# Exercise 2 regression

```r
# Read in the data
data_path <- "/Users/kaz/Desktop/MMA - WINTER Code/"
df <- read_feather(paste0(data_path,"app_applications_starter_coded2.feather"))
```

### Create a quarterly aggregated panel dataset

- how do we aggregate columns like number of race in art unit? because some examiner changes art unit within each quarter
- again how should we deal with art unit columns?

```r
# individual level data
indi_attributes <- df %>%
  select(gender, race, examiner_id) %>%
  distinct(examiner_id, .keep_all = TRUE)
```

### Aggreagate the data by quarter

```r
df_quarter <- df %>%
        group_by(examiner_id, quarter) %>%
        summarize(
                new_applications = mean(new_applications, na.rm = TRUE),
                ISSUED_applications = mean(ISSUED_applications, na.rm = TRUE),
                total_abn_applications = mean(abn_applications, na.rm = TRUE),
                total_PEN_applications = mean(PEN_applications, na.rm = TRUE),
                tenure_days = mean(tenure_days, na.rm = TRUE),
                women_in_art_unit = mean(women_in_art_unit, na.rm = TRUE),
                Asian_in_art_unit = mean(Asian_in_art_unit, na.rm = TRUE),
                Black_in_art_unit = mean(Black_in_art_unit, na.rm = TRUE),
                Other_in_art_unit = mean(Other_in_art_unit, na.rm = TRUE),
                White_in_art_unit = mean(White_in_art_unit, na.rm = TRUE),
                separation_indicator = mean(separation_indicator, na.rm = TRUE),
                au_move_indicator = sum(au_move_indicator, na.rm = TRUE)
        )
```

```
## `summarise()` has grouped output by 'examiner_id'. You can override using the
## `.groups` argument.
```

```r
df_quarter
```

```
## # A tibble: 190,881 x 14
## # Groups:   examiner_id [5,649]
```

```
##    examiner_id quarter new_applications ISSUED_applications
##          <dbl> <chr>              <dbl>               <dbl>
##  1       59012 2004/3                 1                   0
##  2       59012 2006/1                 1                   1
##  3       59012 2006/2                 4                   3
##  4       59012 2006/3                 5                   1
##  5       59012 2006/4                 9                   4
##  6       59012 2007/1                 9                   3
##  7       59012 2007/2                16                   6
##  8       59012 2007/3                11                   7
##  9       59012 2007/4                10                   6
## 10       59012 2008/1                11                   2
## # i 190,871 more rows
## # i 10 more variables: total_abn_applications <dbl>,
## #   total_PEN_applications <dbl>, tenure_days <dbl>, women_in_art_unit <dbl>,
## #   Asian_in_art_unit <dbl>, Black_in_art_unit <dbl>, Other_in_art_unit <dbl>,
## #   White_in_art_unit <dbl>, separation_indicator <dbl>,
## #   au_move_indicator <dbl>
```

**Merge the individual level data with the quarterly aggregated data**

```
# merge individual level data with quarterly aggregated data
df_quarter <- df_quarter %>%
        left_join(indi_attributes, by = "examiner_id")
```

**Change the data types**

```
df_quarter <- df_quarter %>%
        mutate(
                examiner_id = as.integer(examiner_id),
                quarter = as.character(quarter),  # or you could separate into year and quarter
                tenure_days = as.numeric(tenure_days),  # Assuming you keep the .x column
                separation_indicator = as.factor(separation_indicator),
                au_move_indicator = as.integer(au_move_indicator),
                gender = as.factor(gender),
                race = as.factor(race)
        )

# Now check the structure of the dataframe to confirm changes
```

