]$solowResidual <- SolowResidualData[SolowResidualData$year == YEAR, ]$solowResidual
          }
      }
  }
```

Update Romer and Romer (2004) Monetary Shock Series

For the years not covered by Romer and Romer, the value of the shock is assigned to be 0, the mean of the original data. This assignment does not bias the regression coefficient under simple conditions, for instance if the data is i.i.d. It does lower the R^2 by the fraction of years in which the assignment is done, which is 0.52.

```
> MacroeconomicData[is.na(MacroeconomicData$romerMonetaryShocks), ]$romerMonetaryShocks <- 0
```

Update Hamilton Oil Shock Series

I take the raw oil data I downloaded from FRED, and compute the maximum price over the last 4 quarters. The oil shock is the amount by which the current price exceeds this historical high water mark. I then move to a yearly frequency by summing the oil shocks over the course of the calendar year.

```
> NUMBER_OF_OBS <- dim(OilPriceData)[1]
> for (LAG in 1:4) {
      LAGGED_VARIABLE_NAME <- paste("price.lag", LAG, sep = "")
      OilPriceData[, paste("price.lag", LAG, sep = "")] <- c(rep(0, LAG), OilPriceData$price[1:(NUMBER_OF_OBS - LAG)])
  }
> OilPriceData$maxRecentPrice <- apply(OilPriceData[, c("price.lag1", "price.lag2", "price.lag3", "price.lag4")], 1, max, na.rm = TRUE)
> OilPriceData$shock <- with(OilPriceData, price - maxRecentPrice)
> OilPriceData$shock <- OilPriceData$shock * as.numeric(OilPriceData$shock > 0)
> OilPriceData <- ddply(OilPriceData, c("year"), function(X) sum(X$shock, na.rm = TRUE))
> names(OilPriceData) <- c("year", "shock")
> OilPriceData$year <- as.numeric(as.character(OilPriceData$year))
> ListOfYearsWithOilPriceData <- unique(MacroeconomicData$year[MacroeconomicData$year %in% OilPriceData$year])
> for (YEAR in ListOfYearsWithOilPriceData) {
      if (is.na(MacroeconomicData[MacroeconomicData$year == YEAR, ]$hamiltonOilShocks)) {
          MacroeconomicData[MacroeconomicData$year == YEAR, ]$hamiltonOilShocks <- OilPriceData[OilPriceData$year == YEAR, ]$shock
      }
  }
```

Add On Interest Rate Series

Next, I merge on the interest rate data used in Ang, Piasezzi and Wei (2006).

```
> MacroeconomicData <- merge(MacroeconomicData, InterestRateData, by = "year", all.x = TRUE)
```

Save Macroeconomic Data

Next, I plot the formatted and updated macroeconomic data and save it both as an `Rdata` file as well as a `csv`.

```
> MacroeconomicData <- MacroeconomicData[MacroeconomicData$year %in% seq(1950, 2008), ]
> pdf(paste(FIGURES_DIRECTORY, "TimeSeriesFacetWrap_FormattedMacroeconomicData.pdf", sep = ""), height = 5, width = 9)
> PlotData <- melt(MacroeconomicData, c("year"))
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable))
> p <- p + geom_path()
> p <- p + facet_wrap(~variable, ncol = 2, scales = "free_y")
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
> save(MacroeconomicData, file = paste(DATA_DIRECTORY, "CleanMacroeconomicData_10Oct2010.Rdata", sep = ""))
> write.csv(MacroeconomicData, file = paste(DATA_DIRECTORY, "CleanMacroeconomicData_10Oct2010.csv", sep = ""))
```

3 Clean COMPUSTAT Data

Next, I turn to the COMPUSTAT data. I convert the nominal sales to real sales using the `inflation` index housed in the `MacroeconomicData` data frame. To do this I just merge the inflation series from the cleaned macroeconomic data onto the COMPUSTAT data. The `inflation` series represents inflation index normalized to 1, so in order to convert to real sales volume I divide the `sales` series by the `inflation` series.

```
> CompustatData <- merge(CompustatData, MacroeconomicData[, c("year", "inflation")], by = "year", all.x = TRUE)
> CompustatData$sales <- with(CompustatData, sales/inflation)
> CompustatData <- CompustatData[, !(names(CompustatData) %in% c("inflation"))]
```

I filter out oil and oil-related companies (SIC code=2911, 5172, 1311, 4922, 4923, 4924 and 1389), and energy companies (SIC code between 4900 and 4940), as fluctuations of their sales come mostly from worldwide commodity prices, rather than real productivity shocks, and financial firms (SIC code between 6,000 and 7,000), because their sales do not mesh well with the spirit of the model used in the present paper.

