```
> # ------------------------------------------------------------------------
> #                        Program Description
> # -------------------- .... [TRUNCATED]

> # Load in packages
> library(foreign)

> library(data.table)

> library(AER)

> library(scales)

> library(grid)

> # ========================================================================
> #         1. Firm Size Distribution in Different Datasets
> # ======= .... [TRUNCATED]

> CHNprod <- CNEC_avgp

> rm(CNEC_avgp)

> # Drop missing values
> sel <- which(CHNprod$product > 0)

> CHNprod <- CHNprod[sel]

> # CNEC records output in 2004 CNY 1000, but NGSPS records output
> #   in 2007 CNY 10,000.
> deflator <- 96.30/93.50

> CHNprod$rproduct <- (CHNprod$product/10.0)*deflator

> tmpden <- density(log(CHNprod$rproduct),
+       kernel = "gaussian", bw =0.50, na.rm = TRUE)

> pdf("./Results/FigureB1_BotRight.pdf",height=5,width=5)

> plot(tmpden,xlab="Log Output",ylab="Density",main="CNEC All",
+      xlim = c(0, 12), ylim=c(0,0.5), cex.lab = 1.25,
+      lwd = 2.0, col = 4)

> dev.off()
null device
          1

> # Read the large firms of CNEC
> load("./Data/DLARGE_R.RData")
```

```
> # Drop missing values
> sel <- which(DLARGE$product > 0)

> DLARGE <- DLARGE[sel]

> # Adjusting for inflation and the fact that CNEC records
> #     output in CNY 1000.
> deflator <- 96.30/93.50

> DLARGE$rproduct <- (DLARGE$product/10.0)*deflator

> tmpden <- density(log(DLARGE$rproduct),
+       kernel = "gaussian", bw =0.50, na.rm = TRUE)

> pdf("./Results/FigureB1_BotLeft.pdf",height=5,width=5)

> plot(tmpden,xlab="Log Output",ylab="Density",main="CNEC Large",
+      xlim = c(0, 12), ylim=c(0,0.5), cex.lab = 1.25,
+      lwd = 2.0, col = 4)

> dev.off()
null device
          1

> # Read the key firms of NGSPS
> load("./Data/KEYFIRM_R.RData")

> # Drop missing values
> sel <- which(KEYFIRM$product > 0)

> KEYFIRM <- KEYFIRM[sel]

> tmpden <- density(log(KEYFIRM$product),
+       kernel = "gaussian", bw =0.50, na.rm = TRUE)

> pdf("./Results/FigureB1_TopLeft.pdf",height=5,width=5)

> plot(tmpden,xlab="Log Output",ylab="Density",main="NGSPS Key",
+      xlim = c(0, 12), ylim=c(0,0.5), cex.lab = 1.25,
+      lwd = 2.0, col = 4)

> dev.off()
null device
          1

> # Read all firms of NGSPS
> load("./Data/ALLFIRM_R.RData")

> # Drop missing values
```

```
> sel <- which(ALLFIRM$product > 0)

> ALLFIRM <- ALLFIRM[sel]

> tmpden <- density(log(ALLFIRM$product),
+       kernel = "gaussian", bw =0.50, na.rm = TRUE)

> pdf("./Results/FigureB1_TopRight.pdf",height=5,width=5)

> plot(tmpden,xlab="Log Output",ylab="Density",main="NGSPS All",
+      xlim = c(0, 12), ylim=c(0,0.5),
+      cex.lab = 1.25, lwd = 2.0, col = 4)

> dev.off()
null device
          1

> # ============================================================================
> #           2. Industry-level Regression Results
> # ================ .... [TRUNCATED]

> load("./Data/KEYFIRM_R.RData")

> # ------------ Aggregate ownership rights type ------------------
> # 0: missing, 1: State/collective, 3: private, 4: HMT, 5: foreign
> KEYFIRM$type .... [TRUNCATED]

> # State and collective
> sel <- which(KEYFIRM$type == 110 | KEYFIRM$type == 141
+     | KEYFIRM$type == 151 | KEYFIRM$type == 120 | KEYFIRM$type ==  ....
[TRUNCATED]

> KEYFIRM$type_a[sel] <- 1

> # Private
> sel <- which(KEYFIRM$type == 170 | KEYFIRM$type == 171
+              | KEYFIRM$type == 172 | KEYFIRM$type == 173
+              | KEYFI .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 3

> # Hong Kong, Macau and Taiwan
> sel <- which(KEYFIRM$type == 200 | KEYFIRM$type == 210
+              | KEYFIRM$type == 220 | KEYFIRM$type == 230
+ .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 4

> # Foreign
> sel <- which(KEYFIRM$type == 300 | KEYFIRM$type == 310
```

```
+              | KEYFIRM$type == 320 | KEYFIRM$type == 330
+              | KEYF .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 5

> # ------------ Aggregate treatment technology type ------------------
> # For disposal equipment type
> KEYFIRM$dm1_code_a <- 0

> # Physical
> sel <- which(KEYFIRM$dm1_code >= 1000 & KEYFIRM$dm1_code < 2000)

> KEYFIRM$dm1_code_a[sel] <- 1

> # Chemical
> sel <- which(KEYFIRM$dm1_code >= 2000 & KEYFIRM$dm1_code < 3000)

> KEYFIRM$dm1_code_a[sel] <- 2

> # Physiochemical
> sel <- which(KEYFIRM$dm1_code >= 3000 & KEYFIRM$dm1_code < 4000)

> KEYFIRM$dm1_code_a[sel] <- 3

> # Biological
> sel <- which(KEYFIRM$dm1_code >= 4000 & KEYFIRM$dm1_code < 5000)

> KEYFIRM$dm1_code_a[sel] <- 4

> # Combination
> sel <- which(KEYFIRM$dm1_code >= 5000 & KEYFIRM$dm1_code < 6000)

> KEYFIRM$dm1_code_a[sel] <- 5

> # --------------- Declare dummies -----------------
> KEYFIRM$province <- factor(KEYFIRM$province)        # Province

> KEYFIRM$industry_a <- factor(KEYFIRM$industry_a)    # 2-digit industry

> KEYFIRM$Census_Type <- factor(KEYFIRM$Census_Type)  # key 1, regular 2

> KEYFIRM$dm1_code_a <- factor(KEYFIRM$dm1_code_a)    # treatment

> KEYFIRM$type_a <- factor(KEYFIRM$type_a)            # ownership rights

> KEYFIRM <- KEYFIRM[product > 0 & cod_e > 0]

> KEYFIRM$intensity <- with(KEYFIRM, intensity <- cod_e/product)

> # ====================== Table C1 ========================================
```

```
> # -------------------- PAPER ------------------------
> sel <- which .... [TRUNCATED]

> PAPER <- KEYFIRM[sel]

> # All firms with 4-digit
> lm_paper_all <-
+   lm(log(cod_e) ~ log(product) + province + type_a, data = PAPER)

> summary(lm_paper_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + type_a, data = PAPER)

Residuals:
     Min       1Q   Median       3Q      Max
-13.4466  -0.8481   0.1998   1.1580   6.3323

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -4.14142    0.55548  -7.456 1.03e-13 ***
log(product)  0.74414    0.01517  49.062  < 2e-16 ***
province12    1.21524    0.63119   1.925  0.05424 .
province13    2.82398    0.55137   5.122 3.13e-07 ***
province14    2.65629    0.58340   4.553 5.40e-06 ***
province15    3.82537    0.71138   5.377 7.86e-08 ***
province21    2.26444    0.56535   4.005 6.27e-05 ***
province22    2.40185    0.59508   4.036 5.50e-05 ***
province23    1.86424    0.58015   3.213  0.00132 **
province31   -0.11760    0.61249  -0.192  0.84775
province32    1.22156    0.56033   2.180  0.02929 *
province33    2.29858    0.54959   4.182 2.93e-05 ***
province34    2.31478    0.56514   4.096 4.26e-05 ***
province35    2.35863    0.55360   4.260 2.07e-05 ***
province36    3.29481    0.56874   5.793 7.28e-09 ***
province37    1.72475    0.55666   3.098  0.00196 **
province41    2.37019    0.55785   4.249 2.18e-05 ***
province42    2.56179    0.58837   4.354 1.36e-05 ***
province43    3.15141    0.55324   5.696 1.29e-08 ***
province44    1.29017    0.54983   2.346  0.01899 *
province45    2.74734    0.55570   4.944 7.88e-07 ***
province46    1.72483    0.86107   2.003  0.04521 *
province50    2.50623    0.56946   4.401 1.10e-05 ***
province51    2.39404    0.55440   4.318 1.60e-05 ***
province52    2.76975    0.62592   4.425 9.82e-06 ***
province53    2.24780    0.57608   3.902 9.66e-05 ***
province61    3.67029    0.58773   6.245 4.55e-10 ***
province62    3.70177    0.68325   5.418 6.28e-08 ***
province63    2.57425    1.44091   1.787  0.07406 .
province64    4.32447    0.68382   6.324 2.75e-10 ***
```

```
province65    3.45967    0.58782   5.886 4.20e-09 ***
type_a3      -0.11302    0.05887  -1.920  0.05492 .
type_a4      -1.85073    0.14713 -12.579  < 2e-16 ***
type_a5      -1.33887    0.14644  -9.143  < 2e-16 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 1.886 on 5598 degrees of freedom
Multiple R-squared:  0.3802,    Adjusted R-squared:  0.3765
F-statistic: 104.1 on 33 and 5598 DF,  p-value: < 2.2e-16


> # -------------------- FOOD ------------------------
> sel <- which(KEYFIRM$industry_a == 13)

> AGRI <- KEYFIRM[sel]

> lm_agri_all <-
+   lm(log(cod_e) ~ log(product) + province + type_a, data = AGRI)

> summary(lm_agri_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + type_a, data = AGRI)

Residuals:
     Min       1Q   Median       3Q      Max
-10.6780  -1.1722  -0.0038   1.2367   6.2153

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.15420    0.26204  -8.221 2.39e-16 ***
log(product)  0.39208    0.01025  38.233  < 2e-16 ***
province12    0.20345    0.39120   0.520 0.603034
province13    2.44638    0.27483   8.901  < 2e-16 ***
province14    2.23135    0.35611   6.266 3.93e-10 ***
province15    2.44974    0.29869   8.202 2.81e-16 ***
province21    0.58845    0.27222   2.162 0.030675 *
province22    0.93785    0.32037   2.927 0.003429 **
province23    1.09267    0.31169   3.506 0.000458 ***
province31   -0.12385    0.31497  -0.393 0.694162
province32    0.18790    0.26677   0.704 0.481239
province33    0.35068    0.26885   1.304 0.192141
province34    0.16107    0.30069   0.536 0.592203
province35    0.80501    0.27282   2.951 0.003181 **
province36    0.77931    0.31418   2.480 0.013147 *
province37    0.92735    0.25777   3.598 0.000323 ***
province41    1.07815    0.30692   3.513 0.000446 ***
province42    0.82670    0.30837   2.681 0.007362 **
province43    0.51904    0.29701   1.748 0.080583 .