**Check NA and drop them**

```
# colsum na
colSums(is.na(df_quarter))
```

```
##            examiner_id                quarter       new_applications
##                     70                      0                      0
##    ISSUED_applications total_abn_applications total_PEN_applications
```

```
##                       0                       0                       0
##             tenure_days       women_in_art_unit       Asian_in_art_unit
##                       0                       0                       0
##       Black_in_art_unit       Other_in_art_unit       White_in_art_unit
##                       0                       0                       0
##    separation_indicator       au_move_indicator                  gender
##                       0                       0                   28524
##                    race
##                       0
```

```r
# drop na
df_quarter <- df_quarter %>%
        drop_na()
```

```r
# colsum na
colSums(is.na(df_quarter))
```

```
##             examiner_id                 quarter        new_applications
##                       0                       0                       0
##     ISSUED_applications total_abn_applications total_PEN_applications
##                       0                       0                       0
##             tenure_days       women_in_art_unit       Asian_in_art_unit
##                       0                       0                       0
##       Black_in_art_unit       Other_in_art_unit       White_in_art_unit
##                       0                       0                       0
##    separation_indicator       au_move_indicator                  gender
##                       0                       0                       0
##                    race
##                       0
```

```r
dim(df_quarter)
```

```
## [1] 162357     16
```

### to-do

1: single variable analysis 2: correlation 3: some interaction analysis 4: regression

### Explatory Data Analysis

```r
df_unique <- df_quarter %>%
        distinct(examiner_id, .keep_all = TRUE)

# Now create the gender and race distribution tables
gender_distribution <- table(df_unique$gender)
race_distribution <- table(df_unique$race)

# Print the distributions
print(gender_distribution)
```

```
##
##   male female
##   3363   1486
```

```
print(race_distribution)
```

```
##
##    white    Asian    black Hispanic    other
##     3285     1193      167      202        2
```

```
# col wise na sum
colSums(is.na(df_quarter))
```

```
##            examiner_id                quarter       new_applications
##                      0                      0                      0
##    ISSUED_applications total_abn_applications total_PEN_applications
##                      0                      0                      0
##            tenure_days      women_in_art_unit      Asian_in_art_unit
##                      0                      0                      0
##      Black_in_art_unit      Other_in_art_unit      White_in_art_unit
##                      0                      0                      0
##   separation_indicator      au_move_indicator                 gender
##                      0                      0                      0
##                   race
##                      0
```

```
# drop if quarter is 2017/2
df_quarter <- df_quarter %>%
        filter(quarter != "2017/2")

# largest quarter
max(df_quarter$quarter)
```

```
## [1] "2017/1"
```

```
# modify separation indicator
# for each examiner, make the last quarter's separation indicator as 1 and the rest as 0
df_quarter %>%
        group_by(examiner_id) %>%
        mutate(
                separation_indicator = ifelse(
                        quarter == max(quarter),
                        1,
                        0
                )
        )
```

```
## # A tibble: 162,297 x 16
## # Groups:   examiner_id [4,849]
##    examiner_id quarter new_applications ISSUED_applications
##          <int> <chr>              <dbl>               <dbl>
```

```
## 1        59012 2004/3                 1                 0
## 2        59012 2006/1                 1                 1
## 3        59012 2006/2                 4                 3
## 4        59012 2006/3                 5                 1
## 5        59012 2006/4                 9                 4
## 6        59012 2007/1                 9                 3
## 7        59012 2007/2                16                 6
## 8        59012 2007/3                11                 7
## 9        59012 2007/4                10                 6
## 10       59012 2008/1                11                 2
## # i 162,287 more rows
## # i 12 more variables: total_abn_applications <dbl>,
## #   total_PEN_applications <dbl>, tenure_days <dbl>, women_in_art_unit <dbl>,
## #   Asian_in_art_unit <dbl>, Black_in_art_unit <dbl>, Other_in_art_unit <dbl>,
## #   White_in_art_unit <dbl>, separation_indicator <dbl>,
## #   au_move_indicator <int>, gender <fct>, race <fct>
```

**create dataset for each analysis**

```
# for turnover analysis
df_turn <- df_quarter
df_mobi <- df_quarter %>% select(-separation_indicator)
```