```
> CompustatData$id <- as.numeric(as.character(CompustatData$id))
> CompustatData$year <- as.numeric(as.character(CompustatData$year))
> CompustatData$industry <- floor(CompustatData$sicCode/100)
> ListOfOilSicCodes <- c(2911, 5172, 1311, 4922, 4923, 4924, 1389)
> ListOfFinanceSicCodes <- seq(6001, 7000)
> ListOfEnergySicCodes <- seq(4901, 4940)
> CompustatData <- CompustatData[!(CompustatData$sicCode %in% ListOfOilSicCodes), ]
> CompustatData <- CompustatData[!(CompustatData$sicCode %in% ListOfEnergySicCodes), ]
> CompustatData <- CompustatData[!(CompustatData$sicCode %in% ListOfFinanceSicCodes), ]
```

I keep only firms with valid sales and number of employees data at both time t and time $t-1$. To do this, I set the COMPUSTAT data as a panel data frame using the `plm` package and compute the lagged sales volume and number of employees. I then recast the data as a standard data frame and remove any observations with missing, infinite or negative data at either time.

```
> CompustatData <- pdata.frame(CompustatData, index = c("id", "year"), drop.index = FALSE, row.names = FALSE)
> CompustatData$laggedSales <- lag(CompustatData$sales, 1)
> CompustatData$laggedNumberOfEmployees <- lag(CompustatData$numberOfEmployees, 1)
> CompustatData <- as.data.frame(CompustatData)
> CompustatData$year <- as.numeric(as.character(CompustatData$year))
> CompustatData$id <- as.numeric(as.character(CompustatData$id))
> CompustatData$companyName <- as.character(CompustatData$companyName)
> CompustatData$ticker <- as.character(CompustatData$ticker)
> CompustatData <- CompustatData[!is.na(CompustatData$sales), ]
> CompustatData <- CompustatData[!is.na(CompustatData$numberOfEmployees), ]
```

```
> CompustatData <- CompustatData[!is.na(CompustatData$laggedSales), ]
> CompustatData <- CompustatData[!is.na(CompustatData$laggedNumberOfEmployees), ]
> CompustatData <- CompustatData[is.finite(CompustatData$sales), ]
> CompustatData <- CompustatData[is.finite(CompustatData$numberOfEmployees), ]
> CompustatData <- CompustatData[is.finite(CompustatData$laggedSales), ]
> CompustatData <- CompustatData[is.finite(CompustatData$laggedNumberOfEmployees), ]
> CompustatData <- CompustatData[CompustatData$sales > 0, ]
> CompustatData <- CompustatData[CompustatData$numberOfEmployees > 0, ]
> CompustatData <- CompustatData[CompustatData$laggedSales > 0, ]
> CompustatData <- CompustatData[CompustatData$laggedNumberOfEmployees > 0, ]
```

I pick out the top 1000, 100 and 5 firms in the COMPUSTAT data in each year as sorted by lagged sales volume. To do this I first count the number of firms in each year and generate a variable that contains the rank of each firm in each year sorted by lagged sales volume. Then I create 3 dummy variables that are 1 if a firm is one of the top 1000, 100 or 5 firms respectively and 0 otherwise.

```
> NumberOfFirmsInEachYear <- ddply(CompustatData, c("year"), function(X) dim(X)[1])
> names(NumberOfFirmsInEachYear) <- c("year", "count")
> NumberOfFirmsInEachYear$year <- as.numeric(as.character(NumberOfFirmsInEachYear$year))
> CompustatData <- CompustatData[order(CompustatData$year, -CompustatData$laggedSales), ]
> CompustatData$order <- NA
> for (YEAR in seq(1952, 2008)) {
      NUMBER_OF_FIRMS_IN_YEAR <- NumberOfFirmsInEachYear[NumberOfFirmsInEachYear$year == YEAR, ]$count
      CompustatData[CompustatData$year == YEAR, ]$order <- seq(1, NUMBER_OF_FIRMS_IN_YEAR)
  }
> CompustatData$inTop1000Firms <- as.numeric(CompustatData$order <= 1000)
> CompustatData$inTop100Firms <- as.numeric(CompustatData$order <= 100)
> CompustatData$inTop5Firms <- as.numeric(CompustatData$order <= 5)
> CompustatData <- CompustatData[, !(names(CompustatData) %in% c("order"))]
```

Finally, I merge on the cleaned macroeconomic data and save the resulting data frame to file for later analysis as both a `Rdata` file as well as a `csv`.

```
> CompustatData <- merge(CompustatData, MacroeconomicData[, c("year", "realGdp", "logChangeInPerCapitaGdp")], by = "year")
> CompustatData <- CompustatData[CompustatData$year %in% seq(1952, 2008), ]
> save(CompustatData, file = paste(DATA_DIRECTORY, "CleanCompustatData_10Oct2010.Rdata", sep = ""))
> write.csv(CompustatData, file = paste(DATA_DIRECTORY, "CleanCompustatData_10Oct2010.csv", sep = ""))
```

To get an idea of how important the top firms are in terms of their share of GDP, I plot the sum of the sales of the top 50 and 100 firms in terms of sales as a fraction of real GDP.