```

```
province44    0.79431    0.25786   3.080 0.002075 **
province45    2.96632    0.26566  11.166  < 2e-16 ***
province46    1.85295    0.31783   5.830 5.80e-09 ***
province50    0.56771    0.29084   1.952 0.050984 .
province51    0.59426    0.26194   2.269 0.023320 *
province52    1.43044    0.31898   4.484 7.43e-06 ***
province53    2.41292    0.27803   8.679  < 2e-16 ***
province61    0.90886    0.33033   2.751 0.005951 **
province62    3.12809    0.30132  10.381  < 2e-16 ***
province63    1.82011    0.52782   3.448 0.000567 ***
province64    3.93374    0.34815  11.299  < 2e-16 ***
province65    1.91957    0.30343   6.326 2.67e-10 ***
type_a3      -0.11772    0.05363  -2.195 0.028195 *
type_a4      -0.25071    0.13485  -1.859 0.063044 .
type_a5       0.10846    0.09485   1.143 0.252878
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 1.972 on 6859 degrees of freedom
Multiple R-squared:  0.3065,    Adjusted R-squared:  0.3031
F-statistic: 91.84 on 33 and 6859 DF,  p-value: < 2.2e-16


> # -------------------- TEXTILE ------------------------
> sel <- which(KEYFIRM$industry_a == 17)

> TEXT <- KEYFIRM[sel]

> lm_text_all <-
+   lm(log(cod_e) ~ log(product) + province + type_a, data = TEXT)

> summary(lm_text_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + type_a, data = TEXT)

Residuals:
     Min       1Q   Median       3Q      Max
-10.0758  -1.1490   0.1593   1.2852   5.6690

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.710751   0.572259  -6.484 9.59e-11 ***
log(product)  0.697719   0.014140  49.343  < 2e-16 ***
province12    0.625765   0.619972   1.009 0.312849
province13    2.074432   0.575077   3.607 0.000312 ***
province14    0.806669   0.720367   1.120 0.262841
province15    1.048037   0.687700   1.524 0.127566
province21    1.334653   0.583080   2.289 0.022114 *
province22   -0.002802   0.720638  -0.004 0.996898
```

```
province23   -0.323317   0.739058  -0.437 0.661785
province31    0.559988   0.575393   0.973 0.330478
province32    1.055124   0.564520   1.869 0.061661 .
province33    1.623334   0.563909   2.879 0.004006 **
province34    0.864636   0.606064   1.427 0.153733
province35    1.171078   0.580165   2.019 0.043579 *
province36    1.426755   0.616306   2.315 0.020644 *
province37    0.737737   0.568320   1.298 0.194301
province41    1.187712   0.591873   2.007 0.044825 *
province42    1.688945   0.617022   2.737 0.006213 **
province43    1.426143   0.637819   2.236 0.025389 *
province44    0.442193   0.564059   0.784 0.433101
province45    0.656195   0.615952   1.065 0.286766
province46   -4.584671   1.431053  -3.204 0.001364 **
province50    0.684822   0.614695   1.114 0.265286
province51    0.764338   0.583424   1.310 0.190214
province52   -1.311594   1.944471  -0.675 0.500003
province53    0.574173   0.656626   0.874 0.381918
province61    1.000119   0.653652   1.530 0.126055
province62    3.461537   0.678805   5.099 3.51e-07 ***
province63    2.797529   1.212180   2.308 0.021040 *
province64    0.561918   0.814713   0.690 0.490401
province65    0.744799   0.682634   1.091 0.275286
type_a3      -0.027404   0.057099  -0.480 0.631284
type_a4      -0.131236   0.080858  -1.623 0.104634
type_a5      -0.358119   0.093056  -3.848 0.000120 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 1.861 on 6250 degrees of freedom
Multiple R-squared:  0.3802,    Adjusted R-squared:  0.3769
F-statistic: 116.2 on 33 and 6250 DF,  p-value: < 2.2e-16


> # -------------------- CHEMISTRY -------------------------
> sel <- which(KEYFIRM$industry_a == 26)

> CHEM <- KEYFIRM[sel]

> lm_chem_all <-
+   lm(log(cod_e) ~ log(product) + province + type_a, data = CHEM)

> summary(lm_chem_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + type_a, data = CHEM)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5407  -1.6089   0.0723   1.6780   7.3647
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.11964    0.32907 -21.636  < 2e-16 ***
log(product)  0.88204    0.01406  62.719  < 2e-16 ***
province12    1.04072    0.36052   2.887 0.003903 **
province13    2.43543    0.32581   7.475 8.47e-14 ***
province14    1.94797    0.37206   5.236 1.68e-07 ***
province15    1.83035    0.45758   4.000 6.39e-05 ***
province21    1.03319    0.35164   2.938 0.003310 **
province22    1.00895    0.42653   2.366 0.018027 *
province23    1.66496    0.47192   3.528 0.000421 ***
province31   -0.14713    0.32889  -0.447 0.654632
province32    0.58365    0.31202   1.871 0.061445 .
province33    0.86281    0.31584   2.732 0.006312 **
province34    0.95280    0.34031   2.800 0.005125 **
province35    0.79785    0.34319   2.325 0.020104 *
province36    0.78462    0.34590   2.268 0.023333 *
province37    1.11010    0.32259   3.441 0.000582 ***
province41    1.35435    0.33748   4.013 6.04e-05 ***
province42    1.37661    0.33825   4.070 4.75e-05 ***
province43    1.65401    0.33611   4.921 8.77e-07 ***
province44   -0.09812    0.31611  -0.310 0.756277
province45    1.50704    0.35492   4.246 2.20e-05 ***
province46    0.76588    0.39202   1.954 0.050772 .
province50    0.14297    0.35106   0.407 0.683825
province51    0.83383    0.33272   2.506 0.012226 *
province52    0.83275    0.47209   1.764 0.077776 .
province53    1.43894    0.34500   4.171 3.06e-05 ***
province61    1.40996    0.38950   3.620 0.000296 ***
province62    1.42650    0.46008   3.101 0.001938 **
province63    2.34561    0.74504   3.148 0.001648 **
province64    1.89154    0.55394   3.415 0.000641 ***
province65    1.18389    0.45499   2.602 0.009283 **
type_a3      -0.10154    0.05859  -1.733 0.083117 .
type_a4      -0.27850    0.11492  -2.423 0.015397 *
type_a5      -0.42889    0.09032  -4.748 2.08e-06 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 2.352 on 8652 degrees of freedom
Multiple R-squared:  0.3963,    Adjusted R-squared:  0.394
F-statistic: 172.1 on 33 and 8652 DF,  p-value: < 2.2e-16


> # -------------------- BEVERAGE -------------------------
> sel <- which(KEYFIRM$industry_a == 15)

> BEVER <- KEYFIRM[sel]
```

```
> lm_bever_all <-
+   lm(log(cod_e) ~ log(product) + province + type_a, data = BEVER)

> summary(lm_bever_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + type_a, data = BEVER)

Residuals:
    Min      1Q  Median      3Q     Max
-7.7225 -1.0077  0.0767  0.9902  5.3306

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.85605    0.38784  -7.364 2.91e-13 ***
log(product)  0.58756    0.02487  23.621  < 2e-16 ***
province12    1.01958    0.51311   1.987 0.047096 *
province13    1.02969    0.36898   2.791 0.005326 **
province14    0.84691    0.40227   2.105 0.035424 *
province15    1.52263    0.47108   3.232 0.001255 **
province21    1.38780    0.38784   3.578 0.000357 ***
province22    1.35650    0.43184   3.141 0.001715 **
province23    1.61536    0.41118   3.929 8.93e-05 ***
province31   -0.24380    0.42663  -0.571 0.567780
province32    1.67844    0.35216   4.766 2.06e-06 ***
province33    0.88932    0.34591   2.571 0.010235 *
province34    1.43793    0.37681   3.816 0.000141 ***
province35    0.86837    0.38615   2.249 0.024667 *
province36    2.09279    0.43225   4.842 1.42e-06 ***
province37    1.09378    0.34223   3.196 0.001422 **
province41    1.55802    0.36168   4.308 1.75e-05 ***
province42    1.13166    0.41790   2.708 0.006845 **
province43    1.06876    0.44224   2.417 0.015779 *
province44    0.67513    0.35623   1.895 0.058256 .
province45    2.33124    0.38863   5.999 2.48e-09 ***
province46   -0.82495    0.66796  -1.235 0.217010
province50    1.43783    0.45931   3.130 0.001779 **
province51    1.11879    0.35888   3.117 0.001858 **
province52    1.34595    0.44322   3.037 0.002432 **
province53    0.83989    0.43269   1.941 0.052433 .
province54   -0.67731    1.01408  -0.668 0.504298
province61    1.76609    0.38569   4.579 5.05e-06 ***
province62    0.97553    0.46972   2.077 0.037982 *
province63    0.39817    1.01326   0.393 0.694405
province64    2.70473    0.63981   4.227 2.50e-05 ***
province65    1.40095    0.43164   3.246 0.001197 **
type_a3      -0.18566    0.10859  -1.710 0.087513 .
