Log File of Code Running for Floyd, Lee, and Tomar (2023)

2023-04-05

This log file shows the code running to produce the figures and tables in Floyd, Lee, and Tomar (2023). Search "FIGURE 3" or "TABLE 7", et cetera, to find the point where that figure or table is produced.

```
### SETUP ================================================================
library(collapse);
library(data.table);
library(ggplot2);
library(lfe);
library(RColorBrewer);
library(stargazer);
library(stringdist);
library(tables);

rm(list= ls());

set.seed(7777777);

setwd("/Users/gnd");

legitN <- function(dataInput) {
  return(!is.na(dataInput) & !is.infinite(dataInput) & !is.nan(dataInput));
}

renamer <- function(dataInput, separator) {
  simpleCap <- function(stringInput, direction) {
    s <- strsplit(stringInput, " ")[[1]];
    if (direction== "upper") {return(paste(toupper(substring(s, 1,1)),
                                     substring(s, 2), sep="", collapse=" "));
    }
    if (direction== "lower") {return(paste(tolower(substring(s, 1,1)),
```

```
                                        substring(s, 2), sep="", collapse=" "));
    }}}

Mean <- function(x) signif(base::mean(x, na.rm=TRUE), 3);
SD <- function(x) signif(stats::sd(x, na.rm=TRUE), 3);
q5 <- function(x) signif(stats::quantile(x, 0.01, na.rm=TRUE), 2);
q25 <- function(x) signif(stats::quantile(x, 0.25, na.rm=TRUE), 2);
q50 <- function(x) signif(stats::quantile(x, 0.50, na.rm=TRUE), 2);
q75 <- function(x) signif(stats::quantile(x, 0.75, na.rm=TRUE), 2);
q95 <- function(x) signif(stats::quantile(x, 0.99, na.rm=TRUE), 2);


### LABOR ======================================================================
gra <- data.table(read.csv("gra.csv"));
jof <- data.table(read.csv("jof.csv"));
ts <- data.table(read.csv("ts.csv"));
jra <- data.table(read.csv("jra.csv"));
inds <- data.table(read.csv("spInd.csv"));
cpi <- data.table(read.csv("cpi.csv"));


### CLEAN GRADES ===============================================================
gra <- gra[
  ][order(id, termYr)
  ][, c("year", "term", "nterms"):=
      .(termYr %/% 10,
        termYr %% 10,
        .N)
    , id
  ][, c("firstYr", "firstTerm"):=
      .(max(ifelse(1:.N == 1, year, NA), na.rm= T),
        max(ifelse(1:.N == 1, term, NA), na.rm= T))
    , id];


### CLEAN JOB OFFERS ===========================================================
jof <- jof[
  ][decisionDate!= as.Date("1900-01-01") & !is.na(id) &
```

```
      jobOfferStatus== "Accepted"
  ][, c("fixedSalary", "negFixed"):=
      .(sum(baseSalary, earlySignonBonusAmt, signingstartingBonusAmt,
            gauranteedYeaarendBonusAmt, tuitionReimbursementBonusAmt,
            relocationExpenseBonusAmt, na.rm= T),
        (baseSalary < 0 & !is.na(baseSalary)) |
          (earlySignonBonusAmt< 0 & !is.na(earlySignonBonusAmt)) |
          (signingstartingBonusAmt< 0 & !is.na(signingstartingBonusAmt)) |
          (gauranteedYeaarendBonusAmt< 0 & !is.na(gauranteedYeaarendBonusAmt)) |
          (tuitionReimbursementBonusAmt< 0 & !is.na(tuitionReimbursementBonusAmt)) |
          (relocationExpenseBonusAmt< 0 & !is.na(relocationExpenseBonusAmt)))
    , .(id, surveyInstance)];

jofDups <- jof[
  ][duplicated(jof[, .(organizationName, positionType, id)]) |
      duplicated(jof[, .(organizationName, positionType, id)], fromLast = T)
  ][, .SD[fixedSalary== max(fixedSalary)], id
  ][, .SD[decisionDate== max(decisionDate)], id
  ][, .SD[offerAnswerDate== max(offerAnswerDate)], id
  ][, .SD[surveyInstance== max(surveyInstance)], id];
  # iteratively filter out duplicates

jof <- rbind(jof, jofDups);

jof <- jof[
  ][, c("nPositions", "nEmployers"):=
      .(length(organizationName),
        length(unique(organizationName)))
    , id
  ][order(nPositions, id, decisionDate)
  ][, priorAccept:= positionType== "Permanent" & nPositions > nEmployers
  ][, .(id, organizationName, positionType, priorAccept,
        fixedSalary, negFixed, jobOfferStatus)];
  # nPositions > nEmployers when a student did an internship with an employer
  # from whom they eventually accepted a permanent position


### CLEAN GMATs ================================================================
```

```
ts <- ts[
  ][testCategory== "gmat_total"
  ][, .SD[gmat== max(gmat)]
    , id];

ts <- unique(ts);


### CLEAN JOB RANKINGS ============================================================
jra[, employer:= enc2native(employer)];
jra[grepl("Nestl", employer), employer:= "Nestle"];

jra <- rbind(jra[year==2000][, year:= 1997], jra[year==2000][, year:= 1998],
             jra[year==2000][, year:= 1999], jra);
  # Universum rankings not available before 2000, so extrapolate backwards


### MERGE DATA AND TIDY UP ========================================================
labor <- merge(gra, jof, "id", allow.cartesian= T);

labor <- labor[
  ][, numYears:= max(year) - min(year)
    , id
  ][(positionType== "Internship" & term== 4 & year== firstYr) |
    (positionType== "Permanent" & priorAccept== T & term== 4 & year== firstYr) |
    (positionType== "Permanent" & priorAccept== F & term== 3 & year== firstYr + 1)];
  # For internship positions and permanent positions where a prior internship
  # was completed, select GPA at end of first fall. For permanent positions,
  # select GPA at end of first spring.

labor <- merge(labor, ts, "id");

strip1 <- " Group| Company| Partners| Incorporated| Corporation| Inc| Inc\\.| & Company| & Co\\.| Co\\.| GmbH| Ltd\\.|"
strip2 <- " LLC| L\\.P\\.| Corp\\.| Plc| LLP| L\\.L\\.C\\.|\\.|,| Capital| Management| North America| Ltd| Consulting|"
strip3 <- " Financial Services|\\*| Corp| International| Financial|"
strip4 <- " Technologies|The | Investments| Securities| Holdings| Roebuck| HPC| Brands| Scientific| united states|"
strip5 <- " tohmatsu| benson| global| S\\.A\\.| ag| & touche";
```

```
toStrip <- paste0(strip1, strip2, strip3, strip4, strip5);

Encoding(jra$employer) <- 'latin1';
labor[, organizationNameMerge:= tolower(gsub(toStrip, "", organizationName, T))];
jra[, employerMerge:= tolower(gsub(toStrip, "", employer, T))];
  # Make the school data and job rank data organization names similar

matches <- data.frame(namesData= as.character(), namesJra= as.character(),
                      year= as.numeric());

for (i in 1998:2007) {
  miniJra <- jra[year== i];
  miniData <- labor[year== i];

  organizationNameMerge <- unique(miniData$organizationNameMerge);
  organizationNameMerge <- organizationNameMerge[order(organizationNameMerge)];

  employerMerge <- unique(miniJra$employerMerge);
  employerMerge <- employerMerge[order(employerMerge)];

  sd <- stringdistmatrix(employerMerge, organizationNameMerge);

  matchQuality <- 1 - apply(sd, 2, min)*1/nchar(organizationNameMerge);

  matchesMini <- cbind(organizationNameMerge,
                 employerMerge= employerMerge[apply(sd, 2, which.min)], year= i,
                 matchQuality);
  matches <- rbind(matches, matchesMini[matchQuality >= 0.8, c(1:3)]);
}
  # 0.8 always leads to a good match (hand-match the rest below)

matches <- data.table(matches)[, year:= as.numeric(year)];
  #--- Initial matches using string distance

matches[organizationNameMerge %in% c("ubs investment bank"), organizationNameMerge:= "ubs"];
matches <- matches[!grepl("deloitte", organizationNameMerge, T)];

jra <- jra[!(employer=="KPMG" & year== 2002) &
```

```
                !(grepl("Boston Scientific", employer) & year== 2004) &
                !(grepl("Boston Scientific", employer) & year== 2006)];
  # Ambiguous entries

matches <- rbind(matches,
  list("kraft foods", "kraft", 2001),
  list("ubs", "warburg dillon read/ubs", 1998),
  list("ubs", "warburg dillon read/ubs", 1999),
  list("ubs", "warburg dillon read/ubs", 2000),
  list("ubs", "ubs/warburg dillon read", 2001),
  list("ubs", "ubs/warburg", 2002),
  list("ubs", "ubs/warburg", 2003),
  list("ubs", "ubs", 2007),
  list("boston", "boston consutling", 2002),
  list("siebel systems", "siebel", 1999),
  list("siebel systems", "siebel", 2000),
  list("siebel systems", "siebel", 2001),
  list("putnam investment", "putnam", 1998),
  list("putnam investment", "putnam", 1999),
  list("putnam investment", "putnam", 2000),
  list("prtm consultants", "prtm", 2006),
