

The Assignment of Intellectual Property Rights and Innovation

```
##### WORKSPACE & PACKAGES #####
```

```
lapply(c("tidyverse", "lubridate", "fixest", "data.table"), library, character.only = TRUE)
```

```
summarize <- dplyr::summarize
```

```
select <- dplyr::select
```

```
rename <- dplyr::rename
```

```
##### FUNCTIONS #####
```

```
winsorize <- function(object, lower_percentile, upper_percentile, exclude = NULL){
```

```
  if (is.data.frame(object) == TRUE){
```

```
    reference_columns <- names(object)
```

```
    numeric_columns <- sapply(object, is.numeric)
```

```
    exclude_columns <- if(!is.null(exclude)) exclude %>% as.vector() %>% as.character() else vector() %>% as.character()
```

```
    numeric_subset <- object[ , numeric_columns]
```

```
    nonnumeric_subset <- cbind(object[ , !numeric_columns], object[ , exclude_columns])
```

```
    winsorized_variables <- numeric_subset %>% select(- one_of(exclude_columns))
```

```
    columns <- ncol(winsorized_variables)
```

```
    rows <- nrow(winsorized_variables)
```

```
    for (i in 1:columns){
```

```
      L <- (lower_percentile / 100)
```

```
      U <- (upper_percentile / 100)
```

```
      percentile <- as.matrix(quantile(winsorized_variables[[i]], c(L, U) , na.rm = T))
```

```
      min <- percentile[1,1]
```

```
      max <- percentile[2,1]
```

```
      for (j in 1:rows){
```

```
        winsorized_variables[j, i] <- if (is.na(winsorized_variables[j, i]) | is.infinite(winsorized_variables[j, i])) NA else
```

```
        if (winsorized_variables[j, i] < min) min else if (winsorized_variables[j, i] > max) max else winsorized_variables[j, i]
```

```
      }
```

```
    }
```

```
    final_output <- cbind(nonnumeric_subset, winsorized_variables)
```

```
    final_output <- final_output[, c(reference_columns, setdiff(names(final_output), reference_columns))]
```

```
    cat("\n", "In total,", columns, "variables were winsorized:", colnames(winsorized_variables))
```

```
    return(final_output)
```

```
  }
```

```
  else if (is.vector(object) == TRUE){
```

```

  L <- (lower_percentile / 100)
  U <- (upper_percentile / 100)
  percentile <- as.matrix(quantile(object, c(L, U) , na.rm = T))
  min <- percentile[1,1]
  max <- percentile[2,1]
  for (i in 1:length(object)){
    object[i] <- if (is.na(object[i]) | is.infinite(object[i])) NA else
      if (object[i] < min) min else if (object[i] > max) max else object[i]
  }
  return(object)
}
}

```

CREATE DATASET - STEP 1: PATENT-LEVEL DATA

USPTO DATA

<https://bulkdata.uspto.gov/>

<https://bulkdata.uspto.gov/data/patent/pair/economics/2014/>

<https://bulkdata.uspto.gov/data/patent/historical/2014/>

LOAD ALL APPLICATIONS

```

Applications <- fread("Data/application_data.csv.zip") %>%
  filter(!is.na(filing_date)) %>%
  filter(!is.na(patent_issue_date) | application_type == "PCT") %>%
  filter(!is.na(application_number)) %>%
  filter(!is.na(patent_number)) %>%
  filter(application_type != "?") %>%
  transmute(patent_number = patent_number %>% as.numeric,
             application_number = application_number,
             application_type = application_type %>% as.character,
             filing_date = lubridate::ymd(filing_date),
             issue_date = lubridate::ymd(patent_issue_date)) %>%
  arrange(patent_number); gc()

```

ADD OTHER DATES FROM MASTERFILE

```

Master <- fread("Data/historical_masterfile.zip") %>%
  mutate(patent_number = patent %>% as.numeric) %>%
  filter(patent_number > 0) %>%
  filter(!is.na(patent_number)) %>%
  transmute(patent_number = patent_number,
             class = uspc %>% as.character,
             sub_class = uspc_sub %>% as.character,
             priority_date = lubridate::dmy(prior_dt),
             publication_date = lubridate::dmy(pub_dt),
             disposition_date = lubridate::dmy(disposition_dt),
             expiration_date = lubridate::dmy(exp_dt)) %>%
  arrange(patent_number); gc()

```

DETERMINE WHETHER CHILD IS A CONTINUATION PATENT

```

Parents <- fread("Data/continuity_parents.csv.zip") %>%
  transmute(parent_application_number = parent_application_number %>% as.numeric,
             application_number = application_number %>% as.numeric,

```

```

      continuation_type = continuation_type %>% as.character) %>%
    filter(!is.na(application_number))

Children <- fread("Data/continuity_children.csv.zip") %>%
  transmute(parent_application_number = application_number %>% as.numeric,
            application_number = child_application_number %>% as.numeric,
            continuation_type = continuation_type %>% as.character) %>%
  filter(!is.na(application_number))

Families <- bind_rows(Parents,
                     Children) %>%
  distinct %>%
  mutate(continuation_patent = 1) %>% # this applies to the child (dropped later)
  group_by(application_number) %>%
  summarize(parent_application_number = paste(parent_application_number, collapse = ","),
            continuation_patent = continuation_patent[1],
            continuation_type = paste(unique(continuation_type), collapse = ",")) %>%
  ungroup(); rm(Parents, Children); gc()

# DETERMINE WHETHER PARENT HAS A CHILD WITH A PCT APPLICATION #
PCT <- bind_rows(fread("Data/continuity_parents.csv.zip") %>%
                 transmute(parent_application_number = parent_application_number %>% as.numeri
c,
                           child_application_number = application_number) %>% # keep as charac
ter
                           filter(grepl("PCT", child_application_number)), # keep parents with a PCT-chi
ld

                 fread("Data/continuity_children.csv.zip") %>%
                 transmute(parent_application_number = application_number %>% as.numeric,
                           child_application_number = child_application_number) %>% # keep as
character
                 filter(grepl("PCT", child_application_number))) %>% # keep parents with a PCT
-child
  na.omit() %>%
  distinct() %>%
  left_join(Applications %>%
            filter(application_type == "REGULAR") %>% # only regular parent patents
            transmute(parent_application_number = application_number %>% as.numeric,
                      parent_patent_number = patent_number %>% as.numeric,
                      parent_filing_date = filing_date) %>%
            na.omit() %>%
            distinct(),
            by = "parent_application_number") %>%
  left_join(Applications %>%
            select(child_application_number = application_number,
                  child_filing_date = filing_date) %>%
            na.omit() %>%
            distinct(),
            by = "child_application_number") %>%
  transmute(patent_number = parent_patent_number,
            parent_filing_date,