```
> CompustatData$year <- as.numeric(as.character(CompustatData$year))
> CompustatData <- CompustatData[order(CompustatData$year, -CompustatData$sales), ]
> CompustatData$salesAsAFractionOfGDP <- with(CompustatData, sales/realGdp)
> Top100Firms <- ddply(CompustatData, .(year), function(X) sum(X[1:100, ]$salesAsAFractionOfGDP, na.rm = TRUE))
> names(Top100Firms) <- c("year", "Top 100")
> Top50Firms <- ddply(CompustatData, .(year), function(X) sum(X[1:50, ]$salesAsAFractionOfGDP, na.rm = TRUE))
> names(Top50Firms) <- c("year", "Top 50")
> PlotData <- merge(Top100Firms, Top50Firms, by = "year")
> PlotData <- melt(PlotData, id = "year")
> PlotData <- PlotData[PlotData$year >= 1975, ]
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_SalesOfTop50And100FirmsAsFractionOfRealGDP.pdf", sep = ""), height = 5, width = 9)
> p <- ggplot(PlotData, aes(x = year, y = value, group = variable, colour = variable, linetype = variable))
> p <- p + geom_path(size = 1)
> p <- p + ylim(0, 0.35)
> p <- p + ylab("") + xlab("")
> p <- p + opts(legend.title = theme_text(size = 0))
> print(p)
> dev.off()
```

I also plot the number of firms in the cleaned COMPUSTAT data over time.

