Code Replication Instructions

Yuan Yue

2019-09-02

This script replicates all the estimation procedures, tables and graphs in the paper: *Earthquake risk embedded in property prices: Evidence from five Japanese cities.*

The structure of the code is as follows:

1. Estimation of ETAS model
2. Simulation of short-run earthquake probabilities, using the estimated ETAS parameters and historical earthquake catalogue. Replication of Figures 1 and 2
3. Characteristics of the housing dataset. Replication of Table 1
4. Characteristics of the JSHIS long run probabilities. Replication of Table 2
5. Main estimation steps. Replication of Table 3, Figure 3
6. Importance ordering. Replication of Tables 4-5
7. Sensitivity analysis. Replication of Tables 6-11 and Figure 4

Download and installation instructions

- Create a directory called "earthquake-risk-paper" and set it as the current working directory in `R`
- Download the compressed file for R package "mvecr" ("mvecr_0.1.0.zip") from https://github.com/yy112/earthquake-risk into the current working directory
- Within the current working directory, create a subdirectory called "data" and download the files "individual_data.zip", "Xpsi-1.csv", and "JMA_records.csv" into this subdirectory
- Use the following `r` code to install all relevant packages and read data

```
#
# install.packages("PtProcess")
# install.packages("dplyr")
# install.packages("readr")
# install.packages("mvecr_0.1.0.tar.gz", repos = NULL, type = "source")

library(PtProcess)
library(dplyr)
library(mvecr)
library(readr)


# read data
individual_data <- readr::read_csv(paste0(path, "/data/individual_data.zip"),
                                   col_types = cols(
                                      .default = col_double(),
                                      t = col_character(),
                                      Type = col_character(),
                                      Area = col_character(),
                                      Area.Ward.City = col_character(),
                                      City = col_character(),
                                      Nearest.station.Name = col_character(),
                                      Station.City = col_character(),
                                      Building.structure = col_character(),
```

```
                                        City.Planning = col_character()
                                      ))
Xpsi1 <- read.table(paste0(path, "/data/Xpsi-1.csv"),
                     header = TRUE, sep = ",", quote = "\"",
                    dec = ".", stringsAsFactors = F)


city_range <- read.table(paste0(path, "/data/city_range.csv"),
                         header = TRUE, sep = ";",
                         quote = "\"", dec = ".", stringsAsFactors = F)

data <- read.table(paste0(path, "/data/JMA_records.csv"),
                    header = TRUE, sep = ",", quote = "\"",
                    dec = ".", stringsAsFactors = F)
```

Description of data files

- individual_data.zip (contains individual_data.csv, a csv file of 94446 KB): This is the main data file of our paper, with 331343 observations of housing sample from 2006Q2-2015Q3. This data contains the following information: log housing price, total transaction price, transaction period, housing type, city, district(area), ward, name of the nearest station, distance to nearest station, square footage, total floor area, building age, building structure, long term earthquake probability forecast provided by JSHIS, macroeconomic variables, indicators of ward information, dummy variables for the transaction quarter, etc.
- city_range.csv (1 KB): This file contains the names of the five cities included in our study and the corresponding range of coordinates for the space window chosen for each city. The range of coordinates are used for selecting the earthquake records to be included in the estimation of ETAS models.
- JMA_records.csv (24245 KB): This file contains all earthquake records downloaded from the JMA website, with 194882 observations ranging from 1923-01-01 to 2015-12-31. The information related to each record consists of time, location, magnitude, etc. A subset of this file is used to estimate ETAS models.

- Xpsi-1.csv (2 KB) This file is the simulation result of the ETAS models. For each city in our study, the ETAS model is simulated and the simulated probability of having an earthquake exceeding the magnitude threshold in each quarter is recorded in this dataset.

Estimation of ETAS model

The following code will be used to replicate the results of the ETAS model estimation. (Table 48 and Table 49 of the data document.) The estimation takes less than 1 minute.

```
# estimation of ETAS model for each city
origin <- as.Date("1970-01-01")
cat <- EQcatalog(data, t_start = 1970, t_end = 2015, depth = 100, mag_min = 4,
                 origin = origin)
magMin <- c(4.5, 4.5, 4.5, 4.5, 5)
init.params <- c(.1, .1, .5, 1.2, 1.1, 1/mean(cat$magnitude), 0)

# estimate the ETAS model for each city
list.Est <- {}
for(i in 1:length(city_range$City)){
```

```r
  city <- city_range$City[i]
  print(city)
  out <- etas_estim(cat, city = city, city_range = city_range,
                    magMin=magMin[i], params=init.params, t0 = origin, tN = "2016-01-01")
  assign(paste0(city,"_Est"), out)
  print(etas_test(out, T))
  list.Est[[i]] <- out
}

sum.table <- data.frame(city = city_range$City, stringsAsFactors = F)
for(i in 1:5){
  city <- city_range$City[i]
  out <- list.Est[[i]]
  # eq_estim(cat, city = city, city_range = city_range,
  #              magMin=magMin[i], params=init.params)
  ks <- etas_test(out, plot = F)
  sum.table$ks.pval[i] <- ks$p.val
  # assign(paste0(city,"_Est"), out)
  sum.table$N[i] <- nrow(out$data)
  mu <- out$params[1]
  A <- out$params[2]
  alpha <- out$params[3]
  CC <- out$params[4]
  p <- out$params[5]
  K <- A*CC^p
  beta <- alpha
  sum.table$mu[i] <- mu
  sum.table$K[i] <- K
  sum.table$C[i] <- CC
  sum.table$p[i] <- p
  sum.table$beta[i] <- beta
}

kable(city_range %>%
        select(City, latMin, latMax, lngMin, lngMax, codeMin, codeMax) %>%
        arrange(factor(City, levels = c("Tokyo", "Osaka",
                                        "Nagoya", "Fukuoka", "Sapporo"))),
      caption = "Space window of the earthquake catalog",
      digits = 1)

kable(sum.table %>%
        arrange(factor(city, levels = c("Tokyo", "Osaka",
                                        "Nagoya", "Fukuoka", "Sapporo"))),
      caption = "ETAS estimation results for each city", digits = 4)
```