```

```

    subsequent_PCT_patent = 1,
    PCT_request_time = as.numeric(child_filing_date - parent_filing_date)) %>%
  filter(PCT_request_time >= 0) %>%
  arrange(patent_number) %>%
  na.omit()

# ADD FOREIGN PATENT STATUS #
Foreign <- fread("Data/foreign_priority.csv.zip") %>%
  transmute(application_number = application_number %>% as.numeric,
            foreign_parent_date = lubridate::ymd(foreign_parent_date),
            parent_country_code = parent_country_code %>% as.character,
            US = as.numeric(parent_country_code == "US")) %>%
  na.omit() %>%
  mutate(foreign_parallel_patent = 1) %>%
  group_by(application_number) %>%
  arrange(foreign_parent_date) %>%
  mutate(foreign_both = as.numeric(any(parent_country_code == "US") & any(parent_country_code !=
"US"))),
    first_foreign_parent_date = foreign_parent_date[1]) %>%
  ungroup() %>%
  mutate(foreign_parallel_patent.US_first = as.numeric(US == 1 & first_foreign_parent_date == fo
reign_parent_date)) %>%
  group_by(application_number) %>%
  arrange(foreign_parent_date) %>%
  summarize(foreign_parent_date = foreign_parent_date[1],
            foreign_parallel_patent = foreign_parallel_patent[1],
            foreign_parallel_patent.US_first = as.numeric(any(foreign_parallel_patent.US_first =
= 1))) %>%
  ungroup(); gc()

# LOAD TRANSACTIONS DATA FOR EVENT CODES #

# Event_codes <- fread("Data/event_codes.csv.zip")

Transactions <- fread("Data/transactions.csv.zip") %>%
  filter(event_code %in% c("EPRQ", "NPRQ")) %>% # keep only early and non-publication requests
  mutate(application_number = application_number %>% as.numeric) %>%
  filter(application_number %in% Applications$application_number)

Transactions_EPRQ <- Transactions %>%
  filter(event_code == "EPRQ") %>%
  transmute(application_number,
            early_publication_request = ifelse(event_code == "EPRQ", 1, 0),
            early_publication_request_date = lubridate::ymd(recorded_date)) %>%
  distinct()

Transactions_NPRQ <- Transactions %>%
  filter(event_code == "NPRQ") %>%
  transmute(application_number,
            non_publication_request = ifelse(event_code == "NPRQ", 1, 0),
            non_publication_request_date = lubridate::ymd(recorded_date)) %>%
  distinct(); rm(Transactions)

```

```

#### COMBINE ALL ####
Combined <- Applications %>%
  left_join(Master,
            by = "patent_number") %>%
  left_join(PCT,
            by = "patent_number") %>%
  mutate(application_number = application_number %>% as.numeric) %>%
  filter(!is.na(application_number)) %>% # drop PCT and reissues
  left_join(Families,
            by = "application_number") %>%
  left_join(Foreign,
            by = "application_number") %>%
  left_join(Transactions_EPRQ,
            by = "application_number") %>%
  left_join(Transactions_NPRQ,
            by = "application_number") %>%
  mutate(first_priority_date = pmin(priority_date, # earliest of U.S. versus foreign
                                    foreign_parent_date, na.rm = T)) %>%
  filter(first_priority_date >= "1990-01-01") %>% # to speed up
  mutate(continuation_patent = continuation_patent %>% replace_na(., 0),
         foreign_parallel_patent = foreign_parallel_patent %>% replace_na(., 0),
         foreign_parallel_patent.US_first = foreign_parallel_patent.US_first %>% replace_na(.,
0),
         subsequent_PCT_patent = subsequent_PCT_patent %>% replace_na(., 0),
         PCT_request_time = PCT_request_time %>% replace_na(., 0),
         early_publication_request = early_publication_request %>% replace_na(., 0),
         non_publication_request = non_publication_request %>% replace_na(., 0)) %>%
  arrange(patent_number) %>%
  distinct()

Applications <- Applications %>%
  mutate(application_number = application_number %>% as.numeric)

##### CREATE DATASET - STEP 2: ASSIGNEE-LEVEL DATA #####
### INVENTOR DATA ###

# https://www.mikewoeppel.com/data
# https://bulkdata.uspto.gov/data/patent/pair/economics/2014/

Inventors_rank <- fread("Data/all_inventors.csv.zip") %>%
  mutate(application_number = application_number %>% as.numeric) %>%
  left_join(Applications %>%
            select(application_number, patent_number),
            by = "application_number") %>%
  transmute(patent_number,
            inventor_name = inventor_name_last %>% toupper,
            inventor_rank = inventor_rank %>% as.numeric) %>%
  na.omit() %>%
  distinct

Inventors <- bind_rows(fread("Data/inventor_5m.tsv"),