type_a4      -1.01176    0.18610  -5.437 6.32e-08 ***
type_a5      -0.56233    0.12087  -4.652 3.57e-06 ***
---
```

```
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 1.666 on 1523 degrees of freedom
Multiple R-squared:  0.3379,    Adjusted R-squared:  0.3231
F-statistic: 22.86 on 34 and 1523 DF,  p-value: < 2.2e-16


> # ====================== Figure C1 ========================================
> # -------------------- PAPER ------------------------
> lm_paper_aux .... [TRUNCATED]

> lm_paper_aux2 <- lm(log(product) ~
+     province + type_a, data = PAPER)

> PAPER$res_intensity <- residuals(lm_paper_aux1)

> PAPER$res_product <- residuals(lm_paper_aux2)

> pdf("./Results/FigureC1_TopLeft.pdf",height=5,width=5)

> plot(PAPER$res_intensity~PAPER$res_product,cex=0.5,
+      mgp=c(1.75, 0.75, 0),
+      xlab="Log Production",ylab="Log Intensity",main="Paper",
+   .... [TRUNCATED]

> paper_residual <- lm(res_intensity ~ res_product, data = PAPER)

> abline(paper_residual,col="red",lwd=4)

> dev.off()
null device
          1

> # -------------------- FOOD ------------------------
> lm_agri_aux1 <- lm(log(intensity) ~
+       province + type_a, data = AGRI)

> lm_agri_aux2 <- lm(log(product) ~
+       province + type_a, data = AGRI)

> AGRI$res_intensity <- residuals(lm_agri_aux1)

> AGRI$res_product <- residuals(lm_agri_aux2)

> pdf("./Results/FigureC1_TopRight.pdf",height=5,width=5)

> plot(AGRI$res_intensity~AGRI$res_product,
+      cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Log Production",ylab="Log Intensity",
+      main="Agric ..." ... [TRUNCATED]
```

```
> agri_residual <- lm(res_intensity ~ res_product, data = AGRI)

> abline(agri_residual,col="red",lwd=4)

> dev.off()
null device
          1

> # -------------------- TEXTILE ------------------------
> lm_text_aux1 <- lm(log(intensity) ~
+       province + type_a, data = TEXT)

> lm_text_aux2 <- lm(log(product) ~
+       province + type_a, data = TEXT)

> TEXT$res_intensity <- residuals(lm_text_aux1)

> TEXT$res_product <- residuals(lm_text_aux2)

> pdf("./Results/FigureC1_MidLeft.pdf",height=5,width=5)

> plot(TEXT$res_intensity~TEXT$res_product,
+      cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Log Production",ylab="Log Intensity",
+      main="Texti ..." ... [TRUNCATED]

> text_residual <- lm(res_intensity ~ res_product, data = TEXT)

> abline(text_residual,col="red",lwd=4)

> dev.off()
null device
          1

> # -------------------- CHEMISTRY ------------------------
> lm_chem_aux1 <- lm(log(intensity) ~
+       province + type_a, data = CHEM)

> lm_chem_aux2 <- lm(log(product) ~
+       province + type_a, data = CHEM)

> CHEM$res_intensity <- residuals(lm_chem_aux1)

> CHEM$res_product <- residuals(lm_chem_aux2)

> pdf("./Results/FigureC1_MidRight.pdf",height=5,width=5)

> plot(CHEM$res_intensity~CHEM$res_product,
+      cex=0.5,mgp=c(1.75, 0.75, 0),
```

```
+      xlab="Log Production",ylab="Log Intensity",
+      main="Chemi ..." ... [TRUNCATED]

> chem_residual <- lm(res_intensity ~ res_product, data = CHEM)

> abline(chem_residual,col="red",lwd=4)

> dev.off()
null device
          1

> # -------------------- BEVERAGE ------------------------
> lm_bever_aux1 <- lm(log(intensity) ~
+       province + type_a, data = BEVER)

> lm_bever_aux2 <- lm(log(product) ~
+       province + type_a, data = BEVER)

> BEVER$res_intensity <- residuals(lm_bever_aux1)

> BEVER$res_product <- residuals(lm_bever_aux2)

> pdf("./Results/FigureC1_BotLeft.pdf",height=5,width=5)

> plot(BEVER$res_intensity~BEVER$res_product,
+      cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Log Production",ylab="Log Intensity",
+      main="Bev ..." ... [TRUNCATED]

> bever_residual <- lm(res_intensity ~ res_product, data = BEVER)

> abline(bever_residual,col="red",lwd=4)

> dev.off()
null device
          1

> # ---------------------- All Manufacturing ----------------------
> rm(list = ls())

> load("./Data/ALLFIRM_R.RData")

> # ------------ Aggregate ownership rights type ------------------
> # 0: missing, 1: state/collective, 3: private, 4: HMT, 5: foreign
> ALLFIRM$type .... [TRUNCATED]

> # State and collective
> sel <- which(ALLFIRM$type == 110 | ALLFIRM$type == 141
+     | ALLFIRM$type == 151 | ALLFIRM$type == 120 | ALLFIRM$type ==  ....
[TRUNCATED]
```

```
> ALLFIRM$type_a[sel] <- 1

> # Private
> sel <- which(ALLFIRM$type == 170 | ALLFIRM$type == 171
+     | ALLFIRM$type == 172 | ALLFIRM$type == 173
+     | ALLFIRM$type == 174 | A .... [TRUNCATED]

> ALLFIRM$type_a[sel] <- 3

> # Hong Kong, Macau and Taiwan
> sel <- which(ALLFIRM$type == 200 | ALLFIRM$type == 210
+     | ALLFIRM$type == 220 | ALLFIRM$type == 230
+     | AL .... [TRUNCATED]

> ALLFIRM$type_a[sel] <- 4

> # Foreign
> sel <- which(ALLFIRM$type == 300 | ALLFIRM$type == 310
+     | ALLFIRM$type == 320 | ALLFIRM$type == 330
+     | ALLFIRM$type == 340)

> ALLFIRM$type_a[sel] <- 5

> # ------------ Aggregate treatment technology type ------------------
> # For disposal equipment type
> ALLFIRM$dm1_code_a <- 0

> # Physical
> sel <- which(ALLFIRM$dm1_code >= 1000 & ALLFIRM$dm1_code < 2000)

> ALLFIRM$dm1_code_a[sel] <- 1

> # Chemical
> sel <- which(ALLFIRM$dm1_code >= 2000 & ALLFIRM$dm1_code < 3000)

> ALLFIRM$dm1_code_a[sel] <- 2

> # Physiochemical
> sel <- which(ALLFIRM$dm1_code >= 3000 & ALLFIRM$dm1_code < 4000)

> ALLFIRM$dm1_code_a[sel] <- 3

> # Biological
> sel <- which(ALLFIRM$dm1_code >= 4000 & ALLFIRM$dm1_code < 5000)

> ALLFIRM$dm1_code_a[sel] <- 4

> # Combination
> sel <- which(ALLFIRM$dm1_code >= 5000 & ALLFIRM$dm1_code < 6000)
```

```
> ALLFIRM$dm1_code_a[sel] <- 5

> # --------------- Declare dummies -----------------
> ALLFIRM$province <- factor(ALLFIRM$province)

> ALLFIRM$industry_a <- factor(ALLFIRM$industry_a)

> ALLFIRM$Census_Type <- factor(ALLFIRM$Census_Type)

> ALLFIRM$dm1_code_a <- factor(ALLFIRM$dm1_code_a)

> ALLFIRM$type_a <- factor(ALLFIRM$type_a)

> POLLUTEALL <- ALLFIRM

> qup <- quantile(POLLUTEALL$product, probs=c(.99),na.rm= TRUE)

> qdown <- quantile(POLLUTEALL$product, probs=c(.01),na.rm= TRUE)

> sel <- which(POLLUTEALL$product > 0 & POLLUTEALL$product < qup
+              & POLLUTEALL$product > qdown)

> POLLUTEALL <- POLLUTEALL[sel]

> # ================== Regression (!#) Page 7 ================================
> CODR <- POLLUTEALL[cod_e > 0]

> lm_all <- lm(log(cod_e)~log(product)+industry_a+province+type_a,
+              data=CODR)

> summary(lm_all)

Call:
lm(formula = log(cod_e) ~ log(product) + industry_a + province +
    type_a, data = CODR)

Residuals:
     Min       1Q   Median       3Q      Max
-14.5211  -1.2736   0.0248   1.3930  10.0484

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -11.040088   0.840624 -13.133  < 2e-16 ***
log(product)   0.631538   0.002287 276.102  < 2e-16 ***
industry_a6    4.066557   0.183563  22.153  < 2e-16 ***
industry_a7    2.507671   0.260047   9.643  < 2e-16 ***
industry_a8    2.908467   0.184936  15.727  < 2e-16 ***
industry_a9    4.359134   0.186468  23.377  < 2e-16 ***
industry_a10   3.536176   0.187042  18.906  < 2e-16 ***
```

industry_a11	3.173931	0.745297	4.259	2.06e-05	***
industry_a13	4.507878	0.182680	24.676	< 2e-16	***
industry_a14	4.248949	0.185540	22.900	< 2e-16	***
industry_a15	5.942006	0.182934	32.482	< 2e-16	***
industry_a16	3.034569	0.342139	8.869	< 2e-16	***
industry_a17	3.942943	0.182528	21.602	< 2e-16	***
industry_a18	1.815071	0.183074	9.914	< 2e-16	***
industry_a19	2.738482	0.184197	14.867	< 2e-16	***
industry_a20	2.049189	0.183212	11.185	< 2e-16	***
industry_a21	1.774689	0.185544	9.565	< 2e-16	***
industry_a22	5.227787	0.183419	28.502	< 2e-16	***
industry_a23	2.068146	0.184512	11.209	< 2e-16	***
industry_a24	2.069918	0.189068	10.948	< 2e-16	***
industry_a25	3.424940	0.206530	16.583	< 2e-16	***
industry_a26	3.121090	0.183273	17.030	< 2e-16	***
industry_a27	4.035073	0.185248	21.782	< 2e-16	***
industry_a28	4.188194	0.202655	20.667	< 2e-16	***
industry_a29	2.464293	0.184609	13.349	< 2e-16	***
industry_a30	2.162300	0.182830	11.827	< 2e-16	***
industry_a31	1.905929	0.182809	10.426	< 2e-16	***
industry_a32	2.353995	0.189046	12.452	< 2e-16	***
industry_a33	1.897682	0.190049	9.985	< 2e-16	***
industry_a34	2.392907	0.182509	13.111	< 2e-16	***
industry_a35	2.217059	0.182396	12.155	< 2e-16	***
industry_a36	1.879484	0.183142	10.262	< 2e-16	***
industry_a37	2.457572	0.182901	13.437	< 2e-16	***
industry_a39	1.768579	0.183362	9.645	< 2e-16	***
industry_a40	2.222179	0.184599	12.038	< 2e-16	***
industry_a41	2.032183	0.187407	10.844	< 2e-16	***
industry_a42	2.412406	0.184007	13.110	< 2e-16	***