Simulation of short-run earthquake probabilities

The following code can be used to simulate the short-run earthquake probabilities. Note that running this code takes a long time. We used 30000 runs to generate the final similated probabilities, while each run takes about 24 hours for each city (without parallelization). To continue the replication fo this paper use Xpsi-1.csv for the rest of this document.

```r
# This file generates the additional vector of regressor XPsi, for given psi.
library(PtProcess)
library(R.utils)
library(snowfall)
library(zoo)
source('etas_funcs.R')
########################################################################################
# load saved objects X(without probabilities), y, H, district, time, type, ...
load('ETAS_results.RData')

# input arguments: number of simulations, which city, etc.
# input <- commandArgs(trailingOnly=TRUE)
# input = "Nsim=1000,cpucores=4,i=3"
input<- strsplit(input, split=",")[[1]]
for(w in 1:length(input)){
  eval(parse(text=input[w]))
}
tic <- proc.time()[3]

input.ls <- c("psi", "Nsim", "city#", "id")
psi <- ifelse(exists("psi"), psi, 1)
print(paste("psi=", psi))
Nsim <- ifelse(exists("Nsim"), Nsim,  1000)
print(paste("Nsim=", Nsim))
i <- ifelse(exists("i"), i,  1)
print(paste("city#=", i))
id <- ifelse(exists("id"), id,  sample(1:2000)[1])
print(paste("id=", id))
cpucores <- ifelse(exists("cpucores"), cpucores,  4)
print(paste("cpucores=", cpucores))


# generate the time vector: 2006Q2-2015Q3
q.start <- 2006.25
q.end <- 2015.5
time <- as.yearqtr(seq(q.start, q.end, 1/4))
time.date <- gsub(" Q4", "-10-01",
                  gsub(" Q3", "-07-01",
                       gsub(" Q2", "-04-01",
                            gsub(" Q1", "-01-01", time))))
time.end <- gsub(" Q4", "-10-01",
                 gsub(" Q3", "-07-01",
                      gsub(" Q2", "-04-01",
                           gsub(" Q1", "-01-01", as.yearqtr(q.end+1/4)))))


# range, estimated parameters and magnitude threshold for given city
city_range_c <- city_range[i, ]
city <- city_range_c$City
list.Est_c <- list(list.Est[[i]])
magMin_c <- magMin[i]

# # initiate the parallel computing environment
```

```
cpucores <- as.integer(cpucores) #by default snowfall will use the total number of processors, so this
cols <- Nsim/250
sfInit(parallel=TRUE, cpus=cpucores) #
sfLibrary(PtProcess)
sfLibrary(R.utils)
# sfExport(list('gen_Xpsi_city', 'etas_prob', 'expmap',
#               'list.Est_c', 'city_range_c', 'time.date', 'time.end', 'origin', 'magMin_c', 'n.sim'))
sfExportAll()
# n.sim <- as.integer(Nsim/cpucores)
# generate
# parallel execution
results <- sfLapply(x=as.list(1:cols),
                    fun = gen_Xpsi_city, list.Est = list.Est_c,
                    city_range = city_range_c,
                    time = time.date, time.end = time.end, date.start = origin,
                    threshold = 5.5, magMin = magMin_c, n.sim = 250,  time.out = 60)
# sequential execution
# results <- gen_Xpsi_city(Iter.val = 1, list.Est = list.Est_c,
#                          city_range = city_range_c,
#                          time = time.date, time.end = time.end, date.start = origin,
#                          threshold=5.5, magMin = magMin_c, n.sim = Nsim, time.out = 30)

# store the generated probabilities
# Xpsi <- Reduce("+", results) / length(results)
Xpsi <- matrix(0, ncol = cols, nrow = length(time.date))
for(n in 1:length(results)){
  Xpsi[, n] <- as.numeric(results[[n]]$threshold[, 1])
}

elapsed <- proc.time()[3] - tic
#
sfRemoveAll()
sfStop()


write.csv(Xpsi, paste("Xpsi", psi, city, Nsim, id, ".csv", sep="-"), row.names = F)

toc <- proc.time()[3]
print(paste("time elapsed:", toc - tic))
```