```

```

      fread("Data/inventor_6m.tsv"),
      fread("Data/inventor_7m.tsv"),
      fread("Data/inventor_8m.tsv"),
      fread("Data/inventor_9m.tsv"),
      fread("Data/inventor_10m.tsv"),
      fread("Data/inventor_11m.tsv"),
      fread("Data/inventor_12m.tsv"),
      fread("Data/inventor_13m.tsv")) %>%
transmute(patent_number = patnum %>% as.numeric,
  inventor_id = inventor_id %>% as.character,
  inventor_name = last_name %>% toupper %>% as.character,
  inventor_country = country %>% toupper %>% as.character,
  inventor_state = state %>% toupper %>% as.character,
  inventor_city = city %>% toupper %>% as.character) %>%
na.omit() %>%
arrange(patent_number); gc()

### ASSIGNEE DATA ###

# https://www.mikewoeppel.com/data

Assignees <- bind_rows(fread("Data/assignee_5m.tsv"),
  fread("Data/assignee_6m.tsv"),
  fread("Data/assignee_7m.tsv"),
  fread("Data/assignee_8m.tsv"),
  fread("Data/assignee_9m.tsv"),
  fread("Data/assignee_10m.tsv"),
  fread("Data/assignee_11m.tsv"),
  fread("Data/assignee_12m.tsv")) %>%
filter(!is.na(type)) %>%
filter(type > 1) %>%
transmute(patent_number = patnum %>% as.numeric,
  assignee_id = assignee_id %>% as.character,
  assignee_name = organization %>% toupper %>% as.character,
  assignee_country = country %>% toupper %>% as.character,
  assignee_state = state %>% toupper %>% as.character,
  assignee_city = city %>% toupper %>% as.character,
  assignee_type = case_when(type == 2 ~ "U.S. Company",
    type == 3 ~ "Foreign Company",
    type == 4 ~ "U.S. Individual",
    type == 5 ~ "Foreign Individual",
    type == 6 ~ "U.S. Government",
    type == 7 ~ "Foreign Government",
    type == 8 ~ "U.S. University",
    type == 9 ~ "Foreign University",
    type == 10 ~ "U.S. Nonprofit Organization",
    type == 11 ~ "Foreign Nonprofit Organization",
    type == 12 ~ "U.S. Hospital",
    type == 13 ~ "Foreign Hospital",
    type == 14 ~ "U.S. Research Institute",
    type == 15 ~ "Foreign Research Institute",
    type == 16 ~ "U.S. Public Agency",

```

```

      type == 17 ~ "Foreign Public Agency",
      type == 18 ~ "U.S. Cooperative",
      type == 19 ~ "Foreign Cooperative") %>%
    toupper %>% as.character) %>%
  na.omit %>%
  arrange(patent_number); gc()

### FIRM DATA ###

# https://www.mikewoeppel.com/data

Firms <- fread("Data/patnum_permco_1976_2023.tsv") %>%
  transmute(patent_number = patnum %>% as.numeric,
            permco = permco %>% as.numeric) %>%
  distinct(patent_number, .keep_all = T)

##### CREATE DATASET - STEP 3: ADD SUPERSTARS AND TENURE #####
### SUPERSTAR INVENTORS ###
Superstars <- Inventors %>%
  select(patent_number, inventor_id) %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  mutate(priority_year = lubridate::year(first_priority_date)) %>%
  filter(!is.na(priority_year)) %>%
  select(inventor_id, priority_year) %>%
  group_by(inventor_id) %>%
  mutate(rolling_n_patents = sapply(priority_year, function(year) {
    sum(priority_year >= (year - 10) & priority_year <= year) })) %>%
  ungroup() %>%
  group_by(priority_year) %>%
  mutate(rolling_n_patents.p90 = quantile(rolling_n_patents, 0.90),
        rolling_n_patents.p95 = quantile(rolling_n_patents, 0.95),
        rolling_n_patents.p99 = quantile(rolling_n_patents, 0.99)) %>%
  ungroup() %>%
  transmute(inventor_id, priority_year,
            superstar90 = as.numeric(rolling_n_patents >= rolling_n_patents.p90),
            superstar95 = as.numeric(rolling_n_patents >= rolling_n_patents.p95),
            superstar99 = as.numeric(rolling_n_patents >= rolling_n_patents.p99)) %>%
  distinct; gc()

### FIRM-SPECIFIC INVENTOR TENURE ###
Tenure <- Inventors %>%
  select(patent_number, inventor_id) %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  left_join(Assignees %>%
    select(patent_number, assignee_id),
    by = "patent_number") %>%
  left_join(Firms %>%

```

```

      select(patent_number, permco),
      by = "patent_number") %>%
mutate(permco = ifelse(!is.na(permco), permco, assignee_id)) %>%
group_by(permco, inventor_id) %>%
arrange(first_priority_date) %>%
summarize(first_patent = first_priority_date[1]) %>%
ungroup(); gc()

### OVERALL INVENTOR TENURE ###
Tenure_overall <- Inventors %>%
  select(patent_number, inventor_id) %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  group_by(inventor_id) %>%
  arrange(first_priority_date) %>%
  summarize(first_patent_ever = first_priority_date[1]) %>%
  ungroup(); gc()

##### CREATE DATASET - STEP 4: ADD INVENTOR MOVES #####
### INVENTOR MOVES ANYWHERE ###
Moves.all <- Inventors %>%
  select(patent_number, inventor_id) %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  left_join(Assignees %>%
    select(patent_number, assignee_id),
    by = "patent_number") %>%
  left_join(Firms %>%
    select(patent_number, permco),
    by = "patent_number") %>%
  transmute(inventor_id,
    permco = ifelse(!is.na(permco), permco, assignee_id),
    priority_year = lubridate::year(first_priority_date)) %>%
  distinct()

Moves.all <- Moves.all %>%
  left_join(Moves.all %>%
    mutate(priority_year = priority_year - 1) %>%
    rename(permco_future1 = permco) %>%
    distinct(),
    by = c("inventor_id", "priority_year")) %>%
  mutate(changes_employer1 = ifelse(!is.na(permco_future1) & permco != permco_future1, 1, 0)) %
>%
  select(-permco_future1) %>%
  distinct() %>%
  left_join(Moves.all %>%
    mutate(priority_year = priority_year - 2) %>%
    rename(permco_future2 = permco) %>%