  list("pg & e", "pacific gas & electric", 2001),
  list("pg & e", "pacific gas & electric", 2003),
  list("pg & e", "pacific gas & electric", 2004),
  list("nestle", "nestle", 1999),
  list("nestle", "nestle", 2000),
  list("morgan stanley", "morgan stanley dean witter", 1998),
  list("morgan stanley", "morgan stanley dean witter", 1999),
  list("morgan stanley", "morgan stanley dean witter", 2000),
  list("morgan stanley", "morgan stanley dean witter", 2001),
  list("mercer", "mercer mgmt", 1998),
  list("mercer", "mercer mgmt", 1999),
  list("international business machines", "ibm", 1998),
  list("international business machines", "ibm", 1999),
  list("international business machines", "ibm", 2000),
  list("international business machines", "ibm", 2001),
  list("international business machines", "ibm", 2002),
  list("international business machines", "ibm", 2003),
```

```
list("international business machines", "ibm", 2004),
list("houlihan lokey", "houlihan lockey howard & zukin", 1999),
list("harrah's entertainment", "harrahs", 2004),
list("ge solutions", "ge", 1998),
list("ford motor", "ford", 1998),
list("ford motor", "ford", 1999),
list("ford motor", "ford", 2000),
list("fidelity & research", "fidelity", 1999),
list("enron energy services operations", "enron", 1998),
list("enron energy services operations", "enron", 1999),
list("enron energy services operations", "enron", 2000),
list("eli lilly", "lilly", 2002),
list("eli lilly", "lilly (eli lilly)", 2003),
list("eli lilly", "lilly (eli lilly and)", 2004),
list("eli lilly", "lilly (eli lilly and)", 2005),
list("eli lilly", "lilly (eli lilly and)", 2006),
list("diamond & technology consultants ", "diamond technology", 1998),
list("diamond & technology consultants ", "diamond technology", 1999),
list("diamond & technology consultants ", "diamond technology", 2000),
list("diamond & technology consultants ", "diamond technology", 2001),
list("diamond & technology consultants ", "diamond technology", 2002),
list("diamond & technology consultants ", "diamondcluster", 2004),
list("dell", "dell computer", 1998),
list("dell", "dell computer", 1999),
list("dell", "dell computer", 2004),
list("dell", "dell computer", 2005),
list("deere", "john deere", 2002),
list("deere", "john deere", 2003),
list("credit suisse", "credit suisse first boston", 1998),
list("credit suisse", "credit suisse first boston", 1999),
list("credit suisse", "credit suisse first boston", 2000),
list("credit suisse", "credit suisse first boston", 2001),
list("credit suisse", "credit suisse first boston", 2002),
list("credit suisse", "credit suisse first boston", 2003),
list("credit suisse", "credit suisse first boston", 2004),
list("credit suisse", "credit suisse first boston", 2005),
list("credit suisse", "credit suisse first boston", 2006),
list("credit suisse ", "credit suisse first boston", 1998),
```

```
list("credit suisse ", "credit suisse first boston", 1999),
list("credit suisse ", "credit suisse first boston", 2000),
list("credit suisse ", "credit suisse first boston", 2001),
list("credit suisse ", "credit suisse first boston", 2002),
list("credit suisse ", "credit suisse first boston", 2003),
list("credit suisse ", "credit suisse first boston", 2004),
list("credit suisse ", "credit suisse first boston", 2005),
list("credit suisse ", "credit suisse first boston", 2006),
list("citigroup", "citicorp/citibank", 1998),
list("citigroup", "citicorp/citibank", 1999),
list("citigroup", "citicorp/citibank", 2000),
list("citigroup", "citigroup", 2004),
list("cap gemini", "gemini", 1998),
list("cap gemini", "gemini", 1999),
list("cap gemini", "gemini", 2000),
list("dcap gemini ernst & young", "cap gemini ernst & young", 2004),
list("cap gemini", "cap gemini ernst & young", 2004),
list("apple", "apple computer", 2006),
list("compaq computer", "compaq", 1999),
list("compaq computer", "compaq", 2000),
list("chevron", "chevrontexaco", 2004),
list("bank of america merrill lynch", "merrill lynch", 1998),
list("bank of america merrill lynch", "merrill lynch", 1999),
list("bank of america merrill lynch", "merrill lynch", 2000),
list("bank of america merrill lynch", "merrill lynch", 2001),
list("bank of america merrill lynch", "merrill lynch", 2002),
list("bank of america merrill lynch", "merrill lynch", 2003),
list("bank of america merrill lynch", "merrill lynch", 2004),
list("bank of america merrill lynch", "merrill lynch", 2005),
list("bank of america merrill lynch", "merrill lynch", 2006),
list("bp", "bp-amoco", 1998),
list("bp", "bp-amoco", 1999),
list("bp", "bp-amoco", 2000),
list("barclays", "barclay investors", 2000),
list("barclays", "barclays investors", 2001),
list("barclays", "barclays investors", 2002),
list("abbott laboratories", "abbott", 1999),
list("abn amro", "abn amro bank", 1998),
```

```
  list("abn amro", "abn amro bank", 1999),
  list("abn amro", "abn amro bank", 2000),
  list("cowen", "sg cowen", 1998),
  list("cowen", "sg cowen", 1999),
  list("cowen", "sg cowen", 2000),
  list("deloitte", "deloitte touche", 1998),
  list("deloitte", "deloitte touche", 1999),
  list("deloitte", "deloitte touche", 2000),
  list("deloitte", "deloitte", 2001),
  list("deloitte", "deloitte", 2002),
  list("deloitte", "deloitte (braxton)", 2003),
  list("deloitte", "deloitte", 2004),
  list("deloitte", "deloitte", 2005),
  list("deloitte", "deloitte", 2006),
  list("ibm business services", "ibm", 2001),
  list("ibm business services", "ibm", 2002),
  list("ibm business services", "ibm", 2003),
  list("ibm business services", "ibm", 2004),
  list("ibm services (us)", "ibm", 2005));

labor[organizationNameMerge %in% c("ubs asset", "ubs investment bank", "ubs warburg"),
      organizationNameMerge:= "ubs"];
labor[organizationNameMerge %in% c("citigroup markets", "citigroup smith barney equity research ", "citigroup"),
      organizationNameMerge:= "citigroup"];
labor[organizationNameMerge %in% c("nestle usa"), organizationNameMerge:= "nestle"];
labor[organizationNameMerge %in% c("barclays investors national association", "barclays retail bank"),
      organizationNameMerge:= "barclays"];
  #--- Hand-matching non-string distanced matched

labor <- merge(labor, matches, c("organizationNameMerge", "year"), all.x= T);
labor <- merge(labor, jra, c("employerMerge", "year"), all.x= T);

labor <- merge(labor, cpi, c("year"));

labor <- labor[
  ][, firstYr:= firstYr + 1
  ][, c("gnd", "fixedSalary", "rank"):=
      .(firstYr >= 2001,
```

```
        fixedSalary*min(cpi)/cpi/1000,
        -rank)
  ][, c("gndGpa", "gndGmat", "logFixedSalary",
        "pickSal", "pickRank"):=
      .(gnd*gpa,
        gnd*gmat,
        log(fixedSalary),
        firstYr %in% 1998:2006 & !negFixed & fixedSalary > 0
          & positionType== "Permanent" & legitN(gpa),
        firstYr %in% 1998:2006 & legitN(rank) & priorAccept== F)];

quants <- quantile(labor[pickSal==T]$fixedSalary, probs= c(0.005, 0.995));

labor <- labor[
  ][, pickSal:= pickSal== T & fixedSalary > quants[1] & fixedSalary < quants[2]
  ][, typeYear:= .GRP, c("positionType", "firstYr")
  ][firstTerm== 4 & nterms == 8 & firstYr %in% 1998:2006];
  # 8 terms, starting in fall is a regular student

labor[, organizationName:= trimws(organizationName, "r")];
inds[, organizationName:= trimws(organizationName, "r")];

labor <- merge(labor, inds, "organizationName", all.x= T);

labor <- labor[
  ][industry== "", industry:= NA
  ][!(industry %in% c("Investment Banking and Brokerage",
                      "Research and Consulting Services")) & !is.na(industry)
    , industry:= "Other"
  ][, indYear:= .GRP, c("industry", "firstYr")
  ][is.na(industry), indYear:= NA];

laborCopy <- copy(labor);

meanSal <- mean(labor[!is.na(industry)][pickSal==T ]$fixedSalary, na.rm= T);
meanGmat <- mean(labor[!is.na(industry)][pickSal==T | pickRank== T]$gmat, na.rm= T);
labor[, c("fixedSalary", "gmat"):=
        .(fixedSalary - meanSal,
```

```
          gmat - meanGmat)];

tabular((Heading("De-meaned \\emph{FIXED SALARY} ('000s)")*fixedSalary) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= labor[!is.na(industry)][pickSal== T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
De-meaned *FIXED SALARY* ('000s)	2285	$2.74e-16$	23.7	−45	−17	−1.3	17	64