industry_a43	3.006609	0.183522	16.383	< 2e-16	***
industry_a44	3.329125	0.191300	17.403	< 2e-16	***
industry_a45	1.992710	0.271246	7.347	2.04e-13	***
industry_a46	4.963907	0.184613	26.888	< 2e-16	***
province12	3.099888	0.058852	52.672	< 2e-16	***
province13	3.447356	0.045980	74.975	< 2e-16	***
province14	2.954855	0.060233	49.057	< 2e-16	***
province15	3.578830	0.071843	49.814	< 2e-16	***
province21	3.030736	0.045923	65.996	< 2e-16	***
province22	2.462764	0.058758	41.914	< 2e-16	***
province23	2.880822	0.057006	50.536	< 2e-16	***
province31	2.866587	0.053669	53.412	< 2e-16	***
province32	2.014213	0.041076	49.036	< 2e-16	***
province33	2.989034	0.041025	72.859	< 2e-16	***
province34	3.074268	0.047368	64.901	< 2e-16	***
province35	3.503041	0.044886	78.043	< 2e-16	***
province36	3.270371	0.052383	62.433	< 2e-16	***
province37	3.069283	0.043928	69.870	< 2e-16	***
province41	3.122388	0.046151	67.655	< 2e-16	***
province42	3.103378	0.046084	67.342	< 2e-16	***

```
province43     3.214025   0.045758  70.240  < 2e-16 ***
province44     3.173120   0.041620  76.241  < 2e-16 ***
province45     3.831510   0.052353  73.186  < 2e-16 ***
province46     3.503058   0.117570  29.796  < 2e-16 ***
province50     2.979454   0.044907  66.347  < 2e-16 ***
province51     3.426336   0.045236  75.743  < 2e-16 ***
province52     3.288490   0.062069  52.981  < 2e-16 ***
province53     2.007196   0.047815  41.978  < 2e-16 ***
province54     2.928970   0.297823   9.835  < 2e-16 ***
province61     3.099955   0.054788  56.581  < 2e-16 ***
province62     3.696834   0.074047  49.925  < 2e-16 ***
province63     3.272124   0.182763  17.904  < 2e-16 ***
province64     3.558061   0.099900  35.616  < 2e-16 ***
province65     2.661974   0.053246  49.994  < 2e-16 ***
type_a1        0.082039   0.819638   0.100    0.920
type_a3       -0.084779   0.819599  -0.103    0.918
type_a4        0.262972   0.819965   0.321    0.748
type_a5        0.111698   0.819884   0.136    0.892
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 2.168 on 299580 degrees of freedom
Multiple R-squared:  0.4318,    Adjusted R-squared:  0.4317
F-statistic:  3077 on 74 and 299580 DF,  p-value: < 2.2e-16


> # ================== Figure C2 ================================================
> CODR$intensity <- with(CODR, intensity <- cod_e/product)

> lm_codr_aux1 <- lm(log(intensity) ~
+                      province + type_a + industry_a, data = CODR)

> lm_codr_aux2 <- lm(log(product) ~
+                      province + type_a + industry_a, data = CODR)

> CODR$res_intensity <- residuals(lm_codr_aux1)

> CODR$res_product <- residuals(lm_codr_aux2)

> pdf("./Results/FigureC2.pdf",height=5,width=5)

> plot((CODR$res_intensity)~(CODR$res_product),
+      cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Log Production",ylab="Log Intensity Residual",
+     .... [TRUNCATED]

> codr_residual <- lm(res_intensity ~ res_product, data = CODR)

> abline(codr_residual,col="red",lwd=4)
```

```
> dev.off()
null device
          1

> # ================= Regression (!#) Page 9 ================================
> CODR$clean <- 0

> sel <- which(CODR$dm1_code_a == 4 | CODR$dm1_code_a == 5)

> CODR$clean[sel] <- 1

> lm_clean_all <- lm(clean ~ log(product) + industry_a
+                     + province + type_a, data = CODR)

> summary(lm_clean_all)

Call:
lm(formula = clean ~ log(product) + industry_a + province + type_a,
    data = CODR)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93972 -0.13862 -0.04494  0.03174  1.19016

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1635518  0.1053469  -1.553 0.120542
log(product)  0.0371550  0.0002866 129.618  < 2e-16 ***
industry_a6   0.0512322  0.0230041   2.227 0.025942 *
industry_a7   0.2248357  0.0325891   6.899 5.24e-12 ***
industry_a8  -0.0031873  0.0231762  -0.138 0.890616
industry_a9   0.0315518  0.0233682   1.350 0.176953
industry_a10  0.0900005  0.0234400   3.840 0.000123 ***
industry_a11 -0.0138562  0.0934006  -0.148 0.882065
industry_a13  0.2659780  0.0228934  11.618  < 2e-16 ***
industry_a14  0.4391282  0.0232518  18.886  < 2e-16 ***
industry_a15  0.1858051  0.0229253   8.105 5.30e-16 ***
industry_a16  0.1690945  0.0428768   3.944 8.02e-05 ***
industry_a17  0.2044864  0.0228743   8.940  < 2e-16 ***
industry_a18  0.0792209  0.0229428   3.453 0.000555 ***
industry_a19  0.1815586  0.0230835   7.865 3.69e-15 ***
industry_a20  0.0759852  0.0229602   3.309 0.000935 ***
industry_a21  0.0832748  0.0232524   3.581 0.000342 ***
industry_a22  0.2312614  0.0229860  10.061  < 2e-16 ***
industry_a23  0.0625819  0.0231230   2.706 0.006801 **
industry_a24  0.0571279  0.0236941   2.411 0.015907 *
industry_a25  0.2927208  0.0258824  11.310  < 2e-16 ***
industry_a26  0.3511544  0.0229678  15.289  < 2e-16 ***
industry_a27  0.4828530  0.0232152  20.799  < 2e-16 ***
industry_a28  0.2306370  0.0253967   9.081  < 2e-16 ***
```

```
industry_a29  0.0343561  0.0231352   1.485 0.137541
industry_a30  0.0558634  0.0229123   2.438 0.014764 *
industry_a31  0.0467583  0.0229097   2.041 0.041253 *
industry_a32  0.0681663  0.0236912   2.877 0.004011 **
industry_a33  0.0999591  0.0238169   4.197 2.71e-05 ***
industry_a34  0.1536906  0.0228720   6.720 1.83e-11 ***
industry_a35  0.0652444  0.0228579   2.854 0.004313 **
industry_a36  0.0676147  0.0229513   2.946 0.003219 **
industry_a37  0.0826613  0.0229211   3.606 0.000311 ***
industry_a39  0.0356095  0.0229789   1.550 0.121225
industry_a40  0.0629008  0.0231340   2.719 0.006549 **
industry_a41  0.0603604  0.0234858   2.570 0.010168 *
industry_a42  0.1048585  0.0230598   4.547 5.44e-06 ***
industry_a43  0.1310218  0.0229989   5.697 1.22e-08 ***
industry_a44  0.0635314  0.0239737   2.650 0.008049 **
industry_a45  0.1003428  0.0339925   2.952 0.003158 **
industry_a46  0.0690484  0.0231357   2.984 0.002841 **
province12    0.0243161  0.0073754   3.297 0.000978 ***
province13    0.1205531  0.0057622  20.921  < 2e-16 ***
province14    0.0258203  0.0075484   3.421 0.000625 ***
province15    0.0234175  0.0090034   2.601 0.009297 **
province21    0.0200135  0.0057551   3.478 0.000506 ***
province22    0.0150344  0.0073635   2.042 0.041177 *
province23    0.0232220  0.0071439   3.251 0.001152 **
province31    0.0859411  0.0067258  12.778  < 2e-16 ***
province32   -0.0022235  0.0051476  -0.432 0.665780
province33    0.0320675  0.0051412   6.237 4.46e-10 ***
province34    0.0133557  0.0059362   2.250 0.024458 *
province35    0.0394095  0.0056251   7.006 2.46e-12 ***
province36    0.0102126  0.0065646   1.556 0.119776
province37    0.0609264  0.0055051  11.067  < 2e-16 ***
province41    0.0028897  0.0057837   0.500 0.617341
province42   -0.0078912  0.0057753  -1.366 0.171821
province43   -0.0426821  0.0057344  -7.443 9.85e-14 ***
province44    0.0737723  0.0052158  14.144  < 2e-16 ***
province45    0.0157849  0.0065608   2.406 0.016132 *
province46    0.2858789  0.0147338  19.403  < 2e-16 ***
province50    0.0645592  0.0056278  11.472  < 2e-16 ***
province51    0.0533430  0.0056690   9.410  < 2e-16 ***
province52    0.0404404  0.0077785   5.199 2.00e-07 ***
province53    0.0050479  0.0059922   0.842 0.399557
province54   -0.0226461  0.0373232  -0.607 0.544014
province61    0.0132270  0.0068660   1.926 0.054050 .
province62   -0.0070234  0.0092796  -0.757 0.449133
province63   -0.0190256  0.0229039  -0.831 0.406161
province64   -0.0358896  0.0125195  -2.867 0.004148 **
province65   -0.0151226  0.0066728  -2.266 0.023433 *
type_a1      -0.0667727  0.1027170  -0.650 0.515651
type_a3      -0.0932965  0.1027121  -0.908 0.363705
type_a4       0.0038633  0.1027580   0.038 0.970009
```

```
type_a5      -0.0250361  0.1027478  -0.244 0.807490
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 0.2717 on 299580 degrees of freedom
Multiple R-squared:  0.2206,    Adjusted R-squared:  0.2204
F-statistic:  1146 on 74 and 299580 DF,  p-value: < 2.2e-16


> # ----------------- Cluster Standard Errors ---------------------------
> # Cluster the standard error at provincial level
> # Inline Function for  .... [TRUNCATED]

> cl <- function(dat,fm, cluster){
+   require(sandwich, quietly = TRUE)
+   require(lmtest, quietly = TRUE)
+   M <- length(unique(cluster))
+   N <- .... [TRUNCATED]

> load("./Data/KEYFIRM_R.RData")

> # ------------ Aggregate ownership rights type ------------------
> # 0: missing, 1: state/collective, 3: private, 4: HMT, 5: foreign
> KEYFIRM$type .... [TRUNCATED]

> # State and collective
> sel <- which(KEYFIRM$type == 110 | KEYFIRM$type == 141
+     | KEYFIRM$type == 151 | KEYFIRM$type == 120 | KEYFIRM$type ==  ....