The following code can be used to replicate Figure 1 and 2 of the paper.

```
addlabel <- function(x, y, len, lab, dis = 0.03){
  arrows(x0 = x, y0 = y, x1 = x, y1 = y-len, length = 0.08)
  text(x = x, y=y+dis, labels = lab, cex = 0.8)
}


Xpsi1 <- read.table(paste0(path, "/data/Xpsi-1.csv"),
                    header = TRUE, sep = ",", quote = "\"",
                   dec = ".", stringsAsFactors = F)

# short run probs
date.start <- "1970-01-01"
```

```r
time <- c("2006 Q2", "2006 Q3", "2006 Q4",
          paste0("2007 Q", 1:4),
          paste0("2008 Q", 1:4),
          paste0("2009 Q", 1:4),
          paste0("2010 Q", 1:4),
          paste0("2011 Q", 1:4),
          paste0("2012 Q", 1:4),
          paste0("2013 Q", 1:4),
          paste0("2014 Q", 1:4),
          "2015 Q1", "2015 Q2", "2015 Q3")
time1 <- gsub(" Q4", "-10-01",
              gsub(" Q3", "-07-01",
                   gsub(" Q2", "-04-01",
                        gsub(" Q1", "-01-01", time))))
dates.EQ <- as.Date(c("2006-11-15", "2007-01-13", "2007-03-25", "2007-07-16", "2008-06-14",
                      "2009-08-09",  "2009-08-11", "2010-02-26", "2010-12-21",
                      "2011-03-11",
                      "2012-01-01", "2012-12-07", "2013-10-26", "2015-05-30"))
labels.EQ <- julian(dates.EQ,  origin = as.Date(date.start))
text.EQ <- paste0("circle", 1:length(dates.EQ))
names.EQ <- c("Kuril Islands, 8.3M_W", "Kuril Islands, 8.1M_W",
              "Noto Hanto, 6.9M_W", "Chuetsu offshore, 6.6M_w", "Iwate-Miyagi Nairiku, 6.9M_W",
              "Izu Islands, 7.0M_W", "Shizuoka, 6.6M_W",
              "Ryukyu Islands, 7.0M_W", "Bonin Islands, 7.4M_W",
              "Tohoku, 9.1M_W", "Izu Islands, 6.8M_W",
              "Kamaishi, 7.3M_W",
              "Off the east coast of Honshu, 7.1M_W", "Bonin Islands, 7.8M_W")


Est <- list.Est[[5]]
par(xpd=TRUE, pty='m', mar=c(5,5,2,5))
at <- seq(from=1, by = 4, to = 38)
ticks <- julian(as.Date(time1),  origin = as.Date(date.start))
plot(ticks, Xpsi1$Tokyo, type="l", ylim = c(0.25,0.85), lty=2,
     ylab = "90-days probabilities", xlab = "time",
     yaxt='n',
     xaxt='n')
axis(1, at=julian(as.Date(time1),  origin = as.Date(date.start))[at],
     labels=time[at], cex.axis = 0.5)
axis(2, labels = c(0.2,0.4,0.6,0.8), at=c(0.2,0.4,0.6,0.8))
legend(x=ticks[13]+200, y=0.85, legend = c("90-days probs, Tokyo", "ground intensity, Tokyo"),
       cex=0.8,
       lty = c(2,1) , xjust = 1, yjust = 1, text.width = strwidth("ground intensity, Tokyo"))

addlabel(labels.EQ[13], 0.72, 0.12, 'c5')
addlabel(labels.EQ[6], 0.55, 0.12, 'c2')
addlabel(labels.EQ[10], 0.82, 0.05, 'c3')
addlabel(labels.EQ[4], 0.55, 0.12, 'c1')
addlabel(labels.EQ[11], 0.73, 0.12, 'c4')
par(new = T)
lambda_t <- ticks[1]:ticks[length(ticks)]
lambda_y <- etas_gif(data = Est$data, evalpts =lambda_t, param = Est$params)
plot(lambda_t, log(lambda_y), type = 'l', lty=1,col='black',
```

```
     axes=F, xlab=NA, ylab=NA, ylim=c(-4.5, 10))
axis(side = 4, labels = c(-4,-2,0,2), at = c(-4,-2,0,2))
mtext(side = 4, line = 3, '(log) ground intensity lambda(t)')
```

```
par(xpd=TRUE, pty='m', mar=c(5,5,2,5))
at <- seq(from=1, by = 4, to = 38)
ticks <- julian(as.Date(time1),  origin = as.Date(date.start))
plot(ticks, Xpsi1$Nagoya, type="l", ylim = c(0.13,0.2), lty=2,
     ylab = "90-days probabilities", xlab = "time",
     yaxt='n',
     xaxt='n')
axis(1, at=julian(as.Date(time1),  origin = as.Date(date.start))[at],
     labels=time[at], cex.axis = 0.5)
axis(2, labels = c(0.14,0.16,0.18,0.2), at=c( 0.14,0.16,0.18,0.2))


legend(x=ticks[13]+250, y=0.198, legend = c("90-days probs, Nagoya", "ground intensity, Nagoya"),
       cex=0.8,
       lty = c(2,1) , xjust = 1, yjust = 1, text.width = strwidth("ground intensity, Nagoya"))


addlabel(labels.EQ[3], 0.16, 0.01, 'c1', dis=0.003)
addlabel(labels.EQ[7], 0.18, 0.01, 'c2', dis=0.003)
addlabel(labels.EQ[10], 0.185, 0.008, 'c3', dis=0.003)


par(new = T)
Est <- list.Est[[2]]
lambda_t <- ticks[1]:ticks[length(ticks)]
lambda_y <- etas_gif(data = Est$data, evalpts =lambda_t, param = Est$params)
plot(lambda_t, log(lambda_y), type = 'l', lty=1,col='black',
     axes=F, xlab=NA, ylab=NA, ylim=c(-4.5,5))
axis(side = 4, labels = c(-4,-3,-2,-1,0), at =  c(-4,-3,-2,-1,0))
mtext(side = 4, line = 3, '(log) ground intensity lambda(t)')
```

Housing dataset

The following code can be used to replicate Table 1 of the paper.

```
sum_df <- individual_data %>%
  mutate(Ward = sapply(strsplit(as.character(Area.Ward.City),','),
                       "[", 2)) %>%
  group_by(City) %>%
  summarise(Ward = length(unique(Ward)),
            District = length(unique(Area.Ward.City)),
            Land_building = sum(Type_LandBldg),
            Land_only = sum(Type_LandOnly),
            Condo = sum(Type_Condo),
            Station = length(unique(Nearest.station.Name))) %>%
  arrange(factor(City, levels = c("Tokyo", "Osaka",
                                  "Nagoya", "Fukuoka", "Sapporo")))
  sum_df <- rbind(sum_df, c("Total", colSums(sum_df %>% select(-City))))
```

```
kable(sum_df, caption = "Distribution of properties over cities, wards and districts")
```