```
tabular((Heading("\\emph{DESIRABILITY}")*(rank)) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= labor[!is.na(industry)][pickRank==T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
DESIRABILITY	3018	−37.9	41.4	−170	−51	−26	−7	−1

```
tabular((Heading("\\emph{GPA}")*gpa) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= labor[!is.na(industry)][pickSal==T | pickRank==T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
GPA	4200	3.33	0.433	2.2	3	3.3	3.7	4

```
tabular((Heading("De-meaned \\emph{GMAT}")*gmat) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= labor[!is.na(industry)][pickSal==T | pickRank==T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
De-meaned *GMAT*	4200	$-2.89e-14$	49.3	−130	−34	5.9	36	86

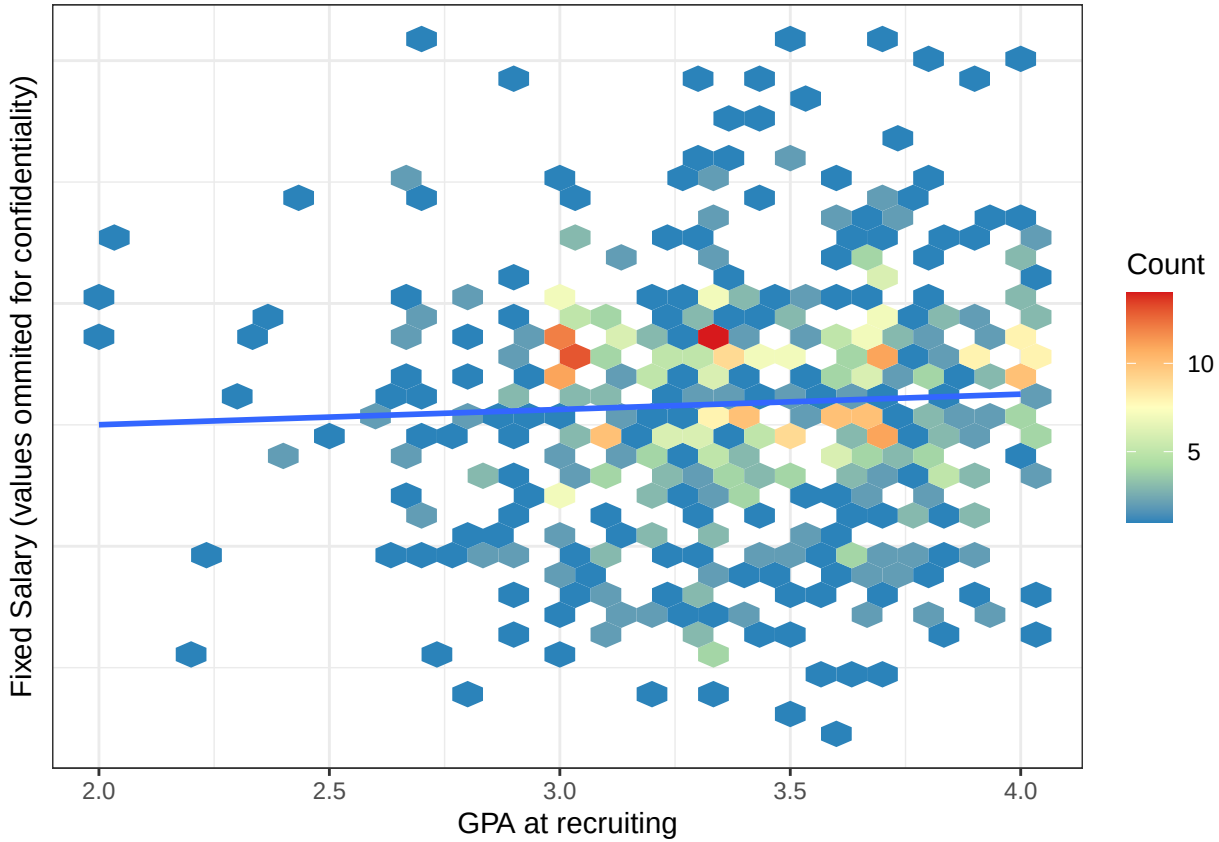
```
  ### TABLE 1 -----

labor <- copy(laborCopy);

reg1 <- felm(fixedSalary ~ gndGpa + gpa | industry + typeYear | 0 | organizationName, labor[pickSal== T]);
reg2 <- felm(fixedSalary ~ gndGpa + gpa + gndGmat + gmat | industry + typeYear | 0 | organizationName, labor[pickSal== T]);
reg3 <- felm(rank ~ gndGpa + gpa| industry + typeYear | 0 | organizationName, labor[pickRank== T]);
reg4 <- felm(rank ~ gndGpa + gpa + gndGmat + gmat | industry + typeYear | 0 | organizationName, labor[pickRank== T]);

stargazer(reg1, reg2, reg3, reg4, type= "text", omit= c("Constant"), omit.stat=c("rsq", "ser"), dep.var.labels.include = F);
```

```
##
## ================================================
##                        Dependent variable:
##                -----------------------------------
##                  (1)      (2)       (3)       (4)
## ------------------------------------------------
```