[TRUNCATED]

> KEYFIRM$type_a[sel] <- 1

> # Private
> sel <- which(KEYFIRM$type == 170 | KEYFIRM$type == 171
+     | KEYFIRM$type == 172 | KEYFIRM$type == 173
+     | KEYFIRM$type == 174 | K .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 3

> # Hong Kong, Macau and Taiwan
> sel <- which(KEYFIRM$type == 200 | KEYFIRM$type == 210
+     | KEYFIRM$type == 220 | KEYFIRM$type == 230 | KEYFIRM$t .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 4

> # Foreign
> sel <- which(KEYFIRM$type == 300 | KEYFIRM$type == 310
+     | KEYFIRM$type == 320 | KEYFIRM$type == 330 | KEYFIRM$type == 340)

> KEYFIRM$type_a[sel] <- 5
```

```
> # ------------ Aggregate treatment technology type ------------------
> # For disposal equipment type
> KEYFIRM$dm1_code_a <- 0

> # Physical
> sel <- which(KEYFIRM$dm1_code >= 1000 & KEYFIRM$dm1_code < 2000)

> KEYFIRM$dm1_code_a[sel] <- 1

> # Chemical
> sel <- which(KEYFIRM$dm1_code >= 2000 & KEYFIRM$dm1_code < 3000)

> KEYFIRM$dm1_code_a[sel] <- 2

> # Physiochemical
> sel <- which(KEYFIRM$dm1_code >= 3000 & KEYFIRM$dm1_code < 4000)

> KEYFIRM$dm1_code_a[sel] <- 3

> # Biological
> sel <- which(KEYFIRM$dm1_code >= 4000 & KEYFIRM$dm1_code < 5000)

> KEYFIRM$dm1_code_a[sel] <- 4

> # Combination
> sel <- which(KEYFIRM$dm1_code >= 5000 & KEYFIRM$dm1_code < 6000)

> KEYFIRM$dm1_code_a[sel] <- 5

> # --------------- Declare dummies -----------------
> KEYFIRM$province <- factor(KEYFIRM$province)        # Province

> KEYFIRM$industry_a <- factor(KEYFIRM$industry_a)    # 2-digit industry

> KEYFIRM$Census_Type <- factor(KEYFIRM$Census_Type)  # key 1, regular 2

> KEYFIRM$dm1_code_a <- factor(KEYFIRM$dm1_code_a)    # treatment

> KEYFIRM$type_a <- factor(KEYFIRM$type_a)            # ownership rights

> sel <- which(KEYFIRM$industry_a == 15 | KEYFIRM$industry_a == 17
+     | KEYFIRM$industry_a == 22 | KEYFIRM$industry_a == 26
+     | KEYFIRM$indus .... [TRUNCATED]

> ALL <- KEYFIRM[sel]

> ALL <- ALL[product > 0 & cod_e > 0]

> # ================= Regression (!#) Page 10 ==============================
```

```
> lm_all <- lm(log(cod_e) ~ log(product)
+       + province + industry .... [TRUNCATED]

> summary(lm_all)

Call:
lm(formula = log(cod_e) ~ log(product) + province + industry_a +
    type_a, data = ALL)

Residuals:
     Min       1Q   Median       3Q      Max
-13.6730  -1.3044   0.1236   1.4294   7.0787

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.74667    0.17009 -22.028  < 2e-16 ***
log(product)  0.63167    0.00650  97.183  < 2e-16 ***
province12    0.91936    0.20300   4.529 5.95e-06 ***
province13    2.20219    0.17029  12.932  < 2e-16 ***
province14    1.92921    0.19939   9.675  < 2e-16 ***
province15    2.32190    0.20945  11.086  < 2e-16 ***
province21    1.04860    0.17693   5.927 3.13e-09 ***
province22    1.16683    0.20672   5.645 1.67e-08 ***
province23    1.14120    0.20289   5.625 1.87e-08 ***
province31    0.15526    0.17847   0.870   0.3843
province32    0.87671    0.16575   5.289 1.24e-07 ***
province33    1.28765    0.16572   7.770 8.10e-15 ***
province34    1.02464    0.18074   5.669 1.45e-08 ***
province35    1.09303    0.17353   6.299 3.04e-10 ***
province36    1.40749    0.18537   7.593 3.22e-14 ***
province37    1.02133    0.16742   6.100 1.07e-09 ***
province41    1.45293    0.17725   8.197 2.56e-16 ***
province42    1.45913    0.18507   7.884 3.27e-15 ***
province43    1.73715    0.17642   9.847  < 2e-16 ***
province44    0.37106    0.16536   2.244   0.0248 *
province45    2.37634    0.17445  13.622  < 2e-16 ***
province46    1.35907    0.21786   6.238 4.48e-10 ***
province50    0.82452    0.18430   4.474 7.71e-06 ***
province51    1.11428    0.17044   6.538 6.36e-11 ***
province52    1.62967    0.21824   7.467 8.41e-14 ***
province53    1.87311    0.18206  10.288  < 2e-16 ***
province54   -0.82673    1.23624  -0.669   0.5037
province61    1.78717    0.19826   9.014  < 2e-16 ***
province62    2.81659    0.20997  13.414  < 2e-16 ***
province63    2.27521    0.38008   5.986 2.17e-09 ***
province64    3.44462    0.24508  14.055  < 2e-16 ***
province65    1.85097    0.20195   9.165  < 2e-16 ***
industry_a15  0.37125    0.06110   6.076 1.25e-09 ***
industry_a17  0.70557    0.03988  17.695  < 2e-16 ***
industry_a22  1.34876    0.03972  33.960  < 2e-16 ***
```

```
industry_a26 -1.65741    0.03646 -45.454  < 2e-16 ***
type_a3      -0.18557    0.02870  -6.466 1.02e-10 ***
type_a4      -0.46878    0.05507  -8.513  < 2e-16 ***
type_a5      -0.44802    0.04846  -9.245  < 2e-16 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 2.121 on 29014 degrees of freedom
Multiple R-squared:  0.4137,    Adjusted R-squared:  0.4129
F-statistic: 538.7 on 38 and 29014 DF,  p-value: < 2.2e-16


> cl(ALL, lm_all, ALL$province)

t test of coefficients:

              Estimate Std. Error  t value  Pr(>|t|)
(Intercept)  -3.746665   0.296323 -12.6439 < 2.2e-16 ***
log(product)  0.631672   0.030517  20.6993 < 2.2e-16 ***
province12    0.919362   0.049603  18.5345 < 2.2e-16 ***
province13    2.202186   0.047127  46.7284 < 2.2e-16 ***
province14    1.929210   0.037036  52.0903 < 2.2e-16 ***
province15    2.321900   0.051610  44.9896 < 2.2e-16 ***
province21    1.048604   0.027495  38.1381 < 2.2e-16 ***
province22    1.166831   0.037086  31.4630 < 2.2e-16 ***
province23    1.141198   0.034574  33.0072 < 2.2e-16 ***
province31    0.155265   0.062142   2.4986 0.0124755 *
province32    0.876714   0.057514  15.2436 < 2.2e-16 ***
province33    1.287647   0.059522  21.6330 < 2.2e-16 ***
province34    1.024643   0.036279  28.2432 < 2.2e-16 ***
province35    1.093033   0.041218  26.5184 < 2.2e-16 ***
province36    1.407485   0.050195  28.0403 < 2.2e-16 ***
province37    1.021327   0.016957  60.2287 < 2.2e-16 ***
province41    1.452927   0.040033  36.2932 < 2.2e-16 ***
province42    1.459134   0.036241  40.2620 < 2.2e-16 ***
province43    1.737147   0.065789  26.4048 < 2.2e-16 ***
province44    0.371061   0.050971   7.2799 3.425e-13 ***
province45    2.376341   0.058849  40.3802 < 2.2e-16 ***
province46    1.359070   0.053419  25.4418 < 2.2e-16 ***
province50    0.824519   0.053006  15.5553 < 2.2e-16 ***
province51    1.114278   0.054969  20.2711 < 2.2e-16 ***
province52    1.629667   0.069490  23.4519 < 2.2e-16 ***
province53    1.873106   0.047122  39.7499 < 2.2e-16 ***
province54   -0.826725   0.117045  -7.0633 1.663e-12 ***
province61    1.787174   0.028097  63.6082 < 2.2e-16 ***
province62    2.816591   0.062099  45.3565 < 2.2e-16 ***
province63    2.275214   0.036998  61.4960 < 2.2e-16 ***
province64    3.444618   0.044963  76.6107 < 2.2e-16 ***
province65    1.850967   0.043709  42.3472 < 2.2e-16 ***
industry_a15  0.371251   0.180039   2.0621 0.0392110 *
```

```
industry_a17  0.705573   0.194711   3.6237 0.0002909 ***
industry_a22  1.348758   0.197344   6.8345 8.389e-12 ***
industry_a26 -1.657409   0.194931  -8.5025 < 2.2e-16 ***
type_a3      -0.185568   0.058757  -3.1582 0.0015890 **
type_a4      -0.468784   0.115161  -4.0707 4.700e-05 ***
type_a5      -0.448016   0.089898  -4.9836 6.277e-07 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1


> # ============================================================================
> #         3. Additional Information of Treatment Technologies
> # === .... [TRUNCATED]

> # ====================== Figure D.3 =======================================
> load("./Data/KEYFIRM_R.RData")

> # ------------ Aggregate ownership rights type ------------------
> # 0: missing, 1: State/collective, 3: private, 4: HMT, 5: foreign
> KEYFIRM$type .... [TRUNCATED]

> # State and collective
> sel <- which(KEYFIRM$type == 110 | KEYFIRM$type == 141
+     | KEYFIRM$type == 151 | KEYFIRM$type == 120 | KEYFIRM$type ==  ....
