Replication Instructions

Masako Ikefuji, Roger J. A. Laeven, Jan R. Magnus, Yuan Yue

June 30, 2020

This document contains instructions to replicate all the estimation procedures, tables and graphs in the paper: *Earthquake risk embedded in property prices: Evidence from five Japanese cities.*

The code consists of two major parts:

A. Estimation and simulation of the ETAS model

- Estimation of the ETAS model, replication of the summary statistics and estimation results in Tables 48 and 50 of the Data Documentation;
- Simulation of short-run earthquake probabilities, using the estimated ETAS parameters and historical earthquake catalog. Replication of Figures 1 and 2 of the paper;

and

B. Estimation of the multivariate error components regression model

- Characteristics of the housing dataset. Replication of Table 1 of the paper;
- Characteristics of the JSHIS long run probabilities. Replication of Table 2 of the paper;
- Main estimation results. Replication of Table 3 and Figure 3 of the paper;
- Sensitivity analysis regarding probability weighting functions. Replication of Table 4 and Figure 4 of the paper;
- Sensitivity analysis regarding other model specifications. Replication of Tables B1 - B5 of the supplementary material;
- Importance ordering and decomposition of risk premia. Replication of Tables C6 - C7 of the supplementary material.

These two parts can be executed independently of each other. The functions related to the estimation and simulation of the ETAS model are contained in the file `etas_funcs.R`. The functions related to the estimation of the multivariate error components regression model are contained in the `R` package `mvecr`.

Download and installation instructions

- Download and install the latest version of Rtools from https://cran.r-project.org/bin/windows/Rtools/.
- After the installation is complete, put the location of the Rtools *make* utilities (bash, make, etc) on the PATH by executing the following command in `R` (for Rtools 40 as an example):

```
writeLines('PATH="${RTOOLS40_HOME}\\usr\\bin;${PATH}"', con = "~/.Renviron")
```

- Restart `R` and verify that the *make* function can be found by executing the command:

```
Sys.which("make")
## "C:\\rtools40\\usr\\bin\\make.exe"
```

- Create a directory called "earthquake-risk" and set it as the current working directory in `R`.
- Download the compressed file for `R` package `mvecr` ("mvecr_0.3.0.tar.gz") 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”, “city_range.csv”, and “JMA_records.csv” into this subdirectory.
- Within the current working directory, create a subdirectory called “output”.
- Use the following `R` code to install all relevant packages and read data.

```
install.packages("PtProcess")
install.packages("dplyr")
install.packages("readr")
install.packages("zoo")
install.packages("R.utils")
install.packages("mvecr_0.3.0.tar.gz", repos = NULL, type = "source")

library(PtProcess)
library(dplyr)
library(mvecr)
library(readr)
library(zoo)
library(R.utils)

path <- "./earthquake-risk"

source('etas_funcs.R')


# 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)
jma_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 our housing sample from 2006Q2 - 2015Q3. Each row represents one housing transaction record, containing information on (the natural logarithm of) total transaction price of a residential property, transaction period, housing type, city, district(area), ward, name of the nearest station, distance to the 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 spatial 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. Each row contains an earthquake record with relevant information. 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 output of the ETAS models. For each city in our study, the ETAS model is estimated and simulated. The simulated probability of having an earthquake exceeding the magnitude threshold 5.5 in each quarter is recorded in this dataset. Each column corresponds to one of the five cities included in the scope of our analysis, and each row is the simulated quarterly short-run earthquake probability.

Main user-facing functions

ETAS estimation and simulation

The main functions included in `etas_func.R` are

- `EQcatalog`: this function generates an earthquake catalog in the format that can be used for the estimation of ETAS models;
- `etas_estim`: this function estimates the parameters of the ETAS model using the `PtProcess` package;
- `etas_prob`: this function calculates the earthquake probability forecasts within a given time period through simulation using the `PtProcess` package.
- `gen_Xpsi_city`: this function generates the simulated earthquake probabilities for each city for the entire sample period (2006Q2 - 2015Q3) using `etas_prob`.

Multivariate error components regression

The main user-facing functions of our code are the `vectorize`, `ec_reg`, `reg_psi`, and `opt_psi` functions in the `mvecr` package.

- `vectorize` takes the individual records as input, groups them by the specified "time", "district", and "type" dimensions, takes the averages within each group, and stacks the results into vectors and matrices that can be used in the multiple error components regression;

- `ec_reg` takes the "vectorized" data and performs maximum likelihood estimation of the multivariate error components model;
- `reg_psi` takes a step further and allows one of the regressors to be transformed by a single-parameter function, with a given parameter ψ;
- `opt_psi` is a wrapper function around `reg_psi` that implements a grid search to find the optimal ψ. Given a list of candidate parameters ψ, it calls the function `reg_psi` for each value of ψ on the list and records the loglikelihood value. The parameter corresponding to the highest value of loglikelihood is chosen as $\hat{\psi}$.

Replication of the tables and figures

Estimation of ETAS model

The following code will be used to replicate the results of the ETAS model estimation. (As shown in Table 48 and Table 50 of *Earthquake Risk Embedded in Property Prices: Evidence from Five Japanese Cities - Data Documentation*). The estimation takes less than 1 minute.