**Run regression for turnover analysis**

- time is a variable we created to represent the time period for each observation. It allows the model to account for the time until separation.
-

```
# regression for turnover analysis
df_turn <- df_turn %>%
  group_by(examiner_id) %>%
  arrange(quarter) %>%
  mutate(time = row_number()) %>%
  ungroup()
```

```
# How many examiners are in the data?
length(unique(df_turn$examiner_id))
```

```
## [1] 4849
```

```
# How many quarters are in the data?
length(unique(df_turn$quarter))
```

```
## [1] 69
```

```
# Model with time fixed effects
separation_model <- glm(separation_indicator ~ time + au_move_indicator + new_applications + ISSUED_appl
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(separation_model)
```

```
##
## Call:
## glm(formula = separation_indicator ~ time + au_move_indicator +
##     new_applications + ISSUED_applications + total_abn_applications +
##     total_PEN_applications + gender + race + women_in_art_unit +
##     Asian_in_art_unit + Black_in_art_unit + Other_in_art_unit +
##     White_in_art_unit, family = binomial(link = "logit"), data = df_turn)
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.0953294  0.0149245 -73.391  < 2e-16 ***
## time                    0.0240507  0.0003339  72.035  < 2e-16 ***
## au_move_indicator      -0.0012826  0.0021880  -0.586  0.55774
## new_applications        0.0317154  0.0016501  19.220  < 2e-16 ***
## ISSUED_applications    -0.0033750  0.0017531  -1.925  0.05421 .
## total_abn_applications  0.0267939  0.0022394  11.964  < 2e-16 ***
## total_PEN_applications         NA         NA      NA       NA
## genderfemale            0.0677485  0.0111434   6.080 1.20e-09 ***
## raceAsian              -0.0047479  0.0127994  -0.371  0.71068
## raceblack               0.0764690  0.0288637   2.649  0.00807 **
## raceHispanic           -0.2409096  0.0281888  -8.546  < 2e-16 ***
## raceother              -0.4673176  0.1969399  -2.373  0.01765 *
## women_in_art_unit       0.0434395  0.0034584  12.561  < 2e-16 ***
## Asian_in_art_unit      -0.0212711  0.0017703 -12.015  < 2e-16 ***
## Black_in_art_unit       0.1326289  0.0057533  23.053  < 2e-16 ***
## Other_in_art_unit       0.2296540  0.0432930   5.305 1.13e-07 ***
## White_in_art_unit       0.0049194  0.0006075   8.098 5.58e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 224974  on 162296  degrees of freedom
## Residual deviance: 212807  on 162281  degrees of freedom
## AIC: 212839
##
## Number of Fisher Scoring iterations: 5
```

**Adding fixed effects with dummies might be computationally hard I don't know how to do fixed effects for non-linear like logit**