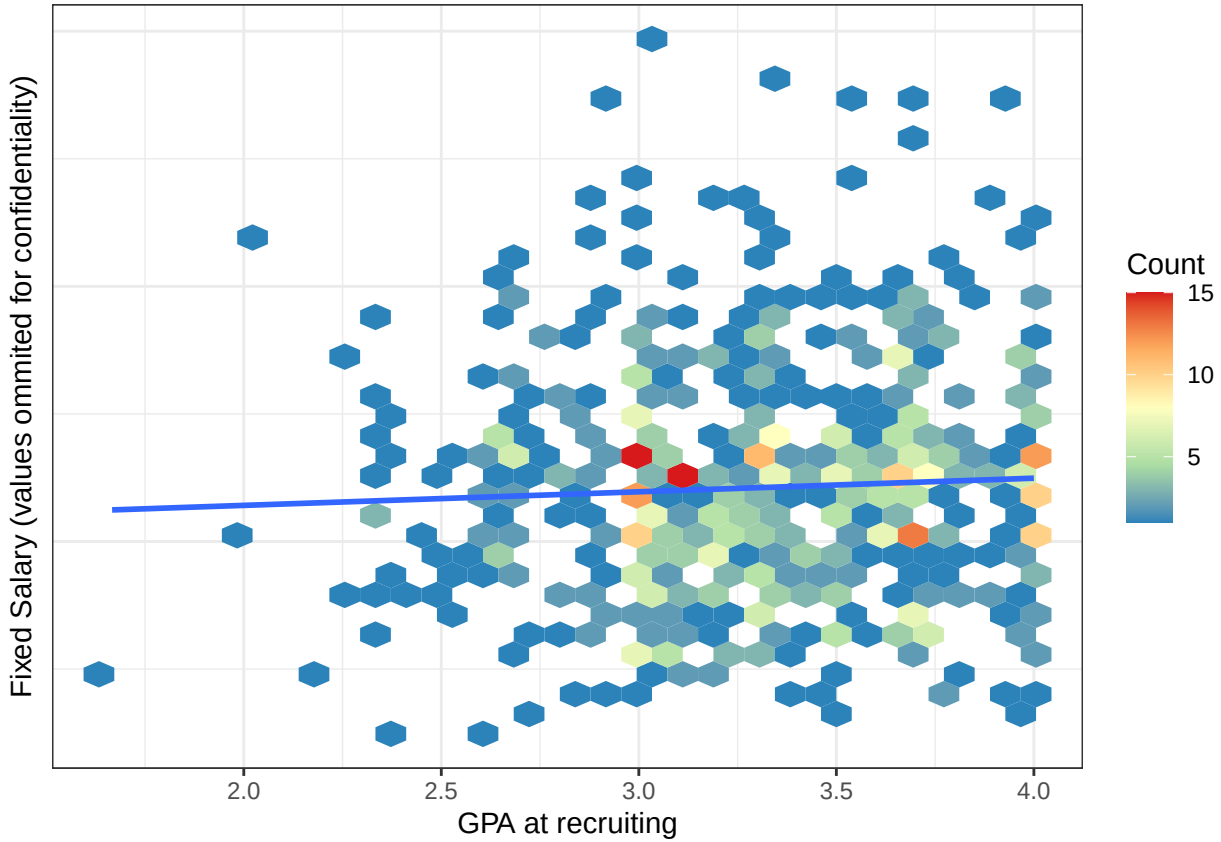
```
## gndGpa       -0.304  -1.221  -10.704** -10.037**
##              (2.190) (2.622)  (4.153)   (4.471)
##
## gpa           2.897  4.660**  9.647**  10.379**
##              (1.875) (2.115)  (4.188)   (4.227)
##
## gndGmat               0.013             -0.016
##                      (0.019)            (0.042)
##
## gmat                 -0.030*            -0.014
##                      (0.017)            (0.042)
##
## ------------------------------------------------
## Observations  2,285   2,285    3,018     3,018
## Adjusted R2   0.272   0.273    0.149     0.149
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
  ### TABLE 2 -----

labor <- copy(laborCopy);

###--- SALARIES
toUse <- c(1:length(labor[pickSal==T & gnd== T]$gpa));
numToUse <- length(labor[pickSal==T & gnd== F]$gpa);
toUse <- sample(toUse, numToUse)

rf <- colorRampPalette(rev(brewer.pal(9,"Spectral")));

salNoGnd <- ggplot(labor[pickSal==T & gnd== F], aes(gpa, fixedSalary)) + geom_hex() +
  scale_fill_gradientn(colours = rev(brewer.pal(5, "Spectral"))) +
  labs(x= "GPA at recruiting", y= "Fixed Salary (values ommited for confidentiality)") + theme_bw() +
  labs(fill = "Count") +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) +
  geom_smooth(method = "lm", se = FALSE);

salGnd <- ggplot(labor[pickSal==T & gnd== T][toUse], aes(gpa, fixedSalary)) + geom_hex() +
```

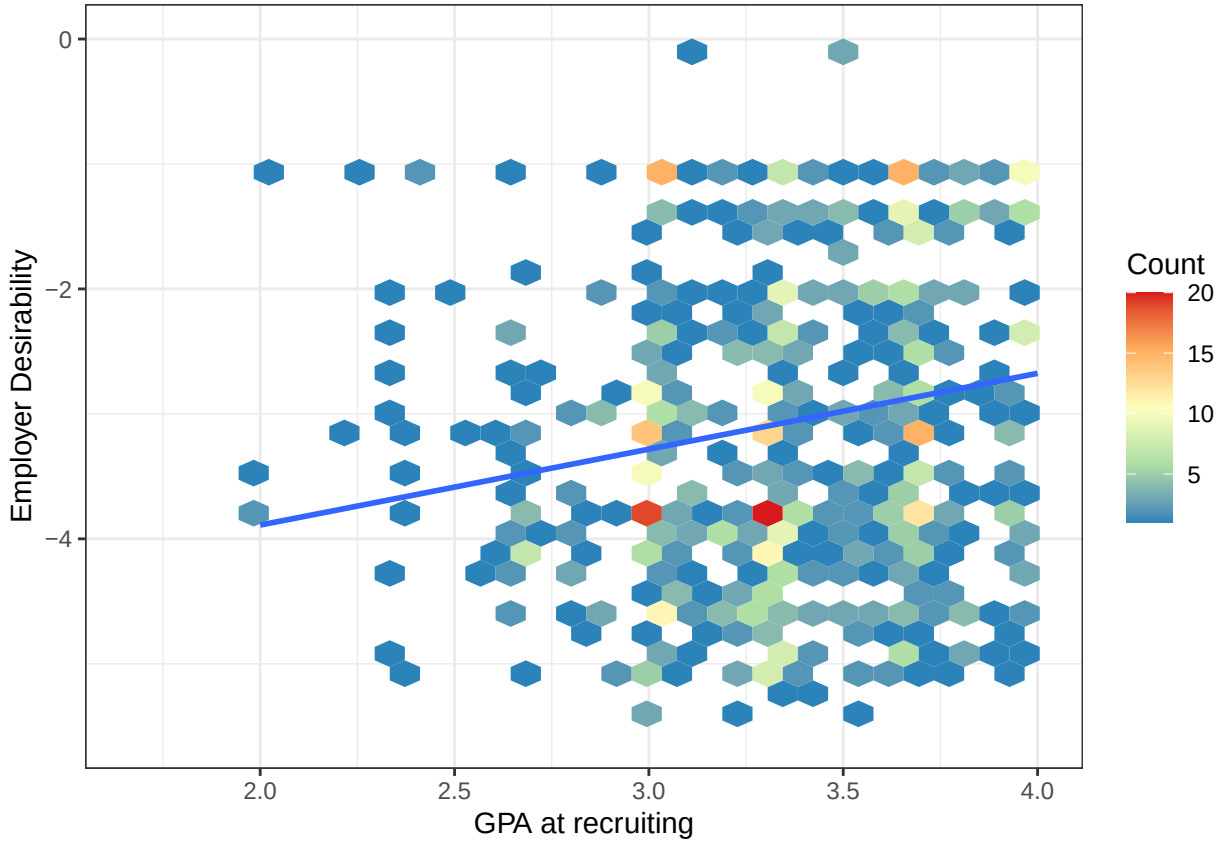
```
  scale_fill_gradientn(colours = rev(brewer.pal(5, "Spectral"))) +
  labs(x= "GPA at recruiting", y= "Fixed Salary (values ommited for confidentiality)") + theme_bw() +
  labs(fill = "Count") +
  theme(axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) +
  geom_smooth(method = "lm", se = FALSE);

labor <- copy(laborCopy);

labor[, rank2:= -log(-labor$rank)];

toUse <- c(1:length(labor[pickRank==T & gnd== T]$gpa));
numToUse <- length(labor[pickRank==T & gnd== F]$gpa);
toUse <- sample(toUse, numToUse);

rankNoGnd <- ggplot(labor[pickRank==T & gnd== F], aes(gpa, rank2)) + geom_hex() +
  scale_x_continuous(limits = range(labor[pickRank== T]$gpa)) +
  scale_y_continuous(limits = range(labor[pickRank== T]$rank2)) +
  scale_fill_gradientn(colours = rev(brewer.pal(5, "Spectral"))) +
  labs(x= "GPA at recruiting", y= "Employer Desirability") + theme_bw() +
  labs(fill = "Count") +
  geom_smooth(method = "lm", se = FALSE);

rankGnd <- ggplot(labor[pickRank==T & gnd== T][toUse], aes(gpa, rank2)) + geom_hex() +
  scale_x_continuous(limits = range(labor[pickRank== T]$gpa)) +
  scale_y_continuous(limits = range(labor[pickRank== T]$rank2)) +
  scale_fill_gradientn(colours = rev(brewer.pal(5, "Spectral"))) +
  labs(x= "GPA at recruiting", y= "Employer Desirability") + theme_bw() +
  labs(fill = "Count") +
  geom_smooth(method = "lm", se = FALSE);

salNoGnd;
```