```

```

      distinct(),
      by = c("inventor_id", "priority_year")) %>%
mutate(changes_employer2 = ifelse(!is.na(permco_future2) & permco != permco_future2, 1, 0)) %
>%
select(-permco_future2) %>%
distinct() %>%
left_join(Moves.all %>%
  mutate(priority_year = priority_year - 3) %>%
  rename(permco_future3 = permco) %>%
  distinct(),
  by = c("inventor_id", "priority_year")) %>%
mutate(changes_employer3 = ifelse(!is.na(permco_future3) & permco != permco_future3, 1, 0)) %
>%
select(-permco_future3) %>%
distinct() %>%
left_join(Moves.all %>%
  mutate(priority_year = priority_year - 4) %>%
  rename(permco_future4 = permco) %>%
  distinct(),
  by = c("inventor_id", "priority_year")) %>%
mutate(changes_employer4 = ifelse(!is.na(permco_future4) & permco != permco_future4, 1, 0)) %
>%
select(-permco_future4) %>%
distinct() %>%
left_join(Moves.all %>%
  mutate(priority_year = priority_year - 5) %>%
  rename(permco_future5 = permco) %>%
  distinct(),
  by = c("inventor_id", "priority_year")) %>%
mutate(changes_employer5 = ifelse(!is.na(permco_future5) & permco != permco_future5, 1, 0)) %
>%
select(-permco_future5) %>%
distinct() %>%
mutate(changes_employer = changes_employer1 + changes_employer2 + changes_employer3 + changes_
employer4 + changes_employer5) %>%
group_by(inventor_id, priority_year) %>%
summarize(changes_employer = sum(changes_employer, na.rm = T)) %>%
ungroup() %>%
mutate(changes_employer = as.numeric(changes_employer > 0)) %>%
arrange(inventor_id, priority_year)

### INVENTOR MOVES TO CONTROL STATES ###
Moves.to_control <- Inventors %>%
  select(patent_number, inventor_id, inventor_country, inventor_state) %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  transmute(inventor_id,
    inventor_state,
    inventor_country,
    control_state = NA,

```

```

      priority_year = lubridate::year(first_priority_date)) %>%
distinct() %>%
filter(inventor_country == "US") %>%
filter(!is.na(inventor_state)) %>%
filter(inventor_state != "TX") %>%
transmute(inventor_id, priority_year,
           control_state = ifelse(inventor_state %in% c("CA", "DE", "IL", "KS", "MN", "NC", "NV", "UT", "WA"), 1, 0),
           treat_state = ifelse(!inventor_state %in% c("CA", "DE", "IL", "KS", "MN", "NC", "NV", "UT", "WA"), 1, 0)) %>%
distinct()

Moves.to_control <- Moves.to_control %>%
  left_join(Moves.to_control %>%
            mutate(priority_year = priority_year - 1) %>%
            rename(control_state_future1 = control_state,
                   treat_state_future1 = treat_state) %>%
            distinct(),
            by = c("inventor_id", "priority_year")) %>%
  mutate(changes_to_control1 = ifelse(!is.na(control_state_future1) & treat_state == 1 & control_state_future1 == 1, 1, 0),
         changes_to_treat1 = ifelse(!is.na(treat_state_future1) & control_state == 1 & treat_state_future1 == 1, 1, 0)) %>%
  select(-c(control_state_future1, treat_state_future1)) %>%
  distinct() %>%
  left_join(Moves.to_control %>%
            mutate(priority_year = priority_year - 2) %>%
            rename(control_state_future2 = control_state,
                   treat_state_future2 = treat_state) %>%
            distinct(),
            by = c("inventor_id", "priority_year")) %>%
  mutate(changes_to_control2 = ifelse(!is.na(control_state_future2) & treat_state == 1 & control_state_future2 == 1, 1, 0),
         changes_to_treat2 = ifelse(!is.na(treat_state_future2) & control_state == 1 & treat_state_future2 == 1, 1, 0)) %>%
  select(-c(control_state_future2, treat_state_future2)) %>%
  distinct() %>%
  left_join(Moves.to_control %>%
            mutate(priority_year = priority_year - 3) %>%
            rename(control_state_future3 = control_state,
                   treat_state_future3 = treat_state) %>%
            distinct(),
            by = c("inventor_id", "priority_year")) %>%
  mutate(changes_to_control3 = ifelse(!is.na(control_state_future3) & treat_state == 1 & control_state_future3 == 1, 1, 0),
         changes_to_treat3 = ifelse(!is.na(treat_state_future3) & control_state == 1 & treat_state_future3 == 1, 1, 0)) %>%
  select(-c(control_state_future3, treat_state_future3)) %>%
  distinct() %>%
  left_join(Moves.to_control %>%
            mutate(priority_year = priority_year - 4) %>%
            rename(control_state_future4 = control_state,