[TRUNCATED]

> KEYFIRM$type_a[sel] <- 1

> # Private
> sel <- which(KEYFIRM$type == 170 | KEYFIRM$type == 171
+              | KEYFIRM$type == 172 | KEYFIRM$type == 173
+              | KEYFI .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 3

> # Hong Kong, Macau and Taiwan
> sel <- which(KEYFIRM$type == 200 | KEYFIRM$type == 210
+              | KEYFIRM$type == 220 | KEYFIRM$type == 230
+ .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 4

> # Foreign
> sel <- which(KEYFIRM$type == 300 | KEYFIRM$type == 310
+              | KEYFIRM$type == 320 | KEYFIRM$type == 330
+              | KEYF .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 5

> # ------------ Aggregate treatment technology type ------------------
```

```
> # For disposal equipment type
> KEYFIRM$dm1_code_a <- 0

> # Physical
> sel <- which(KEYFIRM$dm1_code >= 1000 & KEYFIRM$dm1_code < 2000)

> KEYFIRM$dm1_code_a[sel] <- 1

> # Chemical
> sel <- which(KEYFIRM$dm1_code >= 2000 & KEYFIRM$dm1_code < 3000)

> KEYFIRM$dm1_code_a[sel] <- 2

> # Physiochemical
> sel <- which(KEYFIRM$dm1_code >= 3000 & KEYFIRM$dm1_code < 4000)

> KEYFIRM$dm1_code_a[sel] <- 3

> # Biological
> sel <- which(KEYFIRM$dm1_code >= 4000 & KEYFIRM$dm1_code < 5000)

> KEYFIRM$dm1_code_a[sel] <- 4

> # Combination
> sel <- which(KEYFIRM$dm1_code >= 5000 & KEYFIRM$dm1_code < 6000)

> KEYFIRM$dm1_code_a[sel] <- 5

> # --------------- Declare dummies -----------------
> KEYFIRM$province <- factor(KEYFIRM$province)        # Province

> KEYFIRM$industry_a <- factor(KEYFIRM$industry_a)    # 2-digit industry

> KEYFIRM$Census_Type <- factor(KEYFIRM$Census_Type)  # key 1, regular 2

> KEYFIRM$dm1_code_a <- factor(KEYFIRM$dm1_code_a)    # treatment

> KEYFIRM$type_a <- factor(KEYFIRM$type_a)            # ownership rights

> GB <- KEYFIRM[industry_a == 22]

> GB <- GB[product > 0]

> GB <- GB[cod_e >= 0 & is.na(cod_e)==FALSE]

> GB <- GB[cod_g > 0]

> GB <- GB[dm1_inv > 0]

> GB <- within(GB,{
```

```
+   cod_eg <- cod_e/cod_g
+   dm1_unit <- dm1_quant/dm1_inv
+   dm1_prod <- dm1_inv/product
+   dm1_prod2 <- (dm1_inv + dm1_oprcos .... [TRUNCATED]

> # Plot Densities
> pdf("./Results/FigureD3_TopRight.pdf",height=5,width=5)

> tmpden <- density(log(GB$dm1_inv[GB$dm1_code_a == 1]),
+                   kernel = "gaussian", bw =0.50, na.rm = TRUE)

> plot(tmpden,xlab="Costs",ylab="Density",main="Costs",
+      ylim=c(0,0.35),cex.main=1.50,cex.lab=1.5,lwd = 2.0)

> lines(density(log(GB$dm1_inv[GB$dm1_code_a == 2]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),
+       col=2, lwd = 2.0)

> lines(density(log(GB$dm1_inv[GB$dm1_code_a == 5 | GB$dm1_code_a == 4]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),
+       col=4, .... [TRUNCATED]

> legend("topleft",c("Phy","Chem","Bio"),
+        lty=c(1,1,1), col=c(1,2,4),lwd = 2.0)

> dev.off()
null device
          1

> pdf("./Results/FigureD3_TopLeft.pdf",height=5,width=5)

> tmpden <- density(log(GB$dm1_quant[GB$dm1_code_a == 1]),
+                   kernel = "gaussian", bw =0.50, na.rm = TRUE)

> plot(tmpden,xlab="Capacity",ylab="Density",main="Processing Capacity",
+      ylim=c(0,0.25),cex.main=1.50,cex.lab=1.5,lwd = 2.0)

> lines(density(log(GB$dm1_quant[GB$dm1_code_a == 2]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=2,lwd = 2.0)

> lines(density(log(GB$dm1_quant[GB$dm1_code_a == 4 | GB$dm1_code_a == 5]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=4,lwd = 2 ....
[TRUNCATED]

> legend("topleft",c("Phy","Chem","Bio"),
+        lty=c(1,1,1), col=c(1,2,4),lwd =2.0)

> dev.off()
null device
          1
```

```
> pdf("./Results/FigureD3_BotLeft.pdf",height=5,width=5)

> tmpden <- density(log(GB$dm1_unit[GB$dm1_code_a == 1]),
+                   kernel = "gaussian", bw =0.50, na.rm = TRUE)

> plot(tmpden,xlab="Unit-cost",ylab="Density",main="Unit Capacity Cost",
+      ylim=c(0,0.40),cex.main=1.50,cex.lab=1.5,lwd = 2.0)

> lines(density(log(GB$dm1_unit[GB$dm1_code_a == 2]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=2,lwd = 2.0)

> lines(density(log(GB$dm1_unit[GB$dm1_code_a == 4 | GB$dm1_code_a == 5]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=4,lwd = 2. ....
[TRUNCATED]

> legend("topleft",c("Phy","Chem","Bio"),
+        lty=c(1,1,1), col=c(1,2,4),lwd = 2.0)

> dev.off()
null device
          1

> pdf("./Results/FigureD3_BotRight.pdf",height=5,width=5)

> tmpden <- density(log(GB$product[GB$dm1_code_a == 1]),
+                   kernel = "gaussian", bw =0.50, na.rm = TRUE)

> plot(tmpden,xlab="Production",ylab="Density",main="Production Scale",
+      ylim=c(0,0.40),cex.main=1.50,cex.lab=1.5,lwd= 2.0)

> lines(density(log(GB$product[GB$dm1_code_a == 2]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=2,lwd= 2.0)

> lines(density(log(GB$product[GB$dm1_code_a == 4 | GB$dm1_code_a == 5]),
+               kernel = "gaussian", bw =0.50, na.rm = TRUE),col=4,lwd= 2.0)

> legend("topleft",c("Phy","Chem","Bio"),
+        lty=c(1,1,1), col=c(1,2,4),lwd= 2.0)

> dev.off()
null device
          1

> # ==================== Results in Appendix D.2 ============================
> # Clear memory
> rm(list = ls())

> load("./Data/KEYFIRM_R.RData")
```

```
> # ------------ Aggregate ownership rights type ------------------
> # 0: missing, 1: State/collective, 3: private, 4: HMT, 5: foreign
> KEYFIRM$type .... [TRUNCATED]

> # State and collective
> sel <- which(KEYFIRM$type == 110 | KEYFIRM$type == 141
+     | KEYFIRM$type == 151 | KEYFIRM$type == 120 | KEYFIRM$type ==  ....
[TRUNCATED]

> KEYFIRM$type_a[sel] <- 1

> # Private
> sel <- which(KEYFIRM$type == 170 | KEYFIRM$type == 171
+              | KEYFIRM$type == 172 | KEYFIRM$type == 173
+              | KEYFI .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 3

> # Hong Kong, Macau and Taiwan
> sel <- which(KEYFIRM$type == 200 | KEYFIRM$type == 210
+     | KEYFIRM$type == 220 | KEYFIRM$type == 230 | KEYFIRM$t .... [TRUNCATED]

> KEYFIRM$type_a[sel] <- 4

> # Foreign
> sel <- which(KEYFIRM$type == 300 | KEYFIRM$type == 310
+     | KEYFIRM$type == 320 | KEYFIRM$type == 330 | KEYFIRM$type == 340)

> KEYFIRM$type_a[sel] <- 5

> # ------------ Aggregate treatment technology type ------------------
> # For disposal equipment type
> KEYFIRM$dm1_code_a <- 0

> # Physical
> sel <- which(KEYFIRM$dm1_code >= 1000 & KEYFIRM$dm1_code < 2000)

> KEYFIRM$dm1_code_a[sel] <- 1

> # Chemical
> sel <- which(KEYFIRM$dm1_code >= 2000 & KEYFIRM$dm1_code < 3000)

> KEYFIRM$dm1_code_a[sel] <- 2

> # Physiochemical
> sel <- which(KEYFIRM$dm1_code >= 3000 & KEYFIRM$dm1_code < 4000)

> KEYFIRM$dm1_code_a[sel] <- 3

> # Biological
```

```
> sel <- which(KEYFIRM$dm1_code >= 4000 & KEYFIRM$dm1_code < 5000)

> KEYFIRM$dm1_code_a[sel] <- 4

> # Combination
> sel <- which(KEYFIRM$dm1_code >= 5000 & KEYFIRM$dm1_code < 6000)

> KEYFIRM$dm1_code_a[sel] <- 5

> # --------------- Declare dummies -----------------
> KEYFIRM$province <- factor(KEYFIRM$province)

> KEYFIRM$industry_a <- factor(KEYFIRM$industry_a)

> KEYFIRM$Census_Type <- factor(KEYFIRM$Census_Type)

> KEYFIRM$dm1_code_a <- factor(KEYFIRM$dm1_code_a)

> KEYFIRM$type_a <- factor(KEYFIRM$type_a)

> KEYFIRM <- KEYFIRM[product > 0 & cod_e > 0]

> sel <- which(KEYFIRM$industry_a == 15 | KEYFIRM$industry_a == 17
+     | KEYFIRM$industry_a == 22 | KEYFIRM$industry_a == 26
+     | KEYFIRM$indus .... [TRUNCATED]

> POL5 <- KEYFIRM[sel]

> sel <- which(POL5$dm1_inv > 0)

> POL5 <- POL5[sel]

> POL5$dm_y <- with(POL5, dm_y <- dm1_inv/product)

> qdown <- quantile(POL5$dm_y, probs=c(0.01), na.rm = TRUE)

> qup <- quantile(POL5$dm_y, probs=c(0.99), na.rm = TRUE)

> sel <- which(POL5$dm_y > qdown & POL5$dm_y < qup)

> POL5 <- POL5[sel]

> # ====================== Ratios on page 14 ================================
> qdown <- quantile(POL5$product, probs=c(0.001), na.rm = TRUE)

> qup <- quantile(POL5$product, probs=c(0.20), na.rm = TRUE)

> sel <- which(POL5$product > qdown & POL5$product < qup)

> quint1 <- mean(POL5$dm_y[sel])
```

```
> qdown <- quantile(POL5$product, probs=c(0.20), na.rm = TRUE)

> qup <- quantile(POL5$product, probs=c(0.40), na.rm = TRUE)

> sel <- which(POL5$product > qdown & POL5$product < qup)

> quint2 <- mean(POL5$dm_y[sel])

> qdown <- quantile(POL5$product, probs=c(0.40), na.rm = TRUE)

> qup <- quantile(POL5$product, probs=c(0.60), na.rm = TRUE)

> sel <- which(POL5$product > qdown & POL5$product < qup)

> quint3 <- mean(POL5$dm_y[sel])

> qdown <- quantile(POL5$product, probs=c(0.60), na.rm = TRUE)

> qup <- quantile(POL5$product, probs=c(0.80), na.rm = TRUE)

> sel <- which(POL5$product > qdown & POL5$product < qup)

> quint4 <- mean(POL5$dm_y[sel])

> qdown <- quantile(POL5$product, probs=c(0.80), na.rm = TRUE)

> qup <- quantile(POL5$product, probs=c(0.99), na.rm = TRUE)

> sel <- which(POL5$product > qdown & POL5$product < qup)

> quint5 <- mean(POL5$dm_y[sel])

> quint <- c(quint1,quint2,quint3,quint4,quint5)

> quint
[1] 0.22773883 0.10844430 0.06723317 0.04501052 0.02719179

> # ================== Regression on Page 14 ================================
> lm_dmy <- lm(log(dm_y) ~ log(product) + province + type_a
+          .... [TRUNCATED]

> summary(lm_dmy)

Call:
lm(formula = log(dm_y) ~ log(product) + province + type_a + industry_a,
    data = POL5)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-5.2663 -0.8492  0.0823  0.9316  5.3019

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.732001   0.111173  -6.584 4.65e-11 ***
log(product) -0.404079   0.004274 -94.533  < 2e-16 ***
province12    0.021892   0.132187   0.166 0.868460
province13   -0.067579   0.110908  -0.609 0.542318
province14   -0.029424   0.129386  -0.227 0.820103
province15   -0.564774   0.135786  -4.159 3.20e-05 ***
province21   -0.136697   0.115285  -1.186 0.235740
province22    0.499503   0.135105   3.697 0.000218 ***
province23    0.416848   0.131805   3.163 0.001565 **
province31   -0.282615   0.116240  -2.431 0.015050 *
province32   -0.095530   0.108038  -0.884 0.376582
province33   -0.228779   0.108017  -2.118 0.034185 *
province34    0.035578   0.117499   0.303 0.762050
province35   -0.301867   0.113053  -2.670 0.007587 **
province36   -0.561112   0.120375  -4.661 3.16e-06 ***
province37    0.174192   0.109077   1.597 0.110284
province41    0.209417   0.115394   1.815 0.069565 .