- Control for Time-Specific Effects: By including time dummies (e.g., for each quarter), you control for any unobserved variables that vary over time but are constant across entities (examiners). This might include factors like policy changes, economic trends, seasonal effects, or other time-related influences.

```
# Assuming df_turn already has 'quarter' as a factor
df_turn$quarter <- factor(df_turn$quarter)

# Model with time fixed effects
separation_model <- glm(separation_indicator ~ time + au_move_indicator + new_applications + ISSUED_appl
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(separation_model)
```

```
##
## Call:
## glm(formula = separation_indicator ~ time + au_move_indicator +
##     new_applications + ISSUED_applications + total_abn_applications +
##     total_PEN_applications + gender + race + women_in_art_unit +
##     Asian_in_art_unit + Black_in_art_unit + Other_in_art_unit +
##     White_in_art_unit + model.matrix(~quarter - 1, data = df_turn),
##     family = binomial(link = "logit"), data = df_turn)
##
## Coefficients: (1 not defined because of singularities)
##                                                        Estimate Std. Error
## (Intercept)                                           2.842e+11  1.652e+11
## time                                                  2.083e-02  4.585e-04
## au_move_indicator                                    -5.581e-03  2.231e-03
## new_applications                                      3.757e-02  2.202e-03
## ISSUED_applications                                  -2.726e-03  2.399e-03
## total_abn_applications                                2.313e-02  2.887e-03
## total_PEN_applications                                       NA         NA
## genderfemale                                          7.640e-02  1.123e-02
## raceAsian                                            -9.352e-03  1.287e-02
## raceblack                                             7.966e-02  2.899e-02
## raceHispanic                                         -2.520e-01  2.842e-02
## raceother                                            -4.569e-01  1.982e-01
## women_in_art_unit                                     4.746e-02  3.476e-03
## Asian_in_art_unit                                    -2.057e-02  1.816e-03
## Black_in_art_unit                                     1.306e-01  5.788e-03
## Other_in_art_unit                                     2.242e-01  4.352e-02
## White_in_art_unit                                     4.169e-03  6.240e-04
## model.matrix(~quarter - 1, data = df_turn)quarter2000/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2000/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2000/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2000/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2001/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2001/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2001/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2001/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2002/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2002/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2002/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2002/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2003/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2003/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2003/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2003/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2004/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2004/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2004/3 -2.842e+11  1.652e+11
```

```
## model.matrix(~quarter - 1, data = df_turn)quarter2004/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2005/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2005/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2005/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2005/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2006/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2006/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2006/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2006/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2007/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2007/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2007/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2007/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2008/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2008/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2008/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2008/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2009/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2009/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2009/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2009/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2010/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2010/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2010/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2010/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2011/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2011/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2011/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2011/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2012/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2012/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2012/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2012/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2013/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2013/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2013/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2013/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2014/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2014/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2014/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2014/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2015/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2015/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2015/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2015/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2016/1 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2016/2 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2016/3 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2016/4 -2.842e+11  1.652e+11
## model.matrix(~quarter - 1, data = df_turn)quarter2017/1 -2.842e+11  1.652e+11
##                                                         z value Pr(>|z|)
## (Intercept)                                               1.720  0.08537 .
## time                                                     45.441  < 2e-16 ***
## au_move_indicator                                        -2.501  0.01238 *
```

```
## new_applications                                            17.064  < 2e-16 ***
## ISSUED_applications                                         -1.136  0.25582
## total_abn_applications                                       8.010 1.14e-15 ***
## total_PEN_applications                                          NA       NA
## genderfemale                                                 6.804 1.02e-11 ***
## raceAsian                                                   -0.727  0.46740
## raceblack                                                    2.747  0.00601 **
## raceHispanic                                                -8.865  < 2e-16 ***
## raceother                                                   -2.305  0.02114 *
## women_in_art_unit                                           13.654  < 2e-16 ***
## Asian_in_art_unit                                          -11.328  < 2e-16 ***
## Black_in_art_unit                                           22.559  < 2e-16 ***
## Other_in_art_unit                                            5.151 2.59e-07 ***
## White_in_art_unit                                            6.681 2.38e-11 ***
## model.matrix(~quarter - 1, data = df_turn)quarter2000/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2000/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2000/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2000/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2001/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2001/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2001/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2001/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2002/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2002/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2002/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2002/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2003/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2003/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2003/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2003/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2004/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2004/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2004/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2004/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2005/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2005/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2005/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2005/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2006/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2006/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2006/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2006/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2007/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2007/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2007/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2007/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2008/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2008/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2008/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2008/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2009/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2009/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2009/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2009/4  -1.720  0.08537 .
```

```
## model.matrix(~quarter - 1, data = df_turn)quarter2010/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2010/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2010/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2010/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2011/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2011/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2011/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2011/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2012/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2012/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2012/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2012/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2013/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2013/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2013/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2013/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2014/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2014/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2014/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2014/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2015/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2015/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2015/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2015/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2016/1  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2016/2  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2016/3  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2016/4  -1.720  0.08537 .
## model.matrix(~quarter - 1, data = df_turn)quarter2017/1  -1.720  0.08537 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 224974  on 162296  degrees of freedom
## Residual deviance: 210569  on 162212  degrees of freedom
## AIC: 210739
##
## Number of Fisher Scoring iterations: 25
```