```

```

      treat_state_future4 = treat_state) %>%
    distinct(),
    by = c("inventor_id", "priority_year")) %>%
    mutate(changes_to_control4 = ifelse(!is.na(control_state_future4) & treat_state == 1 & control
_state_future4 == 1, 1, 0),
      changes_to_treat4 = ifelse(!is.na(treat_state_future4) & control_state == 1 & treat_sta
te_future4 == 1, 1, 0)) %>%
    select(-c(control_state_future4, treat_state_future4)) %>%
    distinct() %>%
    left_join(Moves.to_control %>%
      mutate(priority_year = priority_year - 5) %>%
      rename(control_state_future5 = control_state,
        treat_state_future5 = treat_state) %>%
      distinct(),
      by = c("inventor_id", "priority_year")) %>%
    mutate(changes_to_control5 = ifelse(!is.na(control_state_future5) & treat_state == 1 & control
_state_future5 == 1, 1, 0),
      changes_to_treat5 = ifelse(!is.na(treat_state_future5) & control_state == 1 & treat_sta
te_future5 == 1, 1, 0)) %>%
    select(-c(control_state_future5, treat_state_future5)) %>%
    distinct() %>%
    mutate(changes_to_control = changes_to_control1 + changes_to_control2 + changes_to_control3 +
changes_to_control4 + changes_to_control5,
      changes_to_treat = changes_to_treat1 + changes_to_treat2 + changes_to_treat3 + changes_
to_treat4 + changes_to_treat5) %>%
    group_by(inventor_id, priority_year) %>%
    summarize(changes_to_control = sum(changes_to_control, na.rm = T),
      changes_to_treat = sum(changes_to_treat, na.rm = T)) %>%
    ungroup() %>%
    mutate(changes_to_control = as.numeric(changes_to_control > 0),
      changes_to_treat = as.numeric(changes_to_treat > 0)) %>%
    arrange(inventor_id, priority_year)

```

CREATE DATASET - STEP 5: COMBINE ALL DATA FOR BASELINE DATASETS

```
treatment_date <- ymd("2005-01-01")
```

```
Data <- Combined %>%
```

```

  filter(filing_date >= "2003-01-01" & filing_date <= "2006-12-31") %>% # [-2, +2]
  filter(continuation_patent == 0) %>% # drop continuation patents
  filter(foreign_parallel_patent == 0 | foreign_parallel_patent.US_first == 1) %>% # drop foreign
n patents, unless US first
  filter(first_priority_date >= filing_date) %>% # keep only earliest first date
  left_join(Assignees,
    by = "patent_number") %>%
  filter(grepl("U.S.", assignee_type) | is.na(assignee_id)) %>% # only keep U.S. assignees or in
dividuals
  left_join(Inventors,
    by = "patent_number") %>%
  left_join(Inventors_rank,
    by = c("patent_number", "inventor_name")) %>%
  filter(inventor_rank > 0) %>% # keep all inventors
  filter(inventor_country == "US") %>% # only keep U.S. inventors

```

```

mutate(patent_inventor_rank = paste0(patent_number, inventor_rank)) %>%
distinct(patent_inventor_rank, .keep_all = T) %>%
mutate(patent_inventor_id = paste0(patent_number, inventor_id)) %>%
distinct(patent_inventor_id, .keep_all = T) %>%
select(-patent_inventor_id) %>%
left_join(Inventors %>%
  group_by(patent_number) %>%
  summarize(n_inventors = n()) %>%
  ungroup(),
  by = "patent_number") %>%
left_join(Assignees %>%
  mutate(individual_patent = as.numeric(grepl("INDIVIDUAL", assignee_type))) %>%
  group_by(patent_number) %>%
  summarize(individual_patent = mean(individual_patent)) %>%
  ungroup(),
  by = "patent_number") %>%
mutate(individual_patent = ifelse(!is.na(assignee_id), individual_patent, 1)) %>%
mutate(priority_year = lubridate::year(first_priority_date),
  filing_year = lubridate::year(filing_date),
  post = ifelse(filing_date >= treatment_date, 1, 0),
  treat = ifelse(!inventor_state %in% c("CA", "DE", "IL", "KS", "MN", "NC", "NV", "UT",
"WA"), 1, 0),
  treat2 = ifelse(inventor_state == "TX", NA, treat),
  AvB = post * treat,
  AvB2 = post * treat2,
  publication_date = ifelse(!is.na(publication_date), publication_date, issue_date),
  publication_date = lubridate::as_date(publication_date),
  disclosure_days = (publication_date - filing_date) %>% as.numeric,
  disclosure_deadline = ifelse(subsequent_PCT_patent == 1, 547, as.numeric(issue_date - f
iling_date)),
  disclosure_deadline = ifelse(disclosure_days %in% 546:556, disclosure_days, disclosure_
deadline), # correct the "actual" 18 months
  disclosure.perc = (disclosure_days / disclosure_deadline),
  disclosure.perc = case_when(disclosure.perc < 0 ~ 0,
    disclosure.perc > 1 ~ 1,
    .default = disclosure.perc),
  PCT_request_time = PCT_request_time %>% replace_na(., 0),
  early_publication_request = early_publication_request %>% replace_na(., 0),
  early_publication_request = case_when(is.na(early_publication_request_date) ~ 0,
    early_publication_request_date >= first_priority_
date & early_publication_request_date <= publication_date ~ 1,
    .default = 0),
  early_publication_request_time = as.numeric(early_publication_request_date - filing_dat
e) %>% replace_na(., 0),
  non_publication_request_time = as.numeric(non_publication_request_date - filing_date) %
>% replace_na(., 0)) %>%
left_join(Firms,
  by = "patent_number") %>%
mutate(public_firm = ifelse(!is.na(permco), 1, 0),
  permco = ifelse(!is.na(permco), permco, assignee_id),
  permco = ifelse(!is.na(permco), permco, inventor_id)) %>%
left_join(Superstars %>%