```
# estimation of ETAS model for each city
# preparation of the earthquake catalog
eq_catalog <- EQcatalog(jma_data, t_start = 1970, t_end = 2015, depth = 100, mag_min = 4,
                origin = as.Date("1970-01-01"))
# setting magnitude thresholds for each city
magMin <- c(4.5, 4.5, 4.5, 4.5, 5)
# starting value of the parameters
init.params <- c(.1, .1, .5, 1.2, 1.1, 1/mean(eq_catalog$magnitude), 0)
list.Est <- {}
# estimate the ETAS model for each city
sum.table <- data.frame(city = city_range$City, stringsAsFactors = F)
for(i in 1:length(city_range$City)){
  city <- city_range$City[i]
  out <- etas_estim(eq_catalog, city = city, city_range = city_range,
                 magMin=magMin[i], params=init.params,
                 t0 = as.Date("1970-01-01"), tN = "2016-01-01")
  list.Est[[i]] <- out
  ks <- etas_test(out, plot = F)
  sum.table$ks.pval[i] <- ks$p.val
  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) %>%
      arrange(factor(City, levels = c("Tokyo", "Osaka",
```

```
                                        "Nagoya", "Fukuoka", "Sapporo"))),
      caption = "Spatial window of the earthquake catalog",
      format='latex', booktabs=TRUE) %>%
    kable_styling(latex_options=c("HOLD_position"))
```

Table 1: Spatial window of the earthquake catalog

City	latMin	latMax	lngMin	lngMax
Tokyo	34.0	37.0	138.0	141.0
Osaka	33.5	36.5	134.0	137.0
Nagoya	33.5	36.5	135.5	138.5
Fukuoka	32.0	35.0	129.0	132.0
Sapporo	41.5	45.5	138.5	143.5

```
kable(sum.table %>%
      arrange(factor(city, levels = c("Tokyo", "Osaka",
                                      "Nagoya", "Fukuoka", "Sapporo"))),
      caption = "ETAS estimation results for each city (estimated with magnitude
      threshold 4.5 for Osaka, Nagoya, Fukuoka, Sapporo and 5 for Tokyo)",
      digits=4, format='latex', booktabs=TRUE) %>%
    kable_styling(latex_options=c("HOLD_position"))
```

Table 2: ETAS estimation results for each city (estimated with magnitude threshold 4.5 for Osaka, Nagoya, Fukuoka, Sapporo and 5 for Tokyo)

city	ks.pval	N	mu	K	C	p	beta
Tokyo	0.7639	370	0.0076	0.0351	0.0155	1.0072	0.9253
Osaka	0.7382	154	0.0073	0.0014	0.0064	1.1913	2.4854
Nagoya	0.9396	177	0.0071	0.0066	0.0009	0.9616	1.7250
Fukuoka	0.9817	102	0.0037	0.0098	0.0035	1.0330	1.4390
Sapporo	0.8738	486	0.0193	0.0039	0.1044	1.2091	2.4424

Simulation of short-run earthquake probabilities

The following code can be used to simulate the short-run earthquake probabilities. We used 30,000 Monte Carlo runs to generate the final simulated probabilities, which takes about 24 hours for each city. The output of our simulation, "Xpsi-1.csv", is used as an input to the multivariate error components model for the rest of this paper. To continue the replication for this paper use the file "Xpsi-1.csv" in the "data" folder for the rest of this document.

```
# This file generates the additional vector of regressor XPsi, for given psi.
library(PtProcess)
library(R.utils)
library(zoo)
source('etas_funcs.R')

# input arguments: number of simulations, which city, etc.
psi <- 1
Nsim <- 30000
# i = 1, 2, 3, 4, 5 (index of the city)
```

```r
i <- 1

# 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]
origin <- as.Date("1970-01-01")
Xpsi <- matrix(0, ncol = Nsim/10, nrow = length(time.date))
tic <- Sys.time()

for(n in 1:(Nsim/10)){
  print(n)
  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 = 10)
  Xpsi[1:length(time.date), n] <- as.numeric(unlist(results$threshold_5.5))

}
Sys.time() - tic

Xpsi <- rowSums(Xpsi, na.rm = T)
# store the generated probabilities
# write.csv(Xpsi, paste("Xpsi", psi, city, Nsim, id, ".csv", sep="-"), row.names = F)
```

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

```r
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"