Long-run earthquake intensities

The following code can be used to replicate Table 2 of the paper.

```
sum_JSHIS_1 <- individual_data %>%
  mutate(unique = !duplicated(Area.Ward.City)) %>%
  filter(unique == 1) %>%
  group_by(City) %>%
  summarise(mean = mean(JSHIS_I45),
            min = min(JSHIS_I45),
            q25 = quantile(JSHIS_I45, 0.25),
            q50 = quantile(JSHIS_I45, 0.5),
            q75 = quantile(JSHIS_I45, 0.75),
            max = max(JSHIS_I45),
            sd = sqrt(var(JSHIS_I45))) %>%
  arrange(factor(City, levels = c("Tokyo", "Osaka",
                                  "Nagoya", "Fukuoka", "Sapporo")))


kable(sum_JSHIS_1, caption = "Seizmic hazard probabilities per city, exceeding intensity level 5 lower,
      digits = 2)


sum_JSHIS_2 <- individual_data %>%
  mutate(unique = !duplicated(Area.Ward.City)) %>%
  filter(unique == 1) %>%
  group_by(City) %>%
  summarise(mean = mean(JSHIS_I55),
            min = min(JSHIS_I55),
            q25 = quantile(JSHIS_I55, 0.25),
            q50 = quantile(JSHIS_I55, 0.5),
            q75 = quantile(JSHIS_I55, 0.75),
            max = max(JSHIS_I55),
            sd = sqrt(var(JSHIS_I55))) %>%
  arrange(factor(City, levels = c("Tokyo", "Osaka",
                                  "Nagoya", "Fukuoka", "Sapporo")))


kable(sum_JSHIS_2, caption = "Seizmic hazard probabilities per city, exceeding intensity level 6 lower,
      digits=2)
```

Main estimation results

The following code can be used to replicate Table 3 of the paper. For each model specification and each choice of ψ, the estimation takes around 2 hours.

results_LRonly, results_LR_objSR, and results_base are data frames containing the estimation results (parameter estimates, standard errors and log likelihood values) as reported in the last three columns of Table 3.

```
colName.i <- "Area.Ward.City"
colName.t <- "t"
colName.p <- "Type"
# names of the columns of regressors X
Xnames_all <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                 "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                "RC", "SRC", "RC_SRC", "S", "W",
                "LU_Resid", "LU_Comm", "LU_Industr",
                "Region_Residential", "Region_Commercial",
                "Region_Industrial","Region_PotResidential",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",  "Int_rate", "log.TOPIX",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                "PctForeign", "Ndaycare", "Nkindergtn", "Nagedhome","Nhosp",
                 "Nlargeretail", "Ndepstore",
                 "JSHIS_I45", "JSHIS_I55", "JSHIS_I45_55",
                "JSHIS_I45_station", "JSHIS_I55_station",
                "JSHIS_I45_55_station",
                "Xpsi_obj",
                "Q1", "Q2", "Q3", "Q4", "Q123",
                "Q_after_Fukushima", "age_W")

Xnames_base <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
            "distance.num", "area.m2.num", "total.floor.area.m2.num",
            "building.age",
            "LandBldg_RC", "LandBldg_S", "LandBldg_W",
            "built.1981_2000", "built.after2000" ,
            "Urban_Control",
            "max.building.coverage.ratio", "max.floor.area.ratio",
            "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
            "log.nGDP", "log.CPI",
            "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
            "JSHIS_I45_55", "JSHIS_I55", "Xpsi")



# generate averages of y, X and cell counts H
data_vec <- mvecr::vectorize(data = individual_data, colName.i = colName.i,
                 colName.t = colName.t, colName.p = colName.p,
                 colName.y = "log.price",
                 colName.X = Xnames_all)
# retrieve each component of the results
H <- data_vec$H
y <- data_vec$y
X <- data_vec$X
district <- data_vec$district
```

```
time <- data_vec$time
type <- data_vec$type

# choose the theta parameters to include
# in the 2 error component case: 6 parameters for the zeta matrix and 6 for the epsilon matrix
include_2error <- c(rep(1, 6), rep(0,6), rep(1,6))
# choose initial parameters for optimization
initpar <- c(-1.5, 0.1, -0.1, -2.1, -0.01, -1.1,
             -2.5, 0.02, -0.1, -3.5,  0.1, -3.5,
             -0.9, 0.008, 0.005, -0.9, 0.01, -0.9)
# results for the model with only long run risk variables (model 1)
results_LRonly <- mvecr::ec_reg(data_X = X, data_y = y, data_H = H,
                  var = setdiff(Xnames_base, "Xpsi"),
                  district = district, time = time, type = type,
                  par.include = include_2error,
                  par.init = initpar)
# results for the model when the short run risk variable is objective (model 2)
results_LR_objSR <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                        psi = 1, method = "p",
                        district = district, time = time, type = type,
                        var = Xnames_base,
                        par.include = include_2error,
                        par.init = initpar)
#
# results for the base model (model 3)
results_base <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.74, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_base,
                    par.include = include_2error,
                    par.init = initpar)
```

Note that in order to find the value of psi that maximizes likelihood, we perform a grid search over possible values between 0.1 to 10. Refine the grid by each iteration to find the final value (precise up to 2 digits). The following code can be used to replicate this process.

```
list0 <- seq(from = 0.5, to = 10, by = 0.5)
list1 <- seq(from = 3.65, to = 3.85, by = 0.01)
psi_optim <- mvecr::opt_psi(data_X=X, data_y=y, data_H=H,
        list_psi = list1, method = "p",
        district = district, time = time, type = type,
        var = Xnames_base,
        par.include = include_2error,
        par.init = initpar,
        plot = TRUE)
# final estimate: psi_hat = 3.74 maximizes likelihood
```

The following code can be used to replicate Figure 3 of the paper.

```
stepsize <- 1e-3
p <- seq(from = 0 + stepsize, to = 1, by = stepsize)

par(xpd=T, xaxs='i',yaxs='i', pty='s')
psi <- 3.74
plot(p, prelec(p, psi), type = "l", lty=1, xlab='objective probabilities', ylim = c(0,1), xlim=c(0,1),
```

```
     ylab = "distorted probabilities", lwd=0.8, col='black',  asp=1, xaxt='n', yaxt='n')
axis(2, labels = c(0,0.2,0.4,0.6,0.8,1), at=c(0,0.2,0.4,0.6,0.8,1))
axis(1, labels = c(0,0.2,0.4,0.6,0.8,1), at=c(0,0.2,0.4,0.6,0.8,1))
lines(p, prelec(p, 1), lty=3, lwd=0.5, col='grey')
```