Fixed Salary (values ommited for confidentiality)
2.0
2.5
3.0
3.5
4.0
GPA at recruiting
Count
10
5

```
salGnd;
```




```
  ### FIGURE 1 -----

rankNoGnd;
```


Employer Desirability
0
−2
−4
2.0
2.5
3.0
3.5
4.0
GPA at recruiting
Count
20
15
10
5

```
rankGnd;
```


```
  ### FIGURE 2 -----


### GPA EFFORT ========================================================================
effort <- data.table(read.csv("evaluations.csv"));

effort <- effort[
  ][, ayear:= ifelse(quarter== "Autumn", year + 1, year)
  ][, c("full", "part", "gnd", "post0001", "post0102", "core"):=
      .(type== "Full-time",
        type %in% c("Evening", "Weekend"),
        ayear>= 2001,
```

```
        ayear>= 2001,
        ayear>= 2002,
        title %in% c("Microeconomics", "Financial Accounting",
                     "Applied Regression Analysis", "Business Statistics"))];

effortCopy <- copy(effort);

effort <- effort[
  ][!(ins %in% c("", "Staff")) & (full + part== T)
  ][, c("fullGnd", "dFull1998", "dFull1999", "dFull2001", "dFull2002",
        "dFull2003", "dFull2004", "fullPost0001", "fullPost0102"):=
      .(full*gnd,
        full*(ayear== 1998),
        full*(ayear== 1999),
        full*(ayear== 2001),
        full*(ayear== 2002),
        full*(ayear== 2003),
        full*(ayear== 2004),
        full*post0001,
        full*post0102)
  ][, insSub:= .GRP
    , .(ins, sub)
  ][, insSubYear:= .GRP
    , .(ins, sub, ayear)
  ][, c("fullPre", "fullPost", "partPre", "partPost"):=
      .(max(full==T & gnd== F, na.rm= T),
        max(full==T & gnd== T, na.rm= T),
        max(full==F & gnd== F, na.rm= T),
        max(full==F & gnd== T, na.rm= T))
    , sub
  ][, balanced:= fullPre + fullPost + partPre + partPost== 4
  ][, pick:= legitN(hours) & legitN(enrolled) & ayear<= 2004];

data.frame(row.names = c("Full = 0", "Full = 1"),
  Instructors= c(length(unique(effort[pick== T & full== 0]$ins)),
         length(unique(effort[pick== T & full== 1]$ins))),
  Courses= c(length(unique(effort[pick== T & full== 0]$sub)),
         length(unique(effort[pick== T & full== 1]$sub))),
```

```
  InstructorCourses = c(length(unique(effort[pick== T & full== 0]$insSub)),
            length(unique(effort[pick== T & full== 1]$insSub))));
```

```
##          Instructors Courses InstructorCourses
## Full = 0         292     143               428
## Full = 1         244     134               365
```

```
  ### TABLE 3a -----

tabular((Heading("\\emph{HOURS}")*hours) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= effort[pick== T & full== F])
```

	N	Mean	SD	q5	q25	q50	q75	q95
HOURS	1723	5.23	1.73	2.9	4.1	4.8	5.9	11

```
tabular((Heading("\\emph{HOURS}")*hours) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= effort[pick== T & full== T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
HOURS	1353	5.29	2.92	2.5	3.8	4.6	5.8	18

```
tabular((Heading("\\emph{HOURS}")*hours) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= effort[pick== T & core== F])
```

	N	Mean	SD	q5	q25	q50	q75	q95
HOURS	2690	5.25	2.41	2.7	4	4.7	5.8	14

```
tabular((Heading("\\emph{HOURS}")*hours) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= effort[pick== T & core== T])
```

	N	Mean	SD	q5	q25	q50	q75	q95
HOURS	386	5.29	1.67	2.6	4.1	4.9	6.3	11

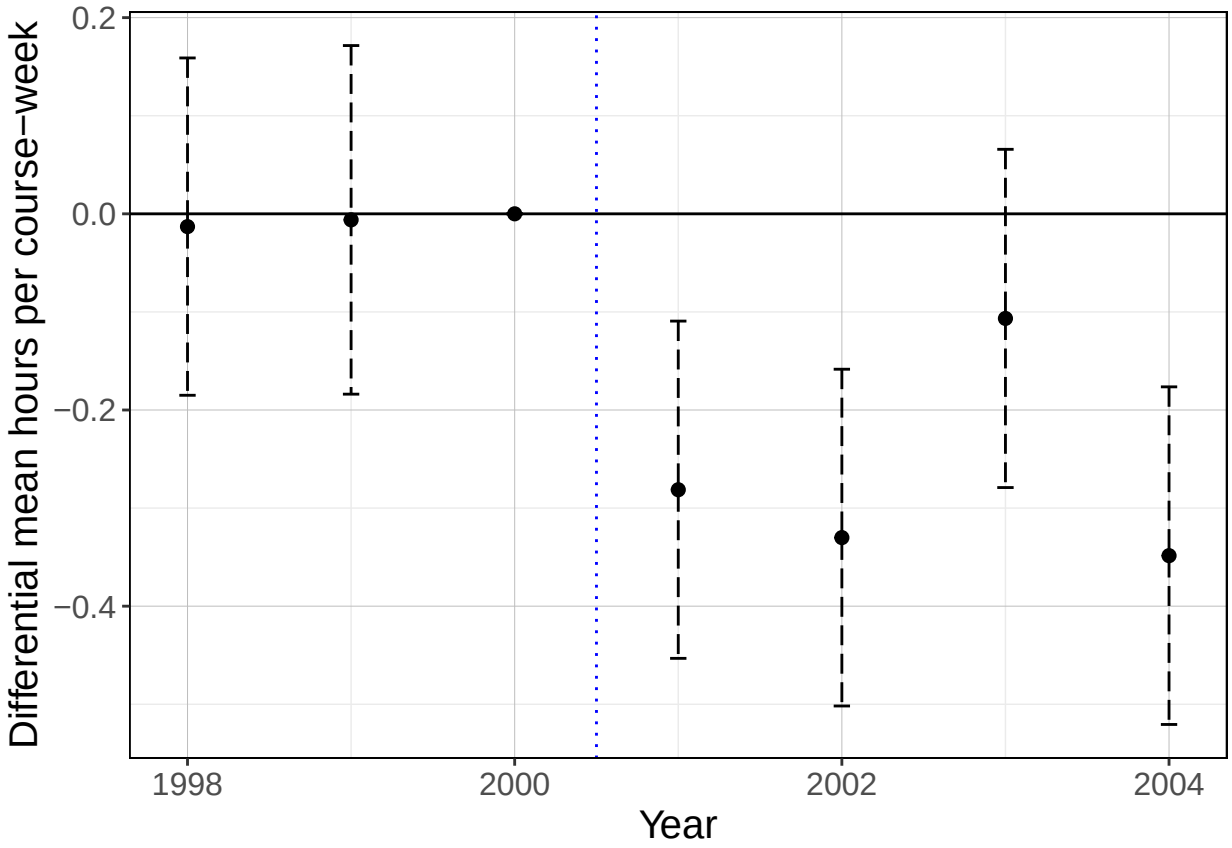
```
  ### TABLE 3b -----