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, '(5)')
addlabel(labels.EQ[6], 0.55, 0.12, '(2)')
addlabel(labels.EQ[10], 0.82, 0.05, '(3)')
addlabel(labels.EQ[4], 0.55, 0.12, '(1)')
addlabel(labels.EQ[11], 0.73, 0.12, '(4)')
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)')
```

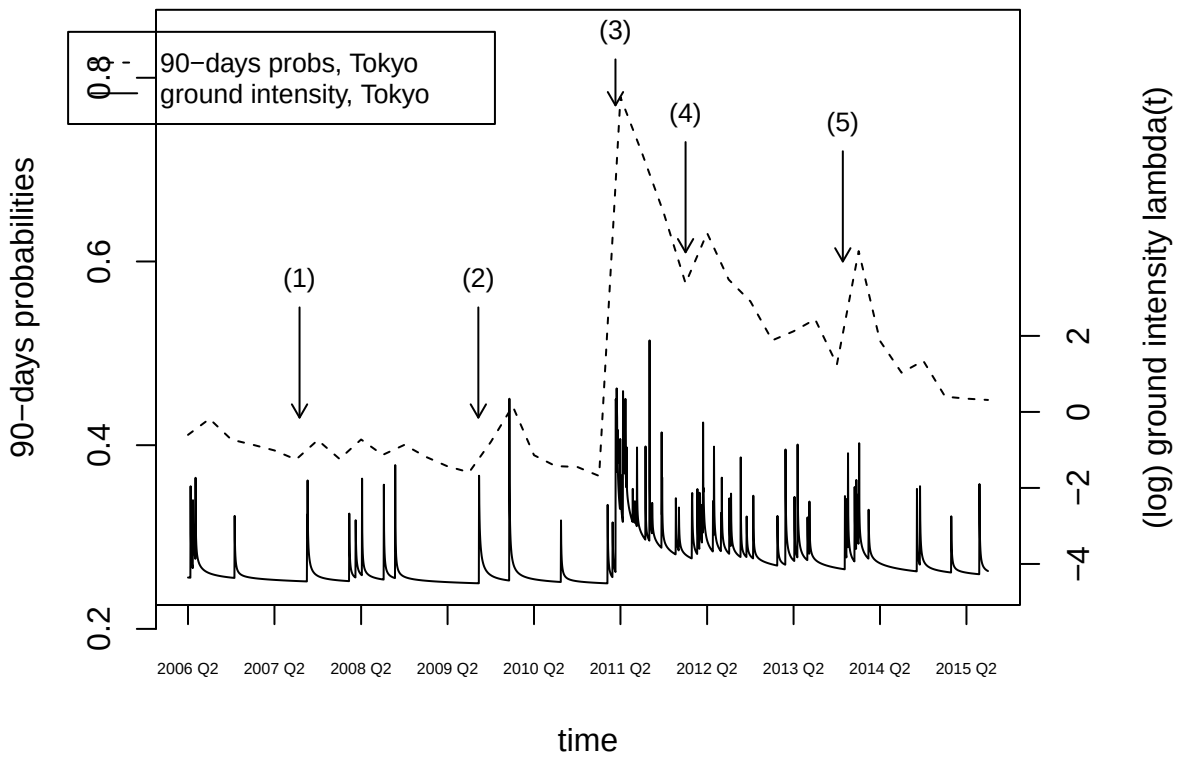

Figure 1: Short run earthquake risk for Tokyo

```
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
)
```

```r
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, '(1)', dis = 0.003)
addlabel(labels.EQ[7], 0.18, 0.01, '(2)', dis = 0.003)
addlabel(labels.EQ[10], 0.185, 0.008, '(3)', 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)')
```

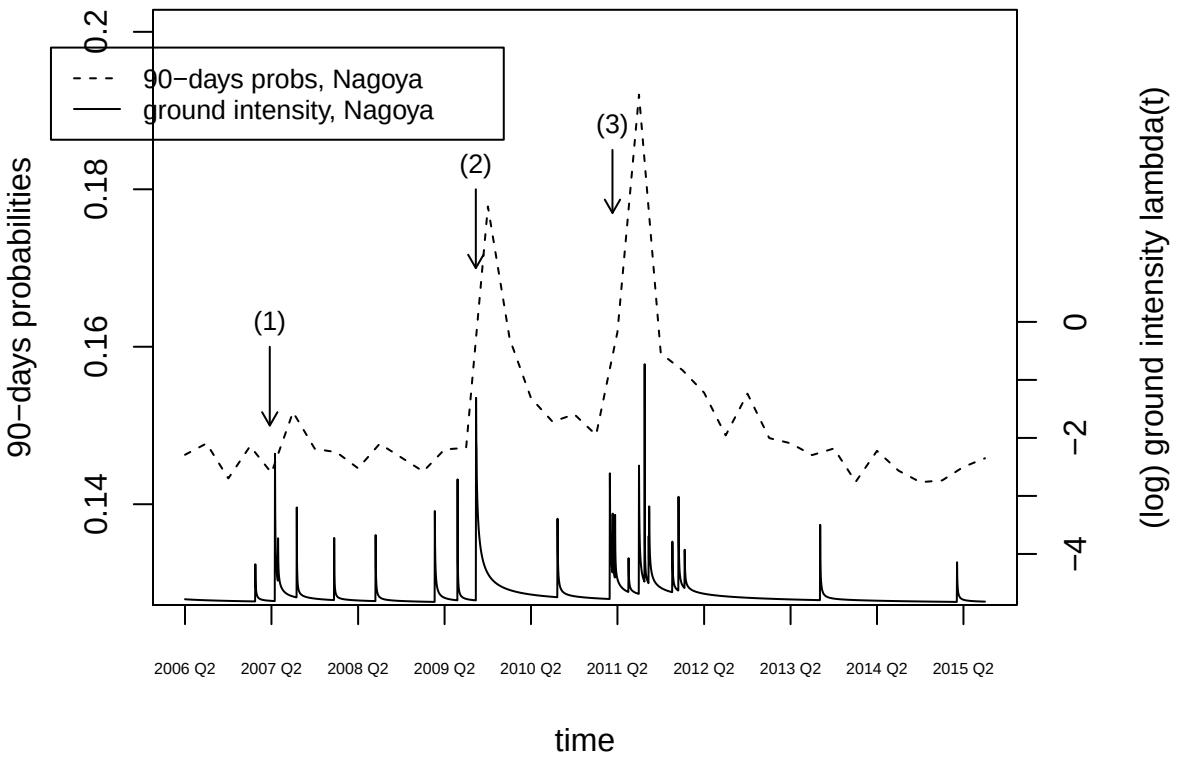

Figure 2: Short run earthquake risk for Nagoya

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))))
```