province42   -0.211726   0.120376  -1.759 0.078609 .
province43   -0.777333   0.114815  -6.770 1.31e-11 ***
province44   -0.330826   0.107823  -3.068 0.002155 **
province45   -0.438602   0.113589  -3.861 0.000113 ***
province46    0.565547   0.141822   3.988 6.69e-05 ***
province50   -0.540336   0.119911  -4.506 6.63e-06 ***
province51   -0.304713   0.111156  -2.741 0.006123 **
province52   -0.429761   0.141967  -3.027 0.002471 **
province53   -0.211960   0.118482  -1.789 0.073632 .
province54   -0.166936   0.787925  -0.212 0.832212
province61   -0.059659   0.128972  -0.463 0.643676
province62   -0.161960   0.135912  -1.192 0.233406
province63   -0.238430   0.248628  -0.959 0.337576
province64   -0.145092   0.160553  -0.904 0.366160
province65   -0.612133   0.131910  -4.641 3.49e-06 ***
type_a3      -0.202528   0.018475 -10.962  < 2e-16 ***
type_a4      -0.044801   0.035509  -1.262 0.207078
type_a5      -0.006529   0.031262  -0.209 0.834577
industry_a15  0.816052   0.039374  20.726  < 2e-16 ***
industry_a17  0.981062   0.025715  38.151  < 2e-16 ***
industry_a22  0.798032   0.025544  31.241  < 2e-16 ***
industry_a26  0.022878   0.023483   0.974 0.329936
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 1.351 on 28430 degrees of freedom
Multiple R-squared:  0.3146,    Adjusted R-squared:  0.3137
F-statistic: 343.4 on 38 and 28430 DF,  p-value: < 2.2e-16
```

```
> # ========================== Table D1 ====================================
> phy <- POL5[POL5$dm1_code_a == 1]

> qdown <- quantile(phy$dm1_inv, probs=c(0.01), na.rm = TRUE)

> qup <- quantile(phy$dm1_inv, probs=c(0.99), na.rm = TRUE)

> sel <- which(phy$dm1_inv > qdown & phy$product < qup)

> phy <- phy[sel]

> summary(phy$dm1_inv)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.24    1.40    3.50   12.41   10.00 1000.00

> bio <- POL5[POL5$dm1_code_a == 4 | POL5$dm1_code_a == 5]

> qdown <- quantile(bio$dm1_inv, probs=c(0.01), na.rm = TRUE)

> qup <- quantile(bio$dm1_inv, probs=c(0.99), na.rm = TRUE)

> sel <- which(bio$dm1_inv > qdown & bio$product < qup)

> bio <- bio[sel]

> summary(bio$dm1_inv)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.05   20.00   50.00  109.60  120.00 3556.00

> # ============================================================================
> #                 4. Correlated Distortions
> # ==================== .... [TRUNCATED]

> # ====================== Figure D1 ========================================
> load("./Data/CNEC_avgp.RData")

> CNEC <- CNEC_avgp

> rm(CNEC_avgp)

> # General sample selection:
> #   regular producing firms with positive production, capital and labor
> CNEC <- CNEC[status == 1]

> CNEC <- CNEC[product > 0]

> CNEC <- CNEC[totcapital > 0]

> CNEC <- CNEC[nbarworkers > 0]
```

```
> CNEC <- CNEC[wage + nonwage > 0]

> PAPER <- CNEC[industry_a == 22]

> PAPER <- within(PAPER,
+                 {lcomp <- wage + nonwage
+                 age <- 2005 - founding_y
+                 })

> alpha = 0.5376

> gamma = 0.93

> # Calculate average factor products
> PAPER <- within(PAPER,
+           {phik <- product/totcapital
+            phil <- product/lcomp
+            .... [TRUNCATED]

> qup <- quantile(PAPER$phik, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(PAPER$phik, probs=c(0.03),na.rm=TRUE)

> sel <- which(PAPER$phik>qdown & PAPER$phik<qup)

> phikz <- PAPER[sel]

> qup <- quantile(phikz$z2, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(phikz$z2, probs=c(0.03),na.rm=TRUE)

> sel <- which(phikz$z2>qdown & phikz$z2<qup)

> phikz <- phikz[sel]

> qup <- quantile(PAPER$phil, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(PAPER$phil, probs=c(0.03),na.rm=TRUE)

> sel <- which(PAPER$phil>qdown & PAPER$phil<qup)

> philz <- PAPER[sel]

> qup <- quantile(philz$z2, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(philz$z2, probs=c(0.03),na.rm=TRUE)

> sel <- which(philz$z2>qdown & philz$z2<qup)
```

```
> philz <- philz[sel]

> qup <- quantile(PAPER$kappa, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(PAPER$kappa, probs=c(0.03),na.rm=TRUE)

> sel <- which(PAPER$kappa>qdown & PAPER$kappa<qup)

> kappaz <- PAPER[sel]

> qup <- quantile(kappaz$z2, probs=c(.97),na.rm=TRUE)

> qdown <- quantile(kappaz$z2, probs=c(0.03),na.rm=TRUE)

> sel <- which(kappaz$z2>qdown & kappaz$z2<qup)

> kappaz <- kappaz[sel]

> # ================== Figure D1 ============================================
> # Plot ARPK
> pdf("./Results/FigureD1_TopLeft.pdf",height=5,width=5)

> plot(log(phikz$phik)~log(phikz$z2),cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Productivity",ylab="ARK",
+      main="Average Product of Capital",ce .... [TRUNCATED]

> p_lm <- lm(log(phik)~log(z2), data = phikz)

> abline(p_lm,col="red",lwd=4)

> dev.off()
null device
          1

> # Plot ARPL
> pdf("./Results/FigureD1_TopRight.pdf",height=5,width=5)

> plot(log(philz$phil)~log(philz$z2),cex=0.5,mgp=c(1.75, 0.75, 0),
+      xlab="Productivity",ylab="ARL",
+      main="Average Product of Labor",cex. .... [TRUNCATED]

> p_lm <- lm(log(phil)~log(z2), data = philz)

> abline(p_lm,col="red",lwd=4)

> dev.off()
null device
          1

> # Plot k/l ratio
```

```
> pdf("./Results/FigureD1_Bot.pdf",height=5,width=5)

> plot(log(kappaz$kappa)~log(kappaz$z2),cex=0.5,
+      mgp=c(1.75, 0.75, 0), xlab="Productivity",ylab="K/L Ratio",
+      main="Capital-labor Ratio", .... [TRUNCATED]

> p_lm <- lm(log(kappa)~log(z2), data = kappaz)

> abline(p_lm,col="red",lwd=4)

> dev.off()
null device
          1

> # ============================================================================
> #         5. Employment Distributions by Industry
> # =============== .... [TRUNCATED]

> # Load in packages
> library(foreign)

> library(data.table)

> library(AER)

> library(ggplot2)

> library(scales)

> library(grid)

> load("./Data/CNEC_avgp.RData")

> CHNall <- CNEC_avgp

> USall <- read.csv("./Data/susb04.csv",header = TRUE)

> USall <- as.data.table(USall)

> USall <- USall[,list(NAICS, ENTRSIZE, FIRM,
+                      ESTB, EMPL, NAICSDSCR, ENTRSIZEDSCR)]

> # ------------------------ Paper Industry ---------------------------
> sel <- which(CHNall$industry_a == 22)

> CH <- CHNall[sel]

> qch <- quantile(CH$nbarworkers, probs=c(.99),na.rm=TRUE)

> sel <- which(CH$nbarworkers < qch)
```

```
> CH <- CH[sel]

> sel <- which(USall$NAICS == 3221)

> US <- USall[sel]

> sel <- which(US$ENTRSIZE != 1 & US$ENTRSIZE != 6 &  US$ENTRSIZE != 9)

> US <- US[sel]

> cf <- c(0,4,9,14,19,24,29,34,39,44,49,74,99,
+         149,199,299,399,499,749,999,1499,2499,5000)

> US <- within(US,
+              {AVGF <- EMPL/FIRM   # Average firm size
+              AVGE <- EMPL/ESTB   # Average plant size
+              CF < .... [TRUNCATED]

> # Calculate the employment share
> cutoff <- c(1,19,99,399)

> n1 <- length(cutoff)

> distchn <- rep(0,n1)

> distus <- distchn

> for (i in 2:n1){
+   sel <- which(US$AVGF > cutoff[i-1] & US$AVGF <= cutoff[i])
+   distus[i-1] <- sum(US$EMPL[sel])
+   sel1 <- which(CH$nbarworker .... [TRUNCATED]

> # Last category
> sel <- which(US$AVGF > cutoff[n1])

> distus[n1] <- sum(US$EMPL[sel])

> distus <- distus/sum(distus)

> sel1 <- which(CH$nbarworkers > cutoff[n1])

> distchn[n1] <- sum(CH$nbarworkers[sel1])

> distchn <- distchn/sum(distchn)

> # ========================== Figure E1 ====================================
> pdf("./Results/FigureE1_TopLeft.pdf",height=6,width=7.5)