```

```

      mutate(priority_year = priority_year + 1), # superstar, as per the prior year
      by = c("inventor_id", "priority_year")) %>%
left_join(Tenure,
      by = c("permco", "inventor_id")) %>%
left_join(Tenure_overall,
      by = c("inventor_id")) %>%
mutate(firm = permco,
      firm_year = paste(permco, priority_year, sep = "_") %>% as.character,
      firm_inventor_id = paste(permco, inventor_id, sep = "_") %>% as.character,
      assignee_year = paste(assignee_id, priority_year, sep = "_") %>% as.character,
      assignee_inventor_id = paste(assignee_id, inventor_id, sep = "_") %>% as.character,
      hqstate_year = paste(assignee_state, priority_year, sep = "_") %>% as.character,
      class_year = paste(class, priority_year, sep = "_") %>% as.character,
      subclass_year = paste(sub_class, priority_year, sep = "_") %>% as.character,
      same_state = as.numeric(assignee_state == inventor_state) %>% replace_na(., 1),
      superstar90 = superstar90 %>% replace_na(., 0),
      superstar95 = superstar95 %>% replace_na(., 0),
      superstar99 = superstar99 %>% replace_na(., 0),
      tenure = as.numeric(1 + priority_year - lubridate::year(first_patent)),
      tenure = tenure %>% winsorize(., 1, 99),
      overall_tenure = as.numeric(1 + priority_year - lubridate::year(first_patent_ever)),
      overall_tenure = overall_tenure %>% winsorize(., 1, 99),
      individual_patent = individual_patent * 100,
      early_publication_request = early_publication_request * 100,
      disclosure.perc = disclosure.perc * 100) %>%
group_by(inventor_id, priority_year) %>%
mutate(n_patents = n()) %>%
ungroup() %>%
filter(!is.na(class))

##### CREATE DATASET - STEP 6: COMBINE ALL DATA FOR AGGLOMERATION DATASETS #####
### CREATE DATASET FOR AGGLOMERATION ANALYSES ###
Data.agglomeration <- Inventors %>%
  select(patent_number, inventor_id) %>%
  left_join(Combined %>%
    select(patent_number, priority_date, foreign_parent_date) %>%
    mutate(first_priority_date = pmin(priority_date, # earliest of U.S. versus foreign
      foreign_parent_date, na.rm = T)),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  left_join(Assignees %>%
    select(patent_number, assignee_id),
    by = "patent_number") %>%
  left_join(Firms %>%
    select(patent_number, permco),
    by = "patent_number") %>%
  transmute(patent_number, inventor_id,
    permco = ifelse(!is.na(permco), permco, assignee_id),
    priority_year = lubridate::year(first_priority_date)) %>%
  na.omit

Data.agglomeration <- Data.agglomeration %>%

```

```

left_join(Data.agglomeration %>%
  select(inventor_id, permco, priority_year) %>%
  distinct() %>%
  mutate(priority_year = priority_year + 1,
    prior_match1 = 1),
  by = c("inventor_id", "permco", "priority_year")) %>%
left_join(Data.agglomeration %>%
  select(inventor_id, permco, priority_year) %>%
  distinct() %>%
  mutate(priority_year = priority_year + 2,
    prior_match2 = 1),
  by = c("inventor_id", "permco", "priority_year")) %>%
left_join(Data.agglomeration %>%
  select(inventor_id, permco, priority_year) %>%
  distinct() %>%
  mutate(priority_year = priority_year + 3,
    prior_match3 = 1),
  by = c("inventor_id", "permco", "priority_year")) %>%
left_join(Data.agglomeration %>%
  select(inventor_id, permco, priority_year) %>%
  distinct() %>%
  mutate(priority_year = priority_year + 4,
    prior_match4 = 1),
  by = c("inventor_id", "permco", "priority_year")) %>%
left_join(Data.agglomeration %>%
  select(inventor_id, permco, priority_year) %>%
  distinct() %>%
  mutate(priority_year = priority_year + 5,
    prior_match5 = 1),
  by = c("inventor_id", "permco", "priority_year")) %>%
mutate(prior_match1 = prior_match1 %>% replace_na(., 0),
  prior_match2 = prior_match2 %>% replace_na(., 0),
  prior_match3 = prior_match3 %>% replace_na(., 0),
  prior_match4 = prior_match4 %>% replace_na(., 0),
  prior_match5 = prior_match5 %>% replace_na(., 0)) %>%
mutate(prior_match = prior_match1 + prior_match2 + prior_match3 + prior_match4 + prior_match5)
%>%
transmute(patent_number, inventor_id,
  new_hire = ifelse(prior_match == 0, 1, 0))

Data.agglomeration <- Data %>%
  as_tibble %>%
  left_join(Data.agglomeration,
    by = c("patent_number", "inventor_id")) %>%
  filter(new_hire == 1) %>%
  distinct() %>%
  mutate(new_state = inventor_state)

Data.agglomeration <- Data.agglomeration[rep(row.names(Data.agglomeration), each = 50), ] # crea
te by-state dataset

Data.agglomeration <- Data.agglomeration %>%

```

```

group_by(patent_number, application_number, inventor_id, permco) %>%
mutate(state_number = 1:50) %>%
ungroup() %>%
left_join(tibble(state_number = 1:50,
                  statecode = state.abb),
          by = "state_number") %>%
select(-state_number) %>%
mutate(firmchoice = as.numeric(new_state == statecode) * 100)

Data.agglomeration <- Data.agglomeration %>%
  filter(assignee_country == "US") %>%
  filter(inventor_country == "US") %>%
  left_join(Inventors %>%
    filter(inventor_country == "US") %>%
    left_join(Combined %>%
      select(patent_number, first_priority_date),
      by = "patent_number") %>%
    filter(!is.na(first_priority_date)) %>%
    mutate(priority_year = lubridate::year(first_priority_date)) %>%
    mutate(statecode = inventor_state) %>%
    distinct() %>%
    na.omit %>%
    group_by(statecode, priority_year) %>%
    summarize(inventors_in_state = n()) %>%
    ungroup(),
    by = c("statecode", "priority_year")) %>%
  left_join(Inventors %>%
    filter(inventor_country == "US") %>%
    left_join(Combined %>%
      select(patent_number, first_priority_date),
      by = "patent_number") %>%
    filter(!is.na(first_priority_date)) %>%
    mutate(priority_year = lubridate::year(first_priority_date) + 1) %>%
    mutate(statecode = inventor_state) %>%
    distinct() %>%
    na.omit %>%
    group_by(statecode, priority_year) %>%
    summarize(lag_inventors_in_state = n()) %>%
    ungroup(),
    by = c("statecode", "priority_year")) %>%
  left_join(Assignees %>%
    filter(assignee_country == "US") %>%
    left_join(Combined %>%
      select(patent_number, first_priority_date),
      by = "patent_number") %>%
    filter(!is.na(first_priority_date)) %>%
    mutate(priority_year = lubridate::year(first_priority_date)) %>%
    mutate(statecode = assignee_state) %>%
    distinct() %>%
    na.omit %>%
    group_by(statecode, priority_year) %>%
    summarize(assignees_in_state = n()) %>%