Importance ordering

The following code can be used to replicate Table 4 and 5 of the paper.

```
individual_data$Xpsi <- prelec(individual_data$Xpsi_obj, psi = 3.74)
# generate the X
X.individual <- individual_data[, c("t", "Area.Ward.City", "log.price",
                                    "Type",
                              "Type_LandBldg", "Type_LandOnly", "Type_Condo",
                              "distance.num", "area.m2.num",
                              "total.floor.area.m2.num",
                              "building.age",
                              "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                              "built.1981_2000", "built.after2000" ,
                              "Urban_Control",
                              "max.building.coverage.ratio",
                              "max.floor.area.ratio",
                              "City_Fukuoka", "City_Nagoya", "City_Osaka",
                              "City_Sapporo",
                              "log.nGDP", "log.CPI",
                              "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                              "JSHIS_I45_55", "JSHIS_I55",
                              "Xpsi_obj", "Xpsi")]
coef <- as.data.frame(t(base_results$coef))
colnames(coef) <- base_results$var
X.individual[is.na(X.individual)] <- 0
X.individual$inf_constant <- X.individual$Type_LandBldg * coef$constant_LandBldg +
  X.individual$Type_LandOnly * coef$constant_LandOnly +
  X.individual$Type_Condo * coef$constant_Condo
X.individual$inf_distance <- X.individual$distance.num * coef$distance.num
X.individual$inf_area <- X.individual$area.m2.num * coef$area.m2.num
X.individual$inf_floorarea <- X.individual$total.floor.area.m2.num * coef$total.floor.area.m2.num
X.individual$inf_age <- X.individual$building.age * coef$building.age
X.individual$inf_bldgyr <- X.individual$built.1981_2000 * coef$built.1981_2000 +
  X.individual$built.after2000 * coef$built.after2000
X.individual$inf_bldgstructure <- X.individual$LandBldg_RC * coef$LandBldg_RC +
  X.individual$LandBldg_S * coef$LandBldg_S +
  X.individual$LandBldg_W * coef$LandBldg_W
X.individual$inf_UC <- X.individual$Urban_Control * coef$Urban_Control
X.individual$inf_bcr <- X.individual$max.building.coverage.ratio * coef$max.building.coverage.ratio
X.individual$inf_far <- X.individual$max.floor.area.ratio * coef$max.floor.area.ratio
X.individual$inf_city <- X.individual$City_Fukuoka * coef$City_Fukuoka +
  X.individual$City_Nagoya * coef$City_Nagoya +
  X.individual$City_Osaka * coef$City_Osaka +
  X.individual$City_Sapporo * coef$City_Sapporo
X.individual$inf_CPI <- X.individual$log.CPI * coef$log.CPI
X.individual$inf_GDP <- X.individual$log.nGDP * coef$log.nGDP_ns
```

```
X.individual$inf_immi <- X.individual$PctImmi * coef$PctImmi
X.individual$inf_crime <- X.individual$Ncrime * coef$Ncrime
X.individual$inf_unemp <- X.individual$PctUnemploy * coef$PctUnemploy
X.individual$inf_exec <- X.individual$PctExec * coef$PctExec
X.individual$inf_lr4555 <- X.individual$JSHIS_I45_55 * coef$JSHIS_I45_55
X.individual$inf_lr55 <- X.individual$JSHIS_I55 * coef$JSHIS_I55
X.individual$inf_sr <- X.individual$Xpsi * coef$Xpsi
X.individual$inf_total <- abs(X.individual$inf_constant) + abs(X.individual$inf_city) +
  abs(X.individual$inf_immi) + abs(X.individual$inf_crime) + abs(X.individual$inf_unemp) +
  abs(X.individual$inf_exec) + abs(X.individual$inf_GDP) + abs(X.individual$inf_CPI) +
  abs(X.individual$inf_area) + abs(X.individual$inf_floorarea) + abs(X.individual$inf_distance) + abs(X
  abs(X.individual$inf_bldgyr) + abs(X.individual$inf_bldgstructure) + abs(X.individual$inf_UC) +
  abs(X.individual$inf_bcr) + abs(X.individual$inf_far) + abs(X.individual$inf_lr4555) + abs(X.individua
  abs(X.individual$inf_sr)
X.individual$pct_intercept <- abs(X.individual$inf_constant) / X.individual$inf_total
X.individual$pct_city <- abs(X.individual$inf_city) / X.individual$inf_total
X.individual$pct_int_city <- X.individual$pct_intercept + X.individual$pct_city
X.individual$pct_ward <- (abs(X.individual$inf_immi) + abs(X.individual$inf_crime) + abs(X.individual$in
                            abs(X.individual$inf_exec)) / X.individual$inf_total
X.individual$pct_macro <- (abs(X.individual$inf_GDP) + abs(X.individual$inf_CPI)) / X.individual$inf_to
X.individual$pct_prop_basic <- ( abs(X.individual$inf_area) + abs(X.individual$inf_floorarea) +
                                   abs(X.individual$inf_distance) + abs(X.individual$inf_age)) / X.indiv
X.individual$pct_prop_ext <- ( abs(X.individual$inf_bldgyr) + abs(X.individual$inf_bldgstructure) + abs
                                 abs(X.individual$inf_bcr) + abs(X.individual$inf_far) ) / X.individual$
X.individual$pct_prop_all <- X.individual$pct_prop_basic + X.individual$pct_prop_ext
X.individual$pct_lr <- (abs(X.individual$inf_lr4555) + abs(X.individual$inf_lr55)) / X.individual$inf_to
X.individual$pct_sr <- abs(X.individual$inf_sr) / X.individual$inf_total


X.individual$prediction <- X.individual$Type_LandBldg * coef$constant_LandBldg +
  X.individual$Type_LandOnly * coef$constant_LandOnly +
  X.individual$Type_Condo * coef$constant_Condo + X.individual$City_Fukuoka * coef$City_Fukuoka +
  X.individual$City_Nagoya * coef$City_Nagoya +
  X.individual$City_Osaka * coef$City_Osaka +
  X.individual$City_Sapporo * coef$City_Sapporo +
  X.individual$PctImmi * coef$PctImmi +
  X.individual$Ncrime * coef$Ncrime +
  X.individual$PctUnemploy * coef$PctUnemploy +
  X.individual$PctExec * coef$PctExec +
  X.individual$log.CPI * coef$log.CPI +
  X.individual$log.nGDP * coef$log.nGDP_ns + X.individual$distance.num * coef$distance.num +
  X.individual$area.m2.num * coef$area.m2.num +
  X.individual$total.floor.area.m2.num * coef$total.floor.area.m2.num +
  X.individual$building.age * coef$building.age +
  X.individual$built.1981_2000 * coef$built.1981_2000 +
  X.individual$built.after2000 * coef$built.after2000 +
  X.individual$LandBldg_RC * coef$LandBldg_RC +