Table 3: Distribution of properties over cities, wards and districts

City	Ward	District	Land_building	Land_only	Condo	Station
Tokyo	23	898	57568	33991	92518	482
Osaka	24	564	21064	6901	21855	220
Nagoya	16	1379	14640	13110	11029	159
Fukuoka	7	318	7847	5660	12475	75
Sapporo	10	551	11763	9461	11461	86
Total	80	3710	112882	69123	149338	1022

Long-run earthquake intensities

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

```
# descriptive statistics of the JSHIS earthquake probabilities
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"
  )))



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"
```

```
))
```

Table 4: Seizmic hazard probabilities per city, exceeding intensity level 5 lower, averaged over districts and time 2005-2014

City	mean	min	q25	q50	q75	max	sd
Tokyo	0.9969	0.9860	0.9950	0.9978	0.9997	1.0000	0.0031
Osaka	0.9347	0.9011	0.9230	0.9391	0.9473	0.9664	0.0163
Nagoya	0.9613	0.9149	0.9434	0.9689	0.9774	0.9844	0.0179
Fukuoka	0.3916	0.0605	0.2966	0.4183	0.4794	0.5648	0.1164
Sapporo	0.3266	0.0457	0.2119	0.3332	0.4437	0.5056	0.1242

Table 5: Seizmic hazard probabilities per city, exceeding intensity level 6 lower, averaged over districts and time 2005-2014

City	mean	min	q25	q50	q75	max	sd
Tokyo	0.3452	0.1648	0.2228	0.2806	0.4884	0.5925	0.1322
Osaka	0.3722	0.2243	0.2954	0.3894	0.4404	0.5198	0.0862
Nagoya	0.5571	0.2086	0.4105	0.6136	0.6743	0.7730	0.1445
Fukuoka	0.0275	0.0035	0.0204	0.0295	0.0339	0.0452	0.0083
Sapporo	0.0145	0.0008	0.0070	0.0141	0.0216	0.0282	0.0078

Main estimation results

The following code can be used to replicate Table 3 of the paper. In the data preparation step, the `vectorize` function takes about 5 minutes to run for our dataset. For each model specification and each choice of ψ, the estimation takes around 1 - 1.5 hours.

`results_LRonly` (model with long run probabilities only), `results_LR_objSR` (model with long run probability and objective short run probability), and `results_base` (base model, with long run and short run probability) 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
# use 2-error error structure
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,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type",
  district = district, time = time, type = type,
  var = setdiff(Xnames_base, "Xpsi"),
  par.include = include_2error,
  par.init = initpar)
# write.csv(results_LRonly, paste0(path, '/output/results_long_run_only_model.csv'))
# 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,
  colName.i = "Area.Ward.City", colName.t = "t",
```

```
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 1,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = Xnames_base,
  par.include = include_2error,
  par.init = initpar)
# write.csv(results_LR_objSR, paste0(path, '/output/results_objective_short_run_model.csv'))