fmla <- as.formula("hours ~ fullGnd + full + gnd | 0 | 0 | sub");
fmlaFE <- as.formula("hours ~ fullGnd + full + gnd | insSubYear | 0 | sub");

reg1 <- felm(fmla, effort[pick== T], weights= effort[pick== T]$enrolled);
reg2 <- felm(fmlaFE, effort[pick== T], weights= effort[pick== T]$enrolled);
reg3 <- felm(fmla, effort[pick== T & balanced== T], weights= effort[pick== T & balanced== T]$enrolled);
reg4 <- felm(fmlaFE, effort[pick== T & balanced== T], weights= effort[pick== T & balanced== T]$enrolled);
```

```
stargazer(reg1, reg2, reg3, reg4, type= "text", omit= c("Constant"), omit.stat=c("rsq", "ser"), dep.var.labels.include = F);
```

```
##
## ================================================
##                        Dependent variable:
##              ----------------------------------
##                (1)      (2)       (3)       (4)
## ------------------------------------------------
## fullGnd      -0.090  -0.261*** -0.236*** -0.286***
##              (0.120)  (0.088)   (0.082)   (0.084)
##
## full         -0.058  -0.268*** -0.185**  -0.260***
##              (0.103)  (0.077)   (0.087)   (0.075)
##
## gnd          -0.112             -0.178*
##              (0.094)  (0.000)   (0.092)   (0.000)
##
## ------------------------------------------------
## Observations  3,076    3,076     2,532     2,532
## Adjusted R2   0.002    0.886     0.019     0.857
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

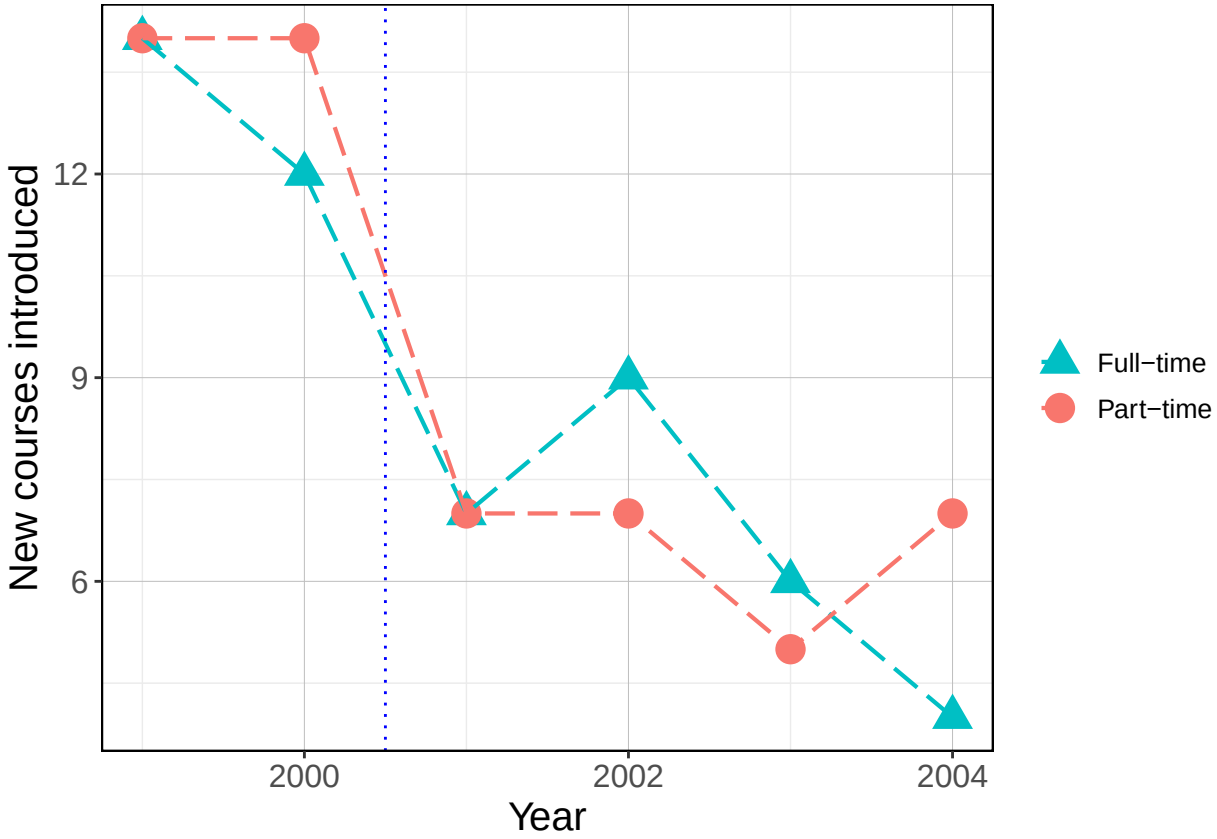
```
  ### TABLE 4 -----

dummies <- names(effort)[!grepl("dFull2000", names(effort)) &
                          (grepl("dp", names(effort)) | grepl("dFull", names(effort)))];

fmla <- as.formula(paste0("hours ~ ", paste(c(dummies, "full"), collapse= "+ "),
                          " | insSubYear | 0 | sub"));

reg <- felm(fmla, data= effort[pick== T & ayear<= 2004], weights= effort[pick== T & ayear<= 2004]$enrolled);

effortYearly <- data.table(coef= reg$coefficients, se= reg$se);
effortYearly <- effortYearly[!is.na(coef.hours)][-.N];
effortYearly <- cbind(
  year= c(min(effort[pick== T & ayear<= 2004]$ayear, na.rm= T):1999,
          2001:max(effort[pick== T & ayear<= 2004]$ayear, na.rm= T)), effortYearly);
effortYearly <- rbind(effortYearly, data.table(year= 2000, coef.hours= 0, se= 0));
```

```
effortYearly <- effortYearly[order(year)];

conf <- 1.645;
effortYearly$lower <- effortYearly$coef.hours - conf * effortYearly$se;
effortYearly$upper <- effortYearly$coef.hours + conf * effortYearly$se;

plot <- ggplot(effortYearly, aes(year)) +
  geom_point(aes(y= coef.hours), size= 2) +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.1,
                position=position_dodge(0.05), linetype= "longdash") +
  labs(x= "Year", y= "Differential mean hours per course-week") + theme_bw() +
  theme(axis.title= element_text(size=15), axis.text = element_text(size= 12),
        panel.background = element_rect(fill = 'white'),
        panel.grid.major = element_line(colour = "grey", size= 0.1),
        panel.border = element_rect(colour = "black", fill=NA, size= 0.5)) +
  scale_x_continuous(breaks = seq(min(effortYearly$year), max(effortYearly$year), by = 2)) +
  geom_hline(yintercept= 0) +
  geom_vline(aes(xintercept= 2000.5), linetype = "dotted", color="blue") # +

plot;
```




```
  ### FIGURE 3 -----

newSubs <- copy(effortCopy)[ayear<= 2004];

newSubs <- newSubs[
  ][order(sub, -full, ayear)
  ][enrolled!= 0 & (full + part== T)];

newSubs <- newSubs[, .SD[1], .(sub, full)];

newSubs[, newSubs:= .N, .(ayear, full)];
newSubs <- unique(newSubs, by= c("ayear", "full"));
```

```
newSubs[full== T, type:= "Full-time"];
newSubs[full== F, type:= "Part-time"];
newSubs[, type:= factor(as.character(type))];

plot <- ggplot(newSubs[ayear>= 1999], aes(y= newSubs, x= ayear, colour= type, shape= type)) +
  geom_point(aes(y= newSubs), size = 5) + geom_line(size= 0.75, linetype= c("longdash")) +
  labs(x= "Year", y= "New courses introduced") + theme_bw() +
  scale_color_manual(values=c("#00BFC4", "#F8766D")) + scale_shape_manual(values=c(17, 16)) +
  scale_x_continuous(breaks = seq(min(newSubs$ayear), max(newSubs$ayear), by = 2)) +
  theme(legend.title=element_blank(), legend.text = element_text(size=10),
        axis.title= element_text(size=15),
        axis.text = element_text(size= 12),
        panel.background = element_rect(fill = 'white'),
        panel.grid.major = element_line(colour = "grey", size= 0.1),
        panel.border = element_rect(colour = "black", fill=NA, size= 0.5)) +
  geom_vline(aes(xintercept= 2000.5), linetype = "dotted", color="blue") #+

plot;
```




```
  ### FIGURE 4 -----

effort[, c("pick1", "pick2", "pick3", "pick4"):=
         .(ayear %in% c(2000:2001) & core== T,
           ayear %in% c(2001:2002) & core== T,
           ayear %in% c(2000:2001) & core== F,
           ayear %in% c(2001:2002) & core== F)];

fmla0001 <- as.formula("hours ~ fullPost0001 + full + post0001 | insSubYear | 0 | insSub");
fmla0102 <- as.formula("hours ~ fullPost0102 + full + post0102 | insSubYear | 0 | insSub");

reg1 <- felm(fmla0001, effort[pick== T & pick1== T], weights= effort[pick== T & pick1== T]$enrolled);
```

```
reg2 <- felm(fmla0102, effort[pick== T & pick2== T], weights= effort[pick== T & pick2== T]$enrolled);
reg3 <- felm(fmla0001, effort[pick== T & pick3== T], weights= effort[pick== T & pick3== T]$enrolled);
reg4 <- felm(fmla0102, effort[pick== T & pick4== T], weights= effort[pick== T & pick4== T]$enrolled);

stargazer(reg1, reg2, reg3, reg4, type= "text", omit= c("Constant", "post0001", "post0102"),
          omit.stat=c("rsq", "ser"), dep.var.labels.include = F);
```

```
##
## ================================================
##                         Dependent variable:
##              ------------------------------------
##                (1)       (2)      (3)       (4)
## ------------------------------------------------
## fullPost0001 -0.658**             -0.182
##              (0.295)             (0.124)
##
## fullPost0102            0.125              -0.097
##                        (0.287)             (0.130)
##
## full         -0.568** -1.226*** -0.226** -0.408***
##              (0.241)   (0.301)  (0.097)   (0.087)
##
## ------------------------------------------------
## Observations   118       114      768       763
## Adjusted R2   0.826     0.816    0.907     0.913
## ================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