  X.individual$LandBldg_S * coef$LandBldg_S +
  X.individual$LandBldg_W * coef$LandBldg_W +
  X.individual$Urban_Control * coef$Urban_Control +
  X.individual$max.building.coverage.ratio * coef$max.building.coverage.ratio +
  X.individual$max.floor.area.ratio * coef$max.floor.area.ratio +
  X.individual$JSHIS_I45_55 * coef$JSHIS_I45_55 +
```

```
  X.individual$JSHIS_I55 * coef$JSHIS_I55 +
  X.individual$Xpsi * coef$Xpsi
X.individual$error <- X.individual$log.price - X.individual$prediction



sumstat.inf <- X.individual %>%
    group_by("Type") %>%
    summarise( intercept= quantile(pct_int_city, 0.75)-quantile(pct_int_city, 0.25),
                        ward = quantile(pct_ward, 0.75)-quantile(pct_ward, 0.25),
                        macro = quantile(pct_macro, 0.75)-quantile(pct_macro, 0.25),
                        property = quantile(pct_prop_all, 0.75)-quantile(pct_prop_all, 0.25),
                        longrun = quantile(pct_lr, 0.75)-quantile(pct_lr, 0.25),
                        shortrun = quantile(pct_sr, 0.75)-quantile(pct_sr, 0.25) )

kable(sumstat.inf, caption = "Influences of each component for each type, real price", digits = 4)
```

```
# risk premia
X.individual$m0 <- X.individual$Type_LandBldg * coef$constant_LandBldg +
    X.individual$Type_LandOnly * coef$constant_LandOnly +
    X.individual$Type_Condo * coef$constant_Condo + X.individual$City_Fukuoka * coef$City_Fukuoka +
    X.individual$City_Nagoya * coef$City_Nagoya +
    X.individual$City_Osaka * coef$City_Osaka +
    X.individual$City_Sapporo * coef$City_Sapporo +
    X.individual$PctImmi * coef$PctImmi +
    X.individual$Ncrime * coef$Ncrime +
    X.individual$PctUnemploy * coef$PctUnemploy +
    X.individual$PctExec * coef$PctExec +
    X.individual$log.CPI * coef$log.CPI +
    X.individual$log.nGDP * coef$log.nGDP_ns + X.individual$distance.num * coef$distance.num +
    X.individual$area.m2.num * coef$area.m2.num +
    X.individual$total.floor.area.m2.num * coef$total.floor.area.m2.num +
    X.individual$building.age * coef$building.age +
    X.individual$built.1981_2000 * coef$built.1981_2000 +
    X.individual$built.after2000 * coef$built.after2000 +
    X.individual$LandBldg_RC * coef$LandBldg_RC +
    X.individual$LandBldg_S * coef$LandBldg_S +
    X.individual$LandBldg_W * coef$LandBldg_W +
    X.individual$Urban_Control * coef$Urban_Control +
    X.individual$max.building.coverage.ratio * coef$max.building.coverage.ratio +
    X.individual$max.floor.area.ratio * coef$max.floor.area.ratio

X.individual$m1 <- X.individual$Type_LandBldg * coef$constant_LandBldg +
    X.individual$Type_LandOnly * coef$constant_LandOnly +
    X.individual$Type_Condo * coef$constant_Condo + X.individual$City_Fukuoka * coef$City_Fukuoka +
    X.individual$City_Nagoya * coef$City_Nagoya +
    X.individual$City_Osaka * coef$City_Osaka +
    X.individual$City_Sapporo * coef$City_Sapporo +
    X.individual$PctImmi * coef$PctImmi +
    X.individual$Ncrime * coef$Ncrime +
    X.individual$PctUnemploy * coef$PctUnemploy +
    X.individual$PctExec * coef$PctExec +
    X.individual$log.CPI * coef$log.CPI +
    X.individual$log.nGDP * coef$log.nGDP_ns + X.individual$distance.num * coef$distance.num +
```

```
    X.individual$area.m2.num * coef$area.m2.num +
    X.individual$total.floor.area.m2.num * coef$total.floor.area.m2.num +
    X.individual$building.age * coef$building.age +
    X.individual$built.1981_2000 * coef$built.1981_2000 +
    X.individual$built.after2000 * coef$built.after2000 +
    X.individual$LandBldg_RC * coef$LandBldg_RC +
    X.individual$LandBldg_S * coef$LandBldg_S +
    X.individual$LandBldg_W * coef$LandBldg_W +
    X.individual$Urban_Control * coef$Urban_Control +
    X.individual$max.building.coverage.ratio * coef$max.building.coverage.ratio +
    X.individual$max.floor.area.ratio * coef$max.floor.area.ratio +
    X.individual$JSHIS_I45_55 * coef$JSHIS_I45_55 +
    X.individual$JSHIS_I55 * coef$JSHIS_I55

X.individual$m2 <- X.individual$Type_LandBldg * coef$constant_LandBldg +
    X.individual$Type_LandOnly * coef$constant_LandOnly +
    X.individual$Type_Condo * coef$constant_Condo + X.individual$City_Fukuoka * coef$City_Fukuoka +
    X.individual$City_Nagoya * coef$City_Nagoya +
    X.individual$City_Osaka * coef$City_Osaka +
    X.individual$City_Sapporo * coef$City_Sapporo +
    X.individual$PctImmi * coef$PctImmi +
    X.individual$Ncrime * coef$Ncrime +
    X.individual$PctUnemploy * coef$PctUnemploy +
    X.individual$PctExec * coef$PctExec +
    X.individual$log.CPI * coef$log.CPI +
    X.individual$log.nGDP * coef$log.nGDP_ns + X.individual$distance.num * coef$distance.num +
    X.individual$area.m2.num * coef$area.m2.num +
    X.individual$total.floor.area.m2.num * coef$total.floor.area.m2.num +
    X.individual$building.age * coef$building.age +
    X.individual$built.1981_2000 * coef$built.1981_2000 +
    X.individual$built.after2000 * coef$built.after2000 +
    X.individual$LandBldg_RC * coef$LandBldg_RC +
    X.individual$LandBldg_S * coef$LandBldg_S +
    X.individual$LandBldg_W * coef$LandBldg_W +
    X.individual$Urban_Control * coef$Urban_Control +
    X.individual$max.building.coverage.ratio * coef$max.building.coverage.ratio +
    X.individual$max.floor.area.ratio * coef$max.floor.area.ratio +
    X.individual$JSHIS_I45_55 * coef$JSHIS_I45_55 +
    X.individual$JSHIS_I55 * coef$JSHIS_I55 +
    X.individual$Xpsi_obj * coef$Xpsi

X.individual$m3 <- X.individual$prediction


sumstat.rp <- X.individual %>%
  dplyr:::group_by_("Type", "City") %>%
  dplyr:::summarise(med.log.price = median(log.price),
                    pred.m0 = median(m0),
                    pred.m1 = median(m1),
                    pred.m2 = median(m2),
                    pred.m3 = median(m3))
sumstat.rp$med.lr <- sumstat.rp$pred.m1 - sumstat.rp$pred.m0
sumstat.rp$med.sr.obj <- sumstat.rp$pred.m2 - sumstat.rp$pred.m1
```

```
sumstat.rp$med.sr.sub <- sumstat.rp$pred.m3 - sumstat.rp$pred.m2
```