# results for the base model (model 3)
results_base <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.74,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = Xnames_base,
  par.include = include_2error,
  par.init = initpar)
# write.csv(results_base, paste0(path, '/output/results_base_model_psi3.74.csv'))
```

The results of the three models estimated using the code above can be used to replicate Table 3.

Table 6: Estimation results under various risk assumptions

var	LRonly_coef	LRonly_tstat	LR_objSR_coef	LR_objSR_tstat	base_coef	base_tstat
constant_LandBldg	3.7592	14.3462	4.5593	16.5082	4.3812	15.1935
constant_LandOnly	3.5949	13.7102	4.3940	15.9012	4.2155	14.6110
constant_Condo	3.1025	11.8396	3.9024	14.1297	3.7244	12.9151
City_Osaka	-0.2273	-41.1507	-0.2625	-40.2917	-0.2615	-41.8210
City_Nagoya	-0.3801	-94.8696	-0.4100	-79.1592	-0.4139	-83.6820
City_Fukuoka	-0.8770	-69.4800	-0.9133	-69.2411	-0.9108	-70.3421
City_Sapporo	-1.2050	-87.9341	-1.2458	-78.5258	-1.2388	-88.4784
PctImmi	6.7245	66.2047	6.7224	66.1855	6.7218	66.1778
Ncrime	-0.0437	-50.1872	-0.0436	-50.1411	-0.0436	-50.1568
PctUnemploy	-4.3360	-44.6225	-4.3395	-44.6623	-4.3399	-44.6669
PctExec	3.3426	37.6424	3.3447	37.6657	3.3464	37.6841
log.nGDP	0.5606	31.3612	0.5220	28.5674	0.5229	28.5039
log.CPI	1.5347	31.0184	1.4687	29.3539	1.5030	28.7177
area.m2.num	0.0025	401.7134	0.0025	401.8103	0.0025	401.8507
total.floor.area.m2.num	0.0006	107.9200	0.0006	107.8982	0.0006	107.8929
distance.num	-0.0145	-78.7586	-0.0145	-78.8349	-0.0145	-78.8931
building.age	-0.0121	-145.0400	-0.0121	-144.8179	-0.0121	-144.7190
built.1981_2000	0.1674	62.1468	0.1658	61.3993	0.1652	61.1469
built.after2000	0.4136	159.5462	0.4126	159.0240	0.4123	158.8869
LandBldg_RC	0.4348	45.8971	0.4344	45.8665	0.4343	45.8634
LandBldg_S	0.1867	21.3975	0.1867	21.4060	0.1867	21.4078
LandBldg_W	-0.1264	-15.9946	-0.1266	-16.0248	-0.1266	-16.0275
Urban_Control	-0.8972	-39.8213	-0.8967	-39.8140	-0.8967	-39.8164
max.building.coverage.ratio	-0.0019	-24.8178	-0.0019	-24.6324	-0.0019	-24.6438
max.floor.area.ratio	0.0004	36.6011	0.0004	36.5363	0.0004	36.5604
JSHIS_I45_55	-0.1433	-7.3644	-0.1427	-7.3362	-0.1427	-7.3337
JSHIS_I55	-0.5037	-24.3076	-0.5039	-24.3173	-0.5041	-24.3262
Xpsi	-	-	-0.0915	-1.9240	-0.0514	-9.1601
psi	-	-	1.0000	0.0000	3.74	2.9085
Delta logL	-68.47	-	-15.84	-	-	-

Note that in the base model (and in the sensitivity analyses in later sections) in order to find the value of ψ that maximizes likelihood, we perform a grid search over possible values of ψ 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 (computation time depends on the length of the list of candidate parameters. For each given value of ψ on the list, the program takes around 1 - 1.5 hours).

```
# list_psis <- seq(from = 0.5, to = 10, by = 0.5)
list_psis <- seq(from = 3.6, to = 3.8, by = 0.01)
psi_optim <- mvecr::opt_psi(
  data.X=X, data.y=y, data.H=H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  list.psi = list_psis,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = Xnames_base,
  par.include = include_2error,
  par.init = initpar)
# final estimate: psi_estimate = 3.74 maximizes likelihood
```

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

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

par(xpd=T, xaxs='i',yaxs='i', pty='s')
psi_estimate <- 3.74
plot(p, mvecr::prelec(p, psi_estimate), type = "l", lty=1,
     xlab='objective probabilities',
     ylab = "distorted probabilities",
     ylim = c(0,1), xlim=c(0,1),
     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, mvecr::prelec(p, 1), lty=3, lwd=0.5, col='grey')
```

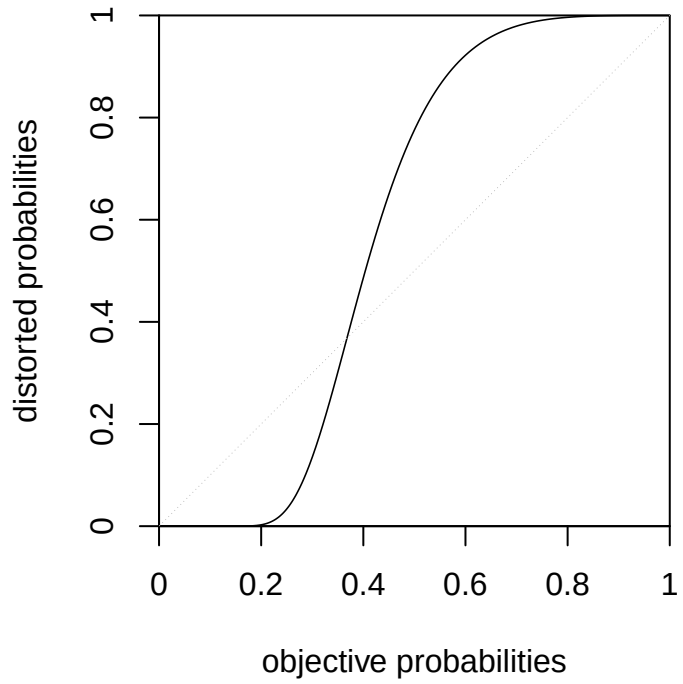

Figure 3: Estimated probability weighting of short-run probabilities, Prelec probability weighting function, psi= 3.74