```
  ### TABLE 5 -----


### COURSE SELECTION ============================================================
pop <- copy(effortCopy);

pop <- pop[
  ][!is.na(enrolledFT) & ayear<= 2004
  ][, numEH:= sum(enrolled*hours, na.rm= T), sub
  ][, numE:= sum(enrolled, na.rm= T), sub
  ][, difficulty:= numEH/numE
```

```
  ][, difficulty2:= sum(enrolled*hours, na.rm= T)/sum(enrolled, na.rm= T)
      , sub
  ][, c("enrSubFTYrQtr", "enrSubPTYrQtr"):=
      .(sum(enrolledFT, na.rm= T),
        sum(enrolledEv + enrolledWE, na.rm= T))
    , .(ayear, sub)
  ][, c("enrFTYrQtr", "enrPTYrQtr"):=
      .(sum(enrolledFT, na.rm= T),
        sum(enrolledEv + enrolledWE, na.rm= T))
    , .(ayear)
  ][, c("popFT", "popPT"):=
      .(enrSubFTYrQtr/enrFTYrQtr*10e3,
        enrSubPTYrQtr/enrPTYrQtr*10e3)];

pop <- collap(pop, ~ .(ayear, sub), fmean,
        cols= c("popFT", "popPT", "difficulty", "field"));

pop <- melt(pop, c("ayear", "sub", "field", "difficulty"),
            variable.name= "full", value.name= "pop");

pop <- pop[
  ][pop> 0
  ][, c("full", "gnd", "logPop"):=
      .(ifelse(full== "popFT", T, F),
        ayear>= 2001,
        log(pop))
  ][, c("fullDifficulty", "gndDifficulty", "fullGndDifficulty", "pick"):=
      .(full*difficulty,
        gnd*difficulty,
        full*gnd*difficulty,
        legitN(logPop))];

tabular((Heading("\\emph{POPULARITY}")*pop + (Heading("\\emph{DIFFICULTY}")*difficulty)) ~
          ((N=length) + Mean + SD + q5 + q25 + q50 + q75 + q95), data= pop[legitN(pop) & legitN(difficulty)]);
```

	N	Mean	SD	q5	q25	q50	q75	q95
POPULARITY	1549	90.00	124.00	0.97	11.0	37.0	110.0	500
DIFFICULTY	1549	5.75	2.99	2.80	4.2	4.8	6.3	17

```
  ### TABLE 3c

fmla <- as.formula("pop ~ fullGndDifficulty + gndDifficulty + fullDifficulty + difficulty | 0 | 0 | sub");
fmlaFE <- as.formula("pop ~ fullGndDifficulty + gndDifficulty + fullDifficulty + difficulty | field | 0 | sub");
fmlaLog <- as.formula("logPop ~ fullGndDifficulty + gndDifficulty + fullDifficulty + difficulty | 0 | 0 | sub");
fmlaFELog <- as.formula("logPop ~ fullGndDifficulty + gndDifficulty + fullDifficulty + difficulty | field | 0 | sub");

reg1 <- felm(fmla, pop);
reg2 <- felm(fmlaFE, pop);
reg3 <- felm(fmlaLog, pop[pick==T]);
reg4 <- felm(fmlaFELog, pop[pick==T]);

stargazer(reg1, reg2, reg3, reg4, type= "text", omit= c("Constant"), omit.stat=c("rsq", "ser"), dep.var.labels.include = F);
```