Sensitivity analysis

Sensitivity to ward and economic indicators

The following code can be used to replicate Table 6 of the paper. Note that as in the section "Main estimation results", the final estimated ψ is given in the following functions. In order to replicate the process of using grid search to arrive at the optimal ψ, use the wrapper function *opt_psi*. Note that for each value of ψ, the estimation takes aroun 2 hours.

```
Xnames_Attr <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                 "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "PctForeign", "Nhosp", "Ndaycare", "Nkindergtn",
                 "Nagedhome","Ndepstore", "Nlargeretail",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")
Xnames_noGDP <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                  "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")

results_attr <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.75, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_Attr,
                    par.include = include_2error,
                    par.init = initpar)

results_noGDP <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 2.63, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_noGDP,
                    par.include = include_2error,
                    par.init = initpar)
```

Sensitivity to property characteristics

The following code can be used to replicate Table 7 of the paper.

```
Xnames_noUrbanControl <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                           "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")
Xnames_noBldgStr <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                      "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")
Xnames_LandUse <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                    "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                 "LU_Resid", "LU_Comm", "LU_Industr",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")

results_UC <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.72, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_noUrbanControl,
                    par.include = include_2error,
                    par.init = initpar)
results_BS <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.89, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_noBldgStr,
                    par.include = include_2error,
                    par.init = initpar)
results_LandUse <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                      psi = 3.76, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_LandUse,
                    par.include = include_2error,
```

```
                    par.init = initpar)
```

Sensitivity to removing one of the cities

The following code can be used to replicate Table 8 of the paper.

```
# generate averages of y, X and cell counts H, removing Tokyo
data_vec_noTokyo <- mvecr::vectorize(
  data = individual_data %>%
    filter(City_Tokyo==0),
   colName.i = colName.i,
   colName.t = colName.t, colName.p = colName.p,
   colName.y = "log.price",
   colName.X = Xnames_all)
# retrieve each component of the results
H_noTokyo <- data_vec_noTokyo$H
y_noTokyo <- data_vec_noTokyo$y
X_noTokyo <- data_vec_noTokyo$X
district_noTokyo <- data_vec_noTokyo$district
time_noTokyo <- data_vec_noTokyo$time
type_noTokyo <- data_vec_noTokyo$type

# generate averages of y, X and cell counts H, removing Osaka
data_vec_noOsaka <- mvecr::vectorize(
  data = individual_data %>%
    filter(City_Osaka==0),
  colName.i = colName.i,
  colName.t = colName.t, colName.p = colName.p,
  colName.y = "log.price",
  colName.X = Xnames_all)
# retrieve each component of the results
H_noOsaka <- data_vec_noOsaka$H
y_noOsaka <- data_vec_noOsaka$y
X_noOsaka <- data_vec_noOsaka$X
district_noOsaka <- data_vec_noOsaka$district
time_noOsaka <- data_vec_noOsaka$time
type_noOsaka <- data_vec_noOsaka$type

# generate averages of y, X and cell counts H, removing Nagoya
data_vec_noNagoya <- mvecr::vectorize(
  data = individual_data %>%
    filter(City_Nagoya==0),
  colName.i = colName.i,
  colName.t = colName.t, colName.p = colName.p,
  colName.y = "log.price",
  colName.X = Xnames_all)
# retrieve each component of the results
H_noNagoya <- data_vec_noNagoya$H
y_noNagoya <- data_vec_noNagoya$y
X_noNagoya <- data_vec_noNagoya$X
district_noNagoya <- data_vec_noNagoya$district
time_noNagoya <- data_vec_noNagoya$time
type_noNagoya <- data_vec_noNagoya$type
```

```
results_noTokyo <- mvecr::reg_psi(data_X = X_noTokyo,
                                  data_y = y_noTokyo, data_H = H_noTokyo,
                    psi = 1.9, method = "p",
                    district = district_noTokyo,
                    time = time_noTokyo,
                    type = type_noTokyo,
                    var = setdiff(Xnames_base, "City_Osaka"),
                    par.include = include_2error,
                    par.init = initpar)
results_noNagoya <- mvecr::reg_psi(data_X = X_noNagoya,
                                   data_y = y_noNagoya,
                                   data_H = H_noNagoya,
                    psi = 4.11, method = "p",
                    district = district_noNagoya,
                    time = time_noNagoya,
                    type = type_noNagoya,
                    var = setdiff(Xnames_base, "City_Nagoya"),
                    par.include = include_2error,
                    par.init = initpar)
results_noOsaka <- mvecr::reg_psi(data_X = X_noOsaka,
                                  data_y = y_noOsaka,
                                  data_H = H_noOsaka,
                    psi = 4.04, method = "p",
                    district = district_noOsaka,
                    time = time_noOsaka,
                    type = type_noOsaka,
                    var = setdiff(Xnames_base, "City_Osaka"),
                    par.include = include_2error,
                    par.init = initpar)
```