**Without Time Dummies**

| Variable | Coefficient | Significance |
|---|---|---|
| Intercept | -1.0953294 | *** |
| time | 0.0240507 | *** |
| au_move_indicator | -0.0012826 | |
| new_applications | 0.0317154 | *** |
| ISSUED_applications | -0.0033750 | . |
| total_abn_applications | 0.0267939 | *** |
| total_PEN_applications | NA | NA |
| genderfemale | 0.0677485 | *** |
| raceAsian | -0.0047479 | |

| Variable | Coefficient | Significance |
|---|---|---|
| raceblack | 0.0764690 | ** |
| raceHispanic | -0.2409096 | *** |
| raceother | -0.4673176 | * |
| women_in_art_unit | 0.0434395 | *** |
| Asian_in_art_unit | -0.0212711 | *** |
| Black_in_art_unit | 0.1326289 | *** |
| Other_in_art_unit | 0.2296540 | *** |
| White_in_art_unit | 0.0049194 | *** |

### With Time Dummies

| Variable | Coefficient | Significance |
|---|---|---|
| Intercept | 2.842e+11 | |
| time | 0.02405 | *** |
| au_move_indicator | -0.0012826 | |
| new_applications | 0.0317154 | *** |
| ISSUED_applications | -0.0033750 | |
| total_abn_applications | 0.0267939 | *** |
| total_PEN_applications | NA | NA |
| genderfemale | 0.0677485 | *** |
| raceAsian | -0.009352 | |
| raceblack | 0.0764690 | ** |
| raceHispanic | -0.2490906 | *** |
| raceother | -0.4673176 | * |
| women_in_art_unit | 0.0434395 | *** |
| Asian_in_art_unit | -0.0212711 | *** |
| Black_in_art_unit | 0.1326289 | *** |
| Other_in_art_unit | 0.2296540 | *** |
| White_in_art_unit | 0.0049194 | *** |

### Run regression for mobility analysis

The Poisson model is appropriate when your response variable represents count data and you expect the variance to be equal to the mean (a key assumption of the Poisson distribution). If the variance significantly exceeds the mean, a negative binomial model might be more appropriate.