```

```

    ungroup(),
    by = c("statecode", "priority_year")) %>%
left_join(Assignees %>%
  filter(assignee_country == "US") %>%
  left_join(Combined %>%
    select(patent_number, first_priority_date),
    by = "patent_number") %>%
  filter(!is.na(first_priority_date)) %>%
  mutate(priority_year = lubridate::year(first_priority_date) + 1) %>%
  mutate(statecode = assignee_state) %>%
  distinct() %>%
  na.omit %>%
  group_by(statecode, priority_year) %>%
  summarize(lag_assignees_in_state = n()) %>%
  ungroup(),
  by = c("statecode", "priority_year")) %>%
mutate(inventors_in_state = inventors_in_state %>% replace_na(., 0) %>% winsorize(., 1, 99),
  lag_inventors_in_state = lag_inventors_in_state %>% replace_na(., 0) %>% winsorize(.,
1, 99),
  assignees_in_state = assignees_in_state %>% replace_na(., 0) %>% winsorize(., 1, 99),
  lag_assignees_in_state = lag_assignees_in_state %>% replace_na(., 0) %>% winsorize(.,
1, 99),
  agglomeration_state.inventors = (inventors_in_state - lag_inventors_in_state) %>% winso
rize(., 1, 99),
  agglomeration_state.assignees = (assignees_in_state - lag_assignees_in_state) %>% winso
rize(., 1, 99),
  agglomeration_state.inventors = agglomeration_state.inventors / 1000,
  agglomeration_state.assignees = agglomeration_state.assignees / 1000)

##### CREATE DATASET - STEP 7: COMBINE ALL DATA FOR CITATIONS DATASETS #####
### CREATE DATASET FOR CITATIONS ANALYSES ###

## LOAD IN DATA ##

# PATENT ATTRIBUTES - CITATIONS #

# https://www.mikewoeppel.com/data

Citations <- bind_rows(fread("Data/us_cites_5m.tsv"),
  fread("Data/us_cites_6m.tsv"),
  fread("Data/us_cites_7m.tsv"),
  fread("Data/us_cites_8m.tsv"),
  fread("Data/us_cites_9m.tsv"),
  fread("Data/us_cites_10m.tsv"),
  fread("Data/us_cites_11m.tsv"),
  fread("Data/us_cites_12m.tsv")) %>%
transmute(citing.patent_number = patnum %>% as.numeric, # patent that cites
  cited.patent_number = patcite_num %>% as.numeric) %>% # patent that got cited
distinct() %>%
na.omit()

Citations.backward <- Citations %>% # to speed up

```

```

filter(citing.patent_number %in% Data$patent_number)

Citations.forward <- Citations %>% # to speed up
  filter(cited.patent_number %in% Data$patent_number); rm(Citations); gc()

# PATENT ATTRIBUTES - FIRM IDS #
Citations.firms <- fread("Data/patnum_permco_1976_2023.tsv") %>%
  transmute(patent_number = patnum %>% as.numeric,
            permco = permco %>% as.numeric) %>%
  distinct()

## COMPUTE ORIGINALITY/GENERALITY MEASURES - HHI OF FIRMS ##
Citations.firms <- Data %>%
  select(patent_number) %>%
  distinct() %>%
  left_join(Citations.backward %>%
    left_join(Citations.firms,
              by = c("cited.patent_number" = "patent_number")) %>%
    select(citing.patent_number, permco) %>%
    rename(cited.permco = permco) %>%
    na.omit %>%
    group_by(citing.patent_number, cited.permco) %>%
    summarize(count = n()) %>%
    ungroup() %>%
    group_by(citing.patent_number) %>%
    summarize(originality = 1 - sum((count / sum(count))^2)) %>%
    ungroup(), # high values are original (i.e., cites many firms themselves)
    by = c("patent_number" = "citing.patent_number")) %>%
  left_join(Citations.forward %>%
    left_join(Citations.firms,
              by = c("citing.patent_number" = "patent_number")) %>%
    select(cited.patent_number, permco) %>%
    rename(citing.permco = permco) %>%
    na.omit %>%
    group_by(cited.patent_number, citing.permco) %>%
    summarize(count = n()) %>%
    ungroup() %>%
    group_by(cited.patent_number) %>%
    summarize(generality = 1 - sum((count / sum(count))^2)) %>%
    ungroup(), # high values are general (i.e., is cited by many firms later)
    by = c("patent_number" = "cited.patent_number")) %>%
  select(patent_number,
        patent_originality = originality,
        patent_generality = generality) %>%
  distinct(patent_number, .keep_all = T)

## COMBINE ALL ##
Data <- Data %>%
  left_join(Citations.firms,
            by = "patent_number") %>%
  mutate(patent_originality = asinh(patent_originality) * 100,
         patent_generality = asinh(patent_generality) * 100); rm(Citations.backward, Citations.f