Sensitivity analysis

Sensitivity to weighting functions and extension

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

```
results_SR_TK <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 1.40,
  transform.func = tversky, transform.gradient = Z_tversky,
  district = district, time = time, type = type,
  var = Xnames_base,
  par.include = include_2error,
  par.init = initpar)

# write.csv(results_SR_TK, paste0(path, '/output/results_SR_TK_psi1.40.csv'))


Xnames_LR <- c(setdiff(Xnames_base, c("JSHIS_I45_55", "JSHIS_I55")),
               "J45_55_sub", "J55_sub")
results_LR_prelec <- mvecr::reg_gamma_psi(data.X=X, data.y=y, data.H=H,
                            colName.i = colName.i,
                            colName.t = colName.t,
                            colName.p = colName.p,
                            gamma = 0.17, psi = 3.78,
                            method_1 = "p", method_2 = "p",
                            district, time, type,
                            var = Xnames_LR,
                            par.include = include_2error,
                            par.init = initpar)
# write.csv(results_LR_prelec, paste0(path, '/output/results_LR_prelec_psi3.78_gamma0.17.csv'))


resukts_LR_TK <- mvecr::reg_gamma_psi(data.X=X, data.y=y, data.H=H,
                            colName.i = colName.i,
                            colName.t = colName.t,
                            colName.p = colName.p,
                            gamma = 0.32, psi = 3.77,
                            method_1 = "t", method_2 = "p",
                            district, time, type,
                            var = Xnames_LR,
                            par.include = include_2error,
                            par.init = initpar)
# write.csv(resukts_LR_TK, paste0(path, '/output/results_LR_TK_psi3.77_gamma0.32.csv'))
```

Table 7: Sensitivity and extension: probability weighting function

var	base_coef	base_tstat	SR_TK_coef	SR_TK_tstat	LR_prelec_coef	LR_prelec_tstat	LR_TK_coef	LR_TK_tstat
area.m2.num	0.0025	401.8507	0.0025	401.8113	0.0025	402.2197	0.0025	402.2321
total.floor.area.m2.num	0.0006	107.8929	0.0006	107.8973	0.0006	107.9342	0.0006	107.9622
distance.num	-0.0145	-78.8931	-0.0145	-78.8399	-0.0143	-77.3433	-0.0143	-77.3533
building.age	-0.0121	-144.7190	-0.0121	-144.8278	-0.0121	-144.7442	-0.0121	-144.7110
JSHIS_I45_55	-0.1427	-7.3337	-0.1427	-7.3347	-0.8644	-17.2315	-0.4856	-15.2149
JSHIS_I55	-0.5041	-24.3262	-0.5039	-24.3156	-1.3838	-8.8647	-1.5028	-5.8513
Xpsi	-0.0514	-9.1601	-0.0733	-1.6737	-0.0517	-9.3097	-0.0518	-9.3094
psi	3.74	2.9085	1.40	0.2319	3.78	2.9498	3.77	2.9513
gamma	-	-	-	-	0.17	-37.0245	0.32	-42.3530
Delta logL	-	-	-14.62	-	152.83	-	167.6	-

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

```
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
)
```

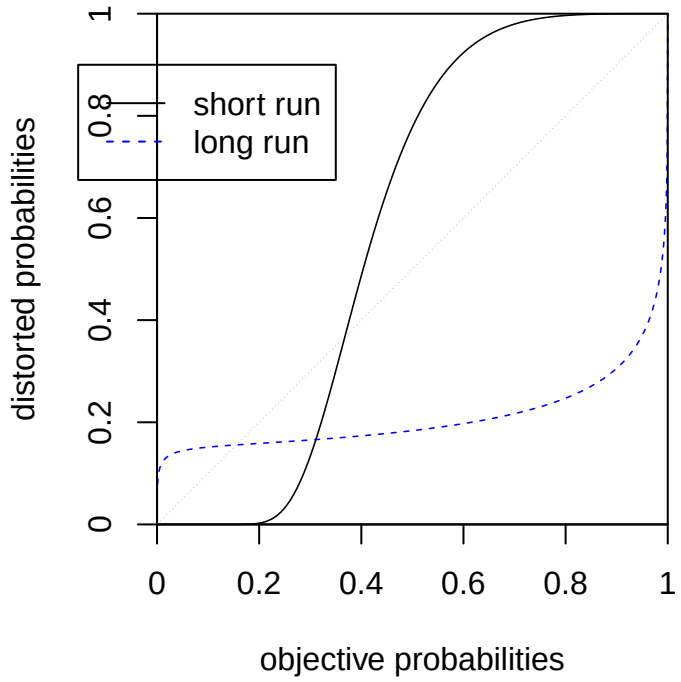

Figure 4: Implied probability weighting functions of long run and short run earthquake risk

Sensitivity to ward and economic indicators

The following code can be used to replicate Table B1 of the supplementary material. Note that as in the section "Main estimation results", the final estimated ψ will be obtained by 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*. For each value of ψ, the estimation takes around 1 - 1.5 hours.

```
Xnames_Attr <- c(Xnames_base, "PctForeign", "Nhosp", "Ndaycare",
                 "Nkindergtn", "Nagedhome", "Ndepstore", "Nlargeretail")

results_attr <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
```

```
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.75,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = Xnames_Attr,
  par.include = include_2error,
  par.init = initpar)
# write.csv(results_attr, paste0(path, '/output/results_attr_psi3.75.csv'))

results_noGDP <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 2.63,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = setdiff(Xnames_base, "log.nGDP"),
  par.include = include_2error,
  par.init = initpar)
# write.csv(results_noGDP, paste0(path, '/output/results_noGDP_psi2.63.csv'))
```