> barplot(rbind(distchn,distus),beside=TRUE,col=c("red","blue"),
```

```
+         ylim=c(0,1.0),xlab="Firm Size",ylab="Employment Share",
+         main="Pa ..." ... [TRUNCATED]

> legend("topleft", c("China","US"),fill=c("red","blue"),
+        bty="o",y.intersp=0.7,x.intersp=0.3,text.width=0.8)

> dev.off()
null device
          1

> # ---------------- Textile Industry -----------------------------
> sel <- which(CHNall$industry_a == 17)

> CH <- CHNall[sel]

> qch <- quantile(CH$nbarworkers, probs=c(.99),na.rm=TRUE)

> sel <- which(CH$nbarworkers < qch)

> CH <- CH[sel]

> sel <- which(USall$NAICS == 313)

> US <- USall[sel]

> sel <- which(US$ENTRSIZE != 1 & US$ENTRSIZE != 6 &  US$ENTRSIZE != 9)

> US <- US[sel]

> cf <- c(0,4,9,14,19,24,29,34,39,44,49,74,99,
+         149,199,299,399,499,749,999,1499,2499,5000)

> US <- within(US,
+              {AVGF <- EMPL/FIRM   # Average firm size
+              AVGE <- EMPL/ESTB   # Average plant size
+              CF < .... [TRUNCATED]

> # Calculate the employment share
> cutoff <- c(1,19,99,399)

> n1 <- length(cutoff)

> distchn <- rep(0,n1)

> distus <- distchn

> for (i in 2:n1){
+   sel <- which(US$AVGF > cutoff[i-1] & US$AVGF <= cutoff[i])
+   distus[i-1] <- sum(US$EMPL[sel])
+   sel1 <- which(CH$nbarworker .... [TRUNCATED]
```

```
> # Last category
> sel <- which(US$AVGF > cutoff[n1])

> distus[n1] <- sum(US$EMPL[sel])

> distus <- distus/sum(distus)

> sel1 <- which(CH$nbarworkers > cutoff[n1])

> distchn[n1] <- sum(CH$nbarworkers[sel1])

> distchn <- distchn/sum(distchn)

> # ========================== Figure E1 ====================================
> pdf("./Results/FigureE1_MidLeft.pdf",height=6,width=7.5)

> barplot(rbind(distchn,distus),beside=TRUE,col=c("red","blue"),
+         ylim=c(0,1.0),xlab="Firm Size",ylab="Employment Share",
+         main="Tex ..." ... [TRUNCATED]

> legend("topleft", c("China","US"),fill=c("red","blue"),
+        bty="o",y.intersp=0.7,x.intersp=0.3,text.width=0.8)

> dev.off()
null device
          1

> # ------------------ Food Industry -----------------------------
> sel <- which(CHNall$industry_a == 13)

> CH <- CHNall[sel]

> qch <- quantile(CH$nbarworkers, probs=c(.99),na.rm=TRUE)

> sel <- which(CH$nbarworkers < qch)

> CH <- CH[sel]

> sel <- which(USall$NAICS == 311)

> US <- USall[sel]

> sel <- which(US$ENTRSIZE != 1 & US$ENTRSIZE != 6 &  US$ENTRSIZE != 9)

> US <- US[sel]

> cf <- c(0,4,9,14,19,24,29,34,39,44,49,74,99,
+         149,199,299,399,499,749,999,1499,2499,5000)
```

```
> US <- within(US,
+              {AVGF <- EMPL/FIRM   # Average firm size
+              AVGE <- EMPL/ESTB   # Average plant size
+              CF < .... [TRUNCATED]

> # Calculate the employment share
> cutoff <- c(1,19,99,399)

> n1 <- length(cutoff)

> distchn <- rep(0,n1)

> distus <- distchn

> for (i in 2:n1){
+   sel <- which(US$AVGF > cutoff[i-1] & US$AVGF <= cutoff[i])
+   distus[i-1] <- sum(US$EMPL[sel])
+   sel1 <- which(CH$nbarworker .... [TRUNCATED]

> # Last category
> sel <- which(US$AVGF > cutoff[n1])

> distus[n1] <- sum(US$EMPL[sel])

> distus <- distus/sum(distus)

> sel1 <- which(CH$nbarworkers > cutoff[n1])

> distchn[n1] <- sum(CH$nbarworkers[sel1])

> distchn <- distchn/sum(distchn)

> # ========================== Figure E1 ===================================
> pdf("./Results/FigureE1_TopRight.pdf",height=6,width=7.5)

> barplot(rbind(distchn,distus),beside=TRUE,col=c("red","blue"),
+         ylim=c(0,1.0),xlab="Firm Size",ylab="Employment Share",
+         main="Agr ..." ... [TRUNCATED]

> legend("topleft", c("China","US"),fill=c("red","blue"),
+        bty="o",y.intersp=0.7,x.intersp=0.3,text.width=0.8)

> dev.off()
null device
          1

> # --------------- Chemistry Industry ----------------------------
> sel <- which(CHNall$industry_a == 26)

> CH <- CHNall[sel]
```

```
> qch <- quantile(CH$nbarworkers, probs=c(.99),na.rm=TRUE)

> sel <- which(CH$nbarworkers < qch)

> CH <- CH[sel]

> sel <- which(USall$NAICS == 3251 | USall$NAICS == 3252
+              | USall$NAICS == 3253 | USall$NAICS == 3255
+              | USall$NAICS ==  .... [TRUNCATED]

> US <- USall[sel]

> sel <- which(US$ENTRSIZE != 1
+              & US$ENTRSIZE != 6 &  US$ENTRSIZE != 9)

> US <- US[sel]

> USSUM <- US[, list(FIRM=sum(FIRM, na.rm = TRUE),
+                    ESTB=sum(ESTB, na.rm = TRUE),
+                    EMPL=sum(EMPL, na.rm = TR .... [TRUNCATED]

> cf <- c(0,4,9,14,19,24,29,34,39,44,49,74,99,
+         149,199,299,399,499,749,999,1499,2499,5000)

> USSUM <- within(USSUM,
+                 {AVGF <- EMPL/FIRM   # Average firm size
+                 AVGE <- EMPL/ESTB   # Average plant size
+       .... [TRUNCATED]

> # Calculate the employment share
> cutoff <- c(1,19,99,399)

> n1 <- length(cutoff)

> distchn <- rep(0,n1)

> distus <- distchn

> for (i in 2:n1){
+   sel <- which(USSUM$AVGF > cutoff[i-1] & USSUM$AVGF <= cutoff[i])
+   distus[i-1] <- sum(USSUM$EMPL[sel])
+   sel1 <- which(CH$n .... [TRUNCATED]

> # Last category
> sel <- which(USSUM$AVGF > cutoff[n1])

> distus[n1] <- sum(USSUM$EMPL[sel])

> distus <- distus/sum(distus)
```

```
> sel1 <- which(CH$nbarworkers > cutoff[n1])

> distchn[n1] <- sum(CH$nbarworkers[sel1])

> distchn <- distchn/sum(distchn)

> # ========================== Figure E1 ===================================
> pdf("./Results/FigureE3_MidRight.pdf",height=6,width=7.5)

> barplot(rbind(distchn,distus),beside=TRUE,col=c("red","blue"),
+         ylim=c(0,1.0),xlab="Firm Size",ylab="Employment Share",
+         main="Che ..." ... [TRUNCATED]

> legend("topleft", c("China","US"),fill=c("red","blue"),
+        bty="o",y.intersp=0.7,x.intersp=0.3,text.width=0.8)

> dev.off()
null device
          1

> # --------------- Beverage Industry ------------------------
> sel <- which(CHNall$industry_a == 15)

> CH <- CHNall[sel]

> qch <- quantile(CH$nbarworkers, probs=c(.99),na.rm=TRUE)

> sel <- which(CH$nbarworkers < qch)

> CH <- CH[sel]

> sel <- which(USall$NAICS == 3121)

> US <- USall[sel]

> sel <- which(US$ENTRSIZE != 1
+              & US$ENTRSIZE != 6 &  US$ENTRSIZE != 9)

> US <- US[sel]

> cf <- c(0,4,9,14,19,24,29,34,39,44,49,74,99,
+         149,199,299,399,499,749,999,1499,2499,5000)

> US <- within(US,
+              {AVGF <- EMPL/FIRM   # Average firm size
+              AVGE <- EMPL/ESTB   # Average plant size
+              CF < .... [TRUNCATED]

> # Calculate the employment share
```

```
> cutoff <- c(1,19,99,399)

> n1 <- length(cutoff)

> distchn <- rep(0,n1)

> distus <- distchn

> for (i in 2:n1){
+   sel <- which(US$AVGF > cutoff[i-1] & US$AVGF <= cutoff[i])
+   distus[i-1] <- sum(US$EMPL[sel])
+   sel1 <- which(CH$nbarworker .... [TRUNCATED]

> # Last category
> sel <- which(US$AVGF > cutoff[n1])

> distus[n1] <- sum(US$EMPL[sel])

> distus <- distus/sum(distus)

> sel1 <- which(CH$nbarworkers > cutoff[n1])

> distchn[n1] <- sum(CH$nbarworkers[sel1])

> distchn <- distchn/sum(distchn)

> # ========================== Figure E1 ====================================
> pdf("./Results/FigureE3_BotLeft.pdf",height=6,width=7.5)

> barplot(rbind(distchn,distus),beside=TRUE,col=c("red","blue"),
+         ylim=c(0,1.0),xlab="Firm Size",ylab="Employment Share",
+         main="Bev ..." ... [TRUNCATED]

> legend("topleft", c("China","US"),fill=c("red","blue"),
+        bty="o",y.intersp=0.7,x.intersp=0.3,text.width=0.8)

> dev.off()
null device
          1
```