```
> pdf(paste(FIGURES_DIRECTORY, "TimeSeries_NumberOfFirmsInCleanCompustatData.pdf", sep = ""), height = 5, width = 9)
> PlotData <- ddply(CompustatData, c("year"), function(X) dim(X)[1])
> names(PlotData) <- c("year", "value")
> PlotData <- PlotData[!is.na(PlotData$year), ]
> p <- ggplot(PlotData, aes(x = year, y = value))
> p <- p + geom_path()
> p <- p + opts(legend.position = "none")
> p <- p + ylab("") + xlab("")
> print(p)
> dev.off()
```

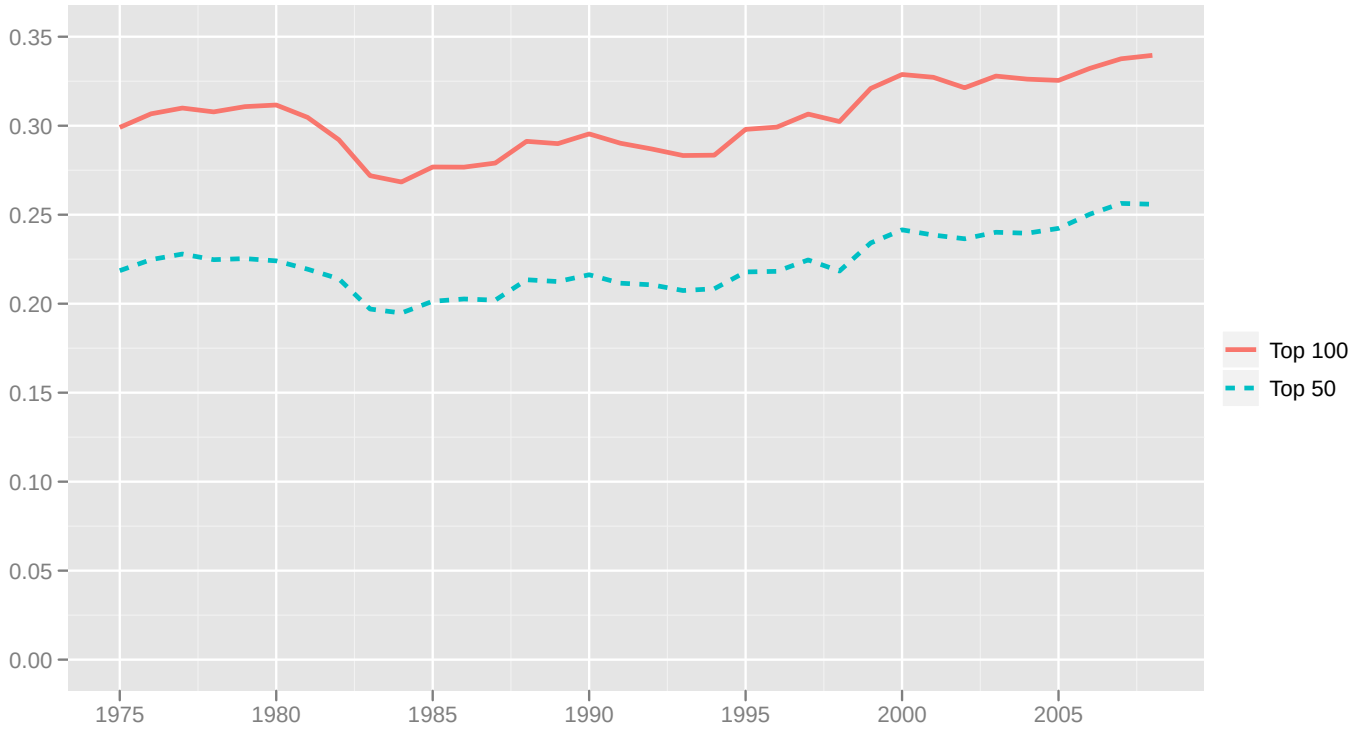


Figure 4: ***Sales of Top 50 and 100 Firms as a Fraction of Real GDP****: This figure shows the sales volume of the top* 50 *and* 100 *firms in the COMPUSTAT database as sorted by current sales as a fraction of the current real GDP. Prior to creating this plot, I remove all firms with missing sales volume or number of employees data in the current or previous year. I also remove any firms in the oil, energy or finance industries.*

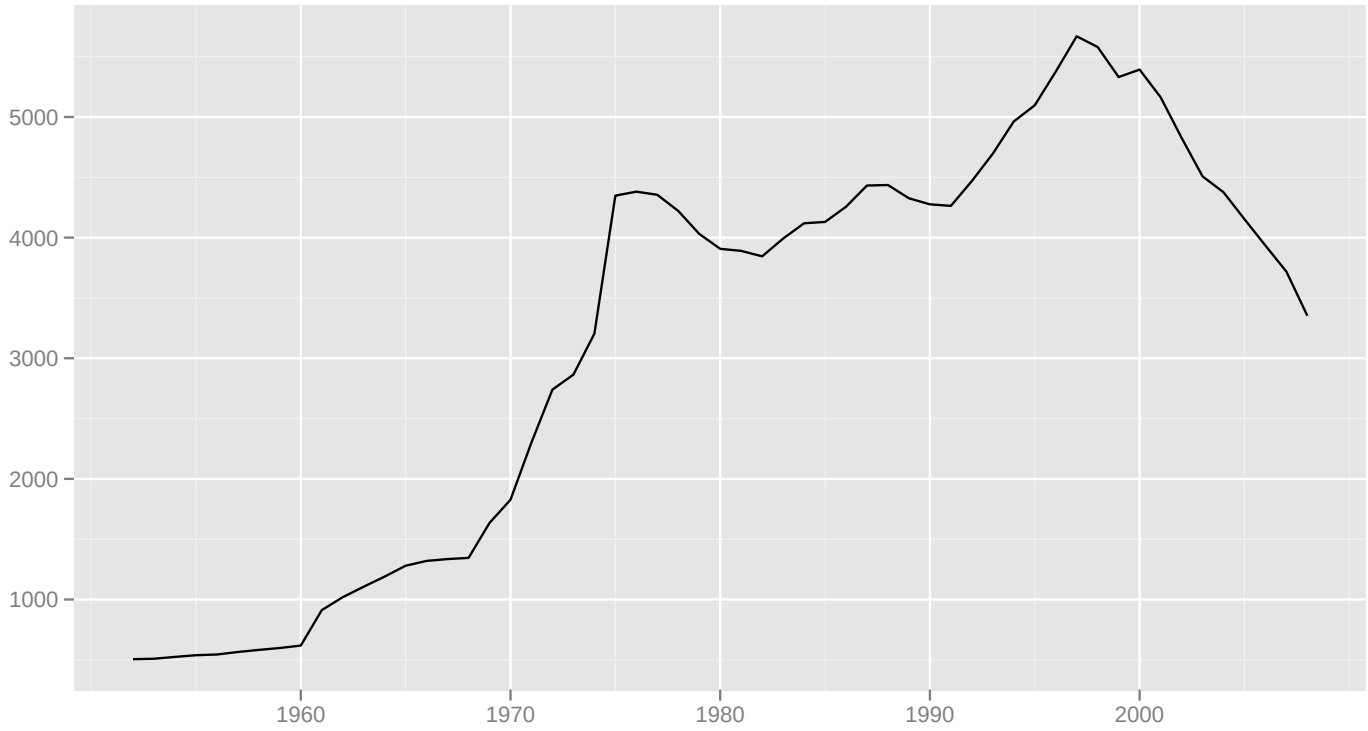


Figure 5: ***Number of Firms in the Clean COMPUSTAT Data****: This figure shows the number of firms listed in the cleaned COMPUSTAT data each year after removing firms with missing sales or number of employees data in the current or prior year and firms in the oil, energy or financial industries.*