```

```
orward, Citations.firms); gc()
```

```
##### CREATE DATASET - STEP 8: ADD FUTURE COMPETITOR PATENTS #####
```

```
### DETERMINE INVENTOR-FIRM MATCHES ###
```

```
Inventors_firms <- Inventors %>%
  select(inventor_id, patent_number) %>%
  distinct() %>%
  left_join(Assignees %>%
    filter(!is.na(assignee_id)) %>%
    filter(grepl("U.S.", assignee_type)) %>%
    select(patent_number, assignee_id) %>%
    distinct,
    by = "patent_number") %>%
  left_join(Firms,
    by = "patent_number") %>%
  mutate(permco = ifelse(!is.na(permco), permco, assignee_id)) %>%
  left_join(Combined %>%
    mutate(priority_year = lubridate::year(priority_date),
      filing_year = lubridate::year(filing_date)) %>%
    select(patent_number, priority_year, filing_year) %>%
    distinct(),
    by = "patent_number") %>%
  na.omit %>%
  group_by(inventor_id, priority_year, permco) %>%
  summarize(n_patents = n()) %>%
  ungroup()
```

```
### MERGE IN PATENTS IN YEAR t+1 AT OTHER FIRMS ###
```

```
Data <- Data %>%
  left_join(Inventors_firms %>%
    select(inventor_id, priority_year, permco) %>%
    distinct() %>%
    left_join(Inventors_firms %>%
      mutate(priority_year = priority_year - 1),
      by = c("inventor_id", "priority_year")) %>%
    filter(permco.x != permco.y) %>%
    transmute(inventor_id, priority_year, permco = permco.x,
      n_patents_competitor = n_patents %>% replace_na(., 0)) %>%
    na.omit() %>%
    distinct() %>%
    group_by(inventor_id, priority_year, permco) %>%
    summarize(n_patents_competitor = sum(n_patents_competitor, na.rm = T)) %>%
    ungroup(),
    by = c("inventor_id", "priority_year", "permco")) %>%
  mutate(n_patents_competitor = n_patents_competitor %>% replace_na(., 0),
    patent_competitor = as.numeric(n_patents_competitor > 0) * 100); rm(Inventors_firms)
```

```
##### CREATE DATASET - STEP 9: COMBINE ALL DATA FOR FIRM-LEVEL CONSEQUENCES #####
```

```
### ADD FIRM-LEVEL CONSEQUENCES ###
```

```
Data.firm <- Data %>%
  filter(!is.na(treat2)) %>% # drop Texas
  filter(firm != inventor_id) %>% # no individuals
```

```

mutate(patents_in_treated_states = ifelse(treat2 == 1, patent_number, NA),
       patents_in_control_states = ifelse(treat2 == 0, patent_number, NA)) %>%
group_by(firm, filing_year) %>% # select all firm-years
summarize(n_patents = n_distinct(patent_number),
          n_inventors = n_distinct(inventor_id)) %>%
ungroup() %>%
distinct() %>%
left_join(Data %>% # one observation per patent
          filter(!is.na(treat2)) %>% # drop Texas
          group_by(firm, filing_year, patent_number) %>%
          summarize(treat_state_patent = as.numeric(any(treat2 == 1)),
                    cross_treat_control_patent = as.numeric(any(treat2 == 1) & any(treat2 ==
0))),
          disclosure.early = early_publication_request[1],
          disclosure.perc = disclosure.perc[1],
          class = class[1]) %>%
ungroup() %>%
group_by(firm, filing_year, class) %>%
mutate(n_patents_in_class = n()) %>%
ungroup() %>%
group_by(firm, filing_year) %>% # one observation per firm
summarize(perc_treat_state_patents = mean(treat_state_patent, na.rm = T),
          perc_cross_treat_control_patent = mean(cross_treat_control_patent, na.rm
= T),
          avg_perc_disclosure.early = mean(disclosure.early, na.rm = T),
          avg_perc_disclosure = mean(disclosure.perc, na.rm = T),
          hhi_class = 1 - sum((n_patents_in_class / sum(n_patents_in_class))^2)) %
>%
ungroup() %>%
distinct(),
by = c("firm", "filing_year")) %>%
mutate(perc_treat_state_patents = perc_treat_state_patents %>% replace_na(., 0),
       perc_cross_treat_control_patent = perc_cross_treat_control_patent %>% replace_na(., 0))
%>%
arrange(firm, filing_year) %>%
left_join(Data %>% # aggregate controls at firm-year level
          filter(!is.na(treat2)) %>% # drop Texas
          select(firm, filing_year, inventor_id, treat2, same_state) %>%
          group_by(firm, filing_year, inventor_id) %>%
          summarize(treat2 = max(treat2, na.rm = T),
                    same_state = max(same_state, na.rm = T)) %>%
          ungroup() %>%
          mutate(treat2 = treat2 %>% replace_na(., 0),
                 same_state = same_state %>% replace_na(., 0)) %>%
          distinct() %>%
          group_by(firm, filing_year) %>%
          summarize(treat2 = mean(treat2, na.rm = T),
                    same_state = mean(same_state, na.rm = T)) %>%
          ungroup(),
          by = c("firm", "filing_year")) %>%
mutate(perc_treat_state_patents = perc_treat_state_patents * 100,
       perc_cross_treat_control_patent = perc_cross_treat_control_patent * 100,

```

```
hhi_class = asinh(hhi_class) * 100)
```

```
Data.firm <- Data.firm %>% # ex-ante exposure  
  mutate(AvB2.ex_ante = ifelse(filing_year <= 2003, treat2, NA)) %>%  
  group_by(firm) %>%  
  mutate(AvB2.ex_ante = mean(AvB2.ex_ante, na.rm = T)) %>%  
  ungroup() %>%  
  mutate(AvB2.ex_ante = ifelse(filing_year >= 2005, AvB2.ex_ante, 0))
```