```
##
## ============================================================
##                              Dependent variable:
##                     ---------------------------------------
##                        (1)        (2)        (3)        (4)
## ------------------------------------------------------------
## fullGndDifficulty    0.806     0.982*     0.037**    0.036**
##                     (0.505)    (0.584)    (0.019)    (0.018)
##
## gndDifficulty        0.524     1.587*      0.019      0.024
##                     (0.743)    (0.881)    (0.014)    (0.015)
##
## fullDifficulty      -1.290*    -0.959     -0.034*    -0.027
##                     (0.686)    (0.726)    (0.018)    (0.018)
##
## difficulty         -5.438*** -8.844*** -0.122*** -0.156***
##                     (1.978)    (2.823)    (0.031)    (0.036)
##
## ------------------------------------------------------------
## Observations         1,549      1,549      1,549      1,549
## Adjusted R2          0.015      0.101      0.042      0.142
## ============================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```

```
  ### TABLE 6 -----


### EXTRACURRICULAR ENGAGEMENT ===================================================
load("profs.rdata");

### CODE TO PRODUCE 'educ' AND 'emp'
### Un-comment code below by one layer only to preserve explanatory comments
# ids <- fread("id.csv");
# educ <- fread("education.csv");
# emp <- fread("employment.csv");
# group <- fread("groups.csv");
#
# ids <- ids[!duplicated(name)];  # Drop 30 profiles with duplicated names
#
# group <- group[
#   ][grepl("Columbia|Stern|NYU|New York University|CBS", groupName, T) &
#     grepl("Club|Network|Women|Association|Circle|Society|Organization|Athletics|Institute|Community|Council",
#           groupName, T) & !grepl("Alumni|class of", groupName, T)
#   ][, numGroup:= .N, id
#   ][, .(id, numGroup)];
# group <- unique(group, by= "id");
# # Consider groups linked to the Business School excluding "Class of" and "Alumni"
#
# educ <- merge(educ, group, "id", all.x= T);
# educ[is.na(numGroup), numGroup:= 0];
#
# educ <- educ[
#   ][id %in% ids$id & !duplicated(id) & educYearStart %in% c(2000:2018) & (educYearEnd - educYearStart) %in% c(2, 3) &
#     !grepl("Executive|EMBA|Exchange|GMBA|Part|Dual|Medicine|Hold|MPH|Pre|Equivalent|Immersion|JD|PGDBM|credits",
#       degree, T) & !grepl("courses|MFA|/MA|MIA|Boot|PhD|Certificate|IMBA|execMBA|MSCF|Tech MBA|BWL|Mumbai", degree, T) &
#     grepl("Columbia Business School|Columbia University|New York University|NYU", college, T) &
#     !grepl("TRIUM", college, T)
#   ][grepl("MBA|M.B.A", degree, T) | (grepl("Master|Masters", degree, T) & grepl("Business Administration", degree, T))
#   ][, act:= str_count(activitiesStrict, "@@") + str_count(activitiesLoose, "@@") + numGroup
#   ][, c("columbia", "nyu"):=
#     .(grepl("Columbia", college, T), grepl("New York University|NYU", college, T))
#   ][, "gnd":= (columbia== T & educYearStart >= 2011) | (nyu== T & educYearStart >= 2013)
```

```
#   ][, .(id, nyu, educYearStart, educYearEnd, act, gnd)];
# # Consider MBA degrees from Columbia Business School and NYU Stern that were
# # candidates' most recent degrees, had matriculation between 2000 and 2018,
# # and took two or three years to complete. "@@" is used to separate different
# # extracurriculars listed in activitiesStrict and activitiesLoose. "act" is
# # the count of extracurriculars.
#
# ids <- educ[, .(id, educYearStart, educYearEnd, gnd, nyu)];
#
# emp <- merge(emp, ids, "id");
# # For employment history, begin by considering the students in "educ" above
#
# emp <- emp[
#   ][, internship:= grepl("intern|summer", position, T)
#   ][, "employer":= as.character(.GRP), by= employer
#   ][, "position":= as.character(.GRP), by= position
#   ][, .(id, employer, empYearStart, empYearEnd, educYearStart, educYearEnd, gnd, nyu, internship)];
# # Recode employers and positions as number-strings for confidentiality
#
# rm(ids, group);

educ <- educ[
  ][, c("actExt", "logAct", "logActPlusOne"):=
      .(act > 0,
        log(act),
        log(1 + act))
  ][, clust:= .GRP
    , by= .(educYearStart, nyu)];
  # Produce extracurricular engagement variables

reg1 <- felm(actExt ~ gnd | educYearStart + nyu | 0 | clust, educ);
reg2 <- felm(act ~ gnd | educYearStart + nyu | 0 | clust, educ);
reg3 <- felm(logActPlusOne ~ gnd | educYearStart + nyu | 0 | clust, educ);

stargazer(reg1, reg2, reg3, type= "text", omit= c("Constant"), omit.stat=c("rsq", "ser"), dep.var.labels.include = F);
```

```
##
## ============================================
##                   Dependent variable:
```

```
##               ----------------------------
##                   (1)       (2)       (3)
## ------------------------------------------
## gnd            0.076**   0.324*   0.135***
##                (0.032)   (0.193)   (0.044)
##
## ------------------------------------------
## Observations   1,272     1,272     1,272
## Adjusted R2    0.045     0.025     0.039
## ==========================================
## Note:          *p<0.1; **p<0.05; ***p<0.01
```

```
  ### TABLE 7 -----


### EMPLOYMENT TURNOVER ============================================================
emp <- emp[
  ][!(empYearStart== educYearEnd - 1) & internship== F
  ][, sameStint:= (id==shift(id, 1, type= "lag") &
                     employer==shift(employer, 1, type= "lag"))
  ][1, sameStint:= F
  ][, startStint:= sameStint== F
  ][, c("sameStint", "empYearEnd"):=
      .(cumsum(startStint),
        as.numeric(empYearEnd))
  ][, empYearEnd:= ifelse(is.na(empYearEnd), 2020, empYearEnd)
  ][, c("empYearStart", "empYearEnd"):=
      .(min(empYearStart),
        max(empYearEnd))
    , sameStint];
  # Drop jobs likely to be internships, produce <sameStint> (which assigns a
  # unique number to each employee-employer spell, allowing for changes in
  # position within the spell). <empYearStart> and <empYearEnd> re-coded to
  # correspond to the respective spell.

emp <- unique(emp, by= "sameStint"); # Consider employee-employer spells

emp <- emp[
  ][!is.na(empYearStart)
```

```
  ][, c("tenureGTE1", "tenureGTE2", "tenureGTE3", "tenureGTE4", "tenureCen4"):=
      .(empYearEnd-empYearStart >= 1, empYearEnd-empYearStart >= 2, empYearEnd-empYearStart >= 3,
        empYearEnd-empYearStart >= 4, empYearEnd-empYearStart)
  ][(empYearStart == educYearEnd | empYearStart == educYearEnd - 1)
  ][tenureCen4 >= 4, tenureCen4:= 4
  ][, clust:= .GRP
    , by= .(educYearStart, nyu)];
  # Produce tenure variables, filter to first jobs after graduation

emp <- unique(emp, by= "id"); # Consider first listed job

reg1 <- felm(tenureGTE1 ~ gnd | educYearStart + nyu | 0 | clust, emp[educYearStart %in% c(2000:2016)]);
reg2 <- felm(tenureGTE2 ~ gnd | educYearStart + nyu | 0 | clust, emp[educYearStart %in% c(2000:2015)]);
reg3 <- felm(tenureGTE3 ~ gnd | educYearStart + nyu | 0 | clust, emp[educYearStart %in% c(2000:2014)]);
reg4 <- felm(tenureGTE4 ~ gnd | educYearStart + nyu | 0 | clust, emp[educYearStart %in% c(2000:2013)]);
reg5 <- felm(tenureCen4 ~ gnd | educYearStart + nyu | 0 | clust, emp[educYearStart %in% c(2000:2013)]);
  # The ending years are chosen to avoid censoring due to the data collection
  # point (summer 2020).

stargazer(reg1, reg2, reg3, reg4, reg5, type= "text", omit= c("Constant"), omit.stat=c("rsq", "ser"),
          dep.var.labels.include = F);
```

```
##
## ============================================================
##                            Dependent variable:
##              ----------------------------------------------
##                 (1)       (2)      (3)      (4)       (5)
## ------------------------------------------------------------
## gnd          -0.077*** -0.128***  0.029  -0.096*** -0.311***
##               (0.010)   (0.038)  (0.035)  (0.033)   (0.093)
##
## ------------------------------------------------------------
## Observations    601       547      456      404       404
## Adjusted R2    0.043     0.012   -0.003   -0.007    -0.008
## ============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
```

```
  ### TABLE 8 -----
```

```r
### SURVEY CORRELATION ========================================================
survey <- data.table(read.csv("survey.csv"));
survey <- survey[, c("sumiq", "sumeq"):=
               .((q1==6) + (q2==7) + (q3==2) + (q4==5) + (q5==5),
                 (q6>2) + (q10>2) + (q11>2) + (q12>2) + (q13>2))];
  # q14 is a participant's age

cor.test(survey$sumiq, survey$sumeq);
```

```
##
##  Pearson's product-moment correlation
##
## data:  survey$sumiq and survey$sumeq
## t = 3.097, df = 133, p-value = 0.002385
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0945406 0.4103315
## sample estimates:
##      cor
## 0.259355
```

```r
  ### IQ/EQ skill correlation
```