Table 8: Sensitivity - ward attractiveness and economic indicators

var	base_coef	base_tstat	attr_coef	attr_tstat	noGDP_coef	noGDP_tstat
area.m2.num	0.0025	401.8507	0.0025	402.0951	0.0025	401.1466
total.floor.area.m2.num	0.0006	107.8929	0.0006	107.9211	0.0006	107.8072
distance.num	-0.0145	-78.8931	-0.0142	-76.4888	-0.0145	-78.6780
building.age	-0.0121	-144.7190	-0.0121	-145.3161	-0.0122	-145.7632
JSHIS_I45_55	-0.1427	-7.3337	-0.1961	-9.9007	-0.1411	-7.2458
JSHIS_I55	-0.5041	-24.3262	-0.5706	-26.9086	-0.5024	-24.2217
Xpsi	-0.0514	-9.1601	-0.0519	-9.2744	-0.0839	-10.4488
psi	3.74	2.9085	3.75	2.9434	2.63	3.3506
Delta logL	-	-	472.92	-	-407.78	-

Sensitivity to property characteristics

The following code can be used to replicate Table B2 of the paper.

```
results_UC <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.72,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = setdiff(Xnames_base, "Urban_Control"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_UC, paste0(path, '/output/results_UC_psi3.72.csv'))
results_BS <- mvecr::reg_psi(
```

```
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.89,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = setdiff(
    Xnames_base, c("LandBldg_RC", "LandBldg_S", "LandBldg_W")),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_BS, paste0(path, '/output/results_BS_psi3.89.csv'))

results_LandUse <-
  mvecr::reg_psi(
    data.X = X, data.y = y, data.H = H,
    colName.i = "Area.Ward.City", colName.t = "t",
    colName.p = "Type", colName.Xpsi = "Xpsi_obj",
    psi = 3.76,
    transform.func = prelec, transform.gradient = Z_prelec,
    district = district, time = time, type = type,
    var = c(Xnames_base, "LU_Resid", "LU_Comm", "LU_Industr"),
    par.include = include_2error,
    par.init = initpar
  )
# write.csv(results_LandUse, paste0(path, '/output/results_LandUse_psi3.76.csv'))
```

Table 9: Sensitivity - property characteristics

var	base_coef	base_tstat	UC_coef	UC_tstat	BS_coef	BS_tstat	LandUse_coef	LandUse_tstat
area.m2.num	0.0025	401.8507	0.0025	398.5671	0.0025	406.2896	0.0025	401.2668
total.floor.area.m2.num	0.0006	107.8929	0.0006	108.8596	0.0009	180.2596	0.0006	107.9776
distance.num	-0.0145	-78.8931	-0.0147	-79.5118	-0.0159	-84.7027	-0.0146	-78.5225
building.age	-0.0121	-144.7190	-0.0121	-144.7181	-0.0119	-143.9166	-0.0121	-144.7344
JSHIS_I45_55	-0.1427	-7.3337	-0.1060	-5.4471	-0.1685	-8.5081	-0.1387	-7.1278
JSHIS_I55	-0.5041	-24.3262	-0.4661	-22.4812	-0.5263	-24.9262	-0.4767	-22.6954
Xpsi	-0.0514	-9.1601	-0.0516	-9.1380	-0.0508	-9.2312	-0.0515	-9.2248
psi	3.74	2.9085	3.72	2.9045	3.89	2.9078	3.76	2.9260
Delta logL	-	-	-786.62	-	-5824.44	-	33.89	-

Sensitivity to removing one of the cities

The following code can be used to replicate Table B3 of the supplementary material.

```
# 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
)
```

```r
# 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,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 1.9,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district_noTokyo,
  time = time_noTokyo,
```

```
  type = type_noTokyo,
  var = setdiff(Xnames_base, "City_Osaka"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_noTokyo, paste0(path, '/output/results_noTokyo_psi1.9.csv'))


results_noNagoya <- mvecr::reg_psi(
  data.X = X_noNagoya, data.y = y_noNagoya, data.H = H_noNagoya,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 4.11,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district_noNagoya,
  time = time_noNagoya,
  type = type_noNagoya,
  var = setdiff(Xnames_base, "City_Nagoya"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_noNagoya, paste0(path, '/output/results_noNagoya_psi4.11.csv'))

results_noOsaka <- mvecr::reg_psi(
  data.X = X_noOsaka, data.y = y_noOsaka, data.H = H_noOsaka,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 4.04,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district_noOsaka,
  time = time_noOsaka,
  type = type_noOsaka,
  var = setdiff(Xnames_base, "City_Osaka"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_noOsaka, paste0(path, '/output/results_noOsaka_psi4.04.csv'))
```