Sensitivity to quarter dummies

The following code can be used to replicate Table 9 of the paper.

```
Xnames_Q123 <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                 "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
                 "Urban_Control",
                 "Q1", "Q2", "Q3",
                 "max.building.coverage.ratio", "max.floor.area.ratio",
                 "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                 "log.nGDP", "log.CPI",
                 "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                 "JSHIS_I45_55", "JSHIS_I55", "Xpsi")
Xnames_Q4 <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
               "distance.num", "area.m2.num", "total.floor.area.m2.num",
                 "building.age",
                 "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                 "built.1981_2000", "built.after2000" ,
```

```
                  "Urban_Control",
                  "Q4",
                  "max.building.coverage.ratio", "max.floor.area.ratio",
                  "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                  "log.nGDP", "log.CPI",
                  "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                  "JSHIS_I45_55", "JSHIS_I55", "Xpsi")
Xnames_Tohoku <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                    "distance.num", "area.m2.num", "total.floor.area.m2.num",
                "building.age",
                "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                "built.1981_2000", "built.after2000" ,
                "Urban_Control",
                "Q_after_Fukushima",
                "max.building.coverage.ratio", "max.floor.area.ratio",
                "City_Fukuoka", "City_Nagoya", "City_Osaka", "City_Sapporo",
                "log.nGDP", "log.CPI",
                "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                "JSHIS_I45_55", "JSHIS_I55", "Xpsi")

results_Q123 <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 4.56, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_Q123,
                    par.include = include_2error,
                    par.init = initpar)
results_Q4 <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.89, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_Q4,
                    par.include = include_2error,
                    par.init = initpar)
results_Tohoku <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.27, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_Tohoku,
                    par.include = include_2error,
                    par.init = initpar)
```

Sensitivity to stochasticity and station versus district

The following code can be used to replicate Table 10 of the paper.

```
include_3error <- rep(1, 18)
results_3error <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 3.52, method = "p",
                    district = district, time = time, type = type,
                    var = Xnames_base,
                    par.include = include_3error,
                    par.init = initpar)
```

```r
data_vec_station <- mvecr::vectorize(data = individual_data,
                             colName.i = "Nearest.station.Name",
                 colName.t = "t", colName.p = "Type",
                 colName.y = "log.price",
                 colName.X = Xnames_all)
# retrieve each component of the results
H_station <- data_vec_station$H
y_station <- data_vec_station$y
X_station <- data_vec_station$X
station <- data_vec_station$district
time_station <- data_vec_station$time
type_station <- data_vec_station$type


Xnames_station <- c("constant_LandBldg", "constant_LandOnly", "constant_Condo",
                   "distance.num", "area.m2.num", "total.floor.area.m2.num",
                   "building.age",
                   "LandBldg_RC", "LandBldg_S", "LandBldg_W",
                   "built.1981_2000", "built.after2000" ,
                   "Urban_Control",
                   "Q_after_Fukushima",
                   "max.building.coverage.ratio", "max.floor.area.ratio",
                   "City_Fukuoka", "City_Nagoya", "City_Osaka",
                   "City_Sapporo",
                   "log.nGDP", "log.CPI",
                   "PctImmi", "Ncrime", "PctUnemploy", "PctExec",
                   "JSHIS_I45_55_station", "JSHIS_I55_station", "Xpsi")
results_station <- mvecr::reg_psi(data_X = X_station, data_y = y_station,
                                  data_H = H_station,
                    psi = 3.56, method = "p",
                    district = station, time = time_station,
                    type = type_station,
                    var = Xnames_base,
                    par.include = include_2error,
                    par.init = initpar)
```

Sensitivity to weighting functions and extension

The following code can be used to replicate Table 11 of the paper.

```r
results_SR_TK <- mvecr::reg_psi(data_X = X, data_y = y, data_H = H,
                    psi = 1.40, method = "t",
                    district = district, time = time, type = type,
                    var = Xnames_base,
                    par.include = include_2error,
                    par.init = initpar)

results_LR_TK
resukts_LR_Prelec
```

The following code can be used to replicate Figure 4 of the paper.

```r
par(xpd=T, xaxs='i',yaxs='i', pty='s')
psi <- 3.77
gamma <- 0.32
plot(p, tversky(p, gamma), type = "l", lty=2, xlab='objective probabilities', ylim = c(0,1), xlim=c(0,1
     ylab = "distorted probabilities", lwd=0.8, col='blue',  asp=1, xaxt='n', yaxt='n')
axis(2, labels = c(0,0.2,0.4,0.6,0.8,1), at=c(0,0.2,0.4,0.6,0.8,1))
axis(1, labels = c(0,0.2,0.4,0.6,0.8,1), at=c(0,0.2,0.4,0.6,0.8,1))
lines(p, prelec(p, 1), lty=3, lwd=0.5, col='grey')
lines(p, prelec(p, psi), lty=1, lwd=0.8, col='black')
legend(x=0.35, y=0.9,  legend = c("short run", "long run"),
       #text.width = strwidth("100"),
       col=c('black', 'blue'),
       lty = c(1:2) , xjust = 1, yjust = 1, ncol = 1)
```