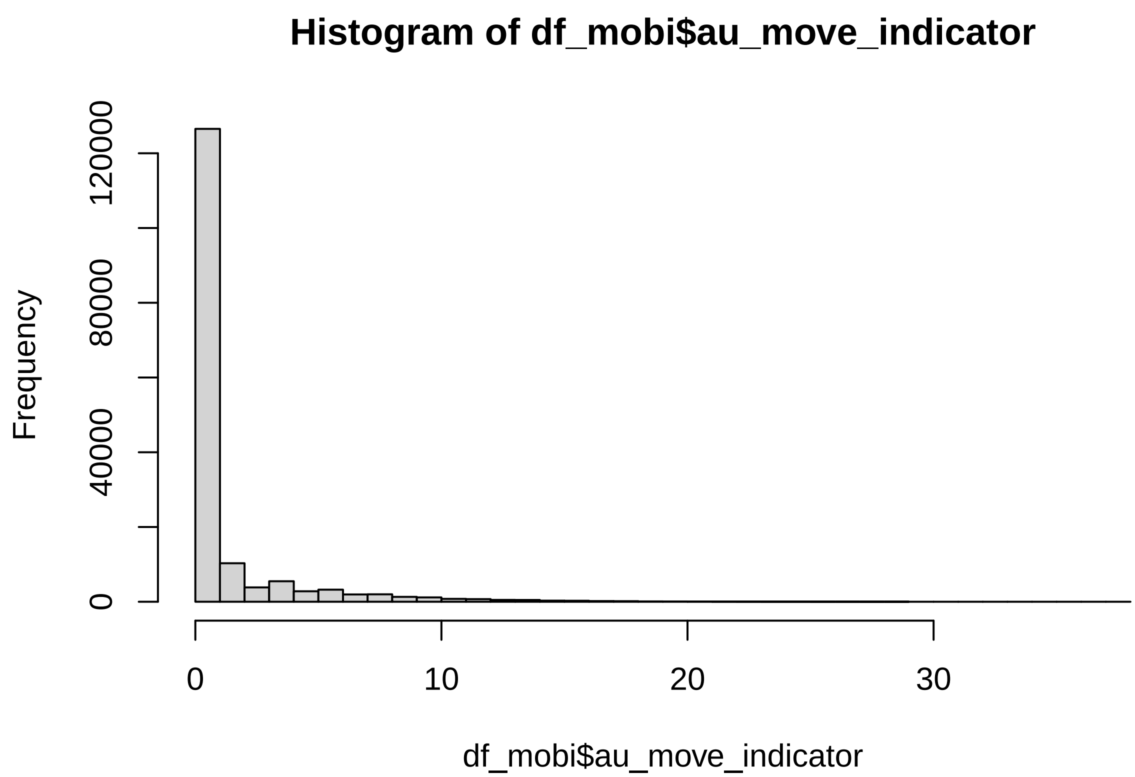
```
# check the assumption
mean(df_mobi$au_move_indicator)
```

```
## [1] 1.228452
```

```
var(df_mobi$au_move_indicator)
```

```
## [1] 8.266517
```

```
# au_move_indicator hist
hist(df_mobi$au_move_indicator, breaks = 50)
```




```
library(plm)
```

```
##
## Attaching package: 'plm'

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
# Convert the data frame to a pdata.frame, specifying the index for entity and time
pdata <- pdata.frame(df_mobi, index = c("examiner_id", "quarter"))

# Fit the fixed effects model
fe_model <- plm(au_move_indicator ~ new_applications + ISSUED_applications +
  total_abn_applications + tenure_days + gender + race +
  women_in_art_unit + Asian_in_art_unit + Black_in_art_unit +
  Other_in_art_unit + White_in_art_unit,
                data = pdata, model = "within")

summary(fe_model)
```

```
## Oneway (individual) effect Within Model
##
## Call:
```

```
## plm(formula = au_move_indicator ~ new_applications + ISSUED_applications +
##     total_abn_applications + tenure_days + gender + race + women_in_art_unit +
##     Asian_in_art_unit + Black_in_art_unit + Other_in_art_unit +
##     White_in_art_unit, data = pdata, model = "within")
##
## Unbalanced Panel: n = 4849, T = 1-69, N = 162297
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -19.318348  -1.054419  -0.099461   0.658011  26.512773
##
## Coefficients:
##                          Estimate  Std. Error  t-value  Pr(>|t|)
## new_applications       -0.00030712  0.00051513  -0.5962  0.551047
## ISSUED_applications     0.08934460  0.00127477  70.0867 < 2.2e-16 ***
## total_abn_applications  0.24136075  0.00184671 130.6980 < 2.2e-16 ***
## women_in_art_unit      -0.00051123  0.00446900  -0.1144  0.908925
## Asian_in_art_unit       0.01950606  0.00316120   6.1705 6.825e-10 ***
## Black_in_art_unit       0.07589236  0.01050402   7.2251 5.031e-13 ***
## Other_in_art_unit       0.20590610  0.07215758   2.8536  0.004324 **
## White_in_art_unit       0.07331964  0.00108585  67.5230 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1041400
## Residual Sum of Squares: 737380
## R-Squared:      0.29193
## Adj. R-Squared: 0.27009
## F-statistic: 8113.8 on 8 and 157440 DF, p-value: < 2.22e-16
```

- you can see that time-invarient variables are dropped because they are not informative in the fixed effects model (gneder, race, tenure days...)

| Variable | Coefficient | Significance |
|---|---|---|
| new_applications | -0.00030712 | |
| ISSUED_applications | 0.08934460 | *** |
| total_abn_applications | 0.24136075 | *** |
| women_in_art_unit | -0.00051123 | |
| Asian_in_art_unit | 0.01950606 | *** |
| Black_in_art_unit | 0.07589236 | *** |
| Other_in_art_unit | 0.20590610 | ** |
| White_in_art_unit | 0.07331964 | *** |