Table 10: Sensitivity - removing one city

var	noTokyo_coef	noTokyo_tstat	noOsaka_coef	noOsaka_tstat	noNagoya_coef	noNagoya_tstat
area.m2.num	0.0023	346.7182	0.0024	365.2707	0.0025	313.8618
total.floor.area.m2.num	0.0006	97.2300	0.0006	95.2889	0.0006	88.5313
distance.num	-0.0152	-77.6015	-0.0145	-78.2464	-0.0147	-58.2518
building.age	-0.0126	-118.8187	-0.0127	-142.6735	-0.0115	-110.3559
JSHIS_I45_55	-0.2427	-14.1219	-0.1124	-5.9960	-0.1571	-6.8878
JSHIS_I55	-0.4302	-21.6568	-0.4759	-22.4301	-0.6160	-23.7981
Xpsi	-0.1873	-0.5128	-0.0627	-12.1959	-0.0525	-8.5577
psi	1.90	0.4172	4.04	3.7967	4.11	2.6550

Sensitivity to quarter dummies

The following code can be used to replicate Table B4 of the supplementary material.

```r
results_Q123 <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 4.56,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = c(Xnames_base, "Q1", "Q2", "Q3"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_Q123, paste0(path, '/output/results_Q123_psi4.56.csv'))

results_Q4 <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.89,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = c(Xnames_base, "Q4"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_Q4, paste0(path, '/output/results_Q4_psi3.89.csv'))

results_Tohoku <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.27,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = c(Xnames_base, "Q_after_Fukushima"),
  par.include = include_2error,
  par.init = initpar
)
# write.csv(results_Tohoku, paste0(path, '/output/results_Tohoku_psi3.27.csv'))
```

Table 11: Sensitivity - quarters and Tohoku dummy

var	base_coef	base_tstat	Q123_coef	Q123_tstat	Q4_coef	Q4_tstat	Tohoku_coef	Tohoku_tstat
area.m2.num	0.0025	401.8507	0.0025	402.7649	0.0025	403.0047	0.0025	401.8897
total.floor.area.m2.num	0.0006	107.8929	0.0006	107.9723	0.0006	107.9659	0.0006	107.8849
distance.num	-0.0145	-78.8931	-0.0145	-79.0499	-0.0145	-79.0312	-0.0145	-78.9033
building.age	-0.0121	-144.7190	-0.0120	-144.4921	-0.0120	-144.5878	-0.0121	-144.6980
JSHIS_I45_55	-0.1427	-7.3337	-0.1415	-7.2955	-0.1406	-7.2460	-0.1426	-7.3318
JSHIS_I55	-0.5041	-24.3262	-0.5033	-24.3528	-0.5025	-24.3116	-0.5040	-24.3221
Xpsi	-0.0514	-9.1601	-0.0162	-3.3402	-0.0208	-3.8225	-0.0562	-8.1928
psi	3.74	2.9085	4.56	1.0219	3.89	1.2142	3.27	2.6694
Delta logL	-	-	1091.27	-	1007.79	-	6.31	-

Sensitivity to stochasticity and station versus district

The following code can be used to replicate Table B5 of the supplementary material. The regression involving 3 error components need around 3 - 4 hours to run. The regression with the station as district needs 40 - 60 minutes.

```
include_3error <- rep(1, 18)
results_3error <- mvecr::reg_psi(
  data.X = X, data.y = y, data.H = H,
  colName.i = "Area.Ward.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.52,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = district, time = time, type = type,
  var = Xnames_base,
  par.include = include_3error,
  par.init = initpar
)
# write.csv(results_3error, paste0(path, '/output/results_3error_psi3.52.csv'))


Xnames_station <- gsub("JSHIS_I55", "JSHIS_I55_station",
                       gsub("JSHIS_I45_55",
                       "JSHIS_I45_55_station",
                       Xnames_base))

data_vec_station <- mvecr::vectorize(
  data = individual_data,
  colName.i = "Station.City",
  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

results_station <- mvecr::reg_psi(
  data.X = X_station, data.y = y_station, data.H = H_station,
  colName.i = "Station.City", colName.t = "t",
  colName.p = "Type", colName.Xpsi = "Xpsi_obj",
  psi = 3.41,
  transform.func = prelec, transform.gradient = Z_prelec,
  district = station, time = time_station, type = type_station,
  var = Xnames_station,
  par.include = include_2error,
  par.init = initpar
)

# write.csv(results_station, paste0(path, '/output/results_station_psi3.41.csv'))
```

Table 12: Sensitivity - stochasticity and station versus district

var	base_coef	base_tstat	three_error_coef	three_error_tstat	station_coef	station_tstat
area.m2.num	0.0025	401.8507	0.0025	402.2596	0.0026	217.4550
total.floor.area.m2.num	0.0006	107.8929	0.0006	108.0428	0.0006	54.4706
distance.num	-0.0145	-78.8931	-0.0146	-79.0891	-0.0134	-34.5818
building.age	-0.0121	-144.7190	-0.0121	-145.1792	-0.0116	-80.6588
JSHIS_I45_55	-0.1427	-7.3337	-0.1448	-7.4577	-0.1386	-2.6339
JSHIS_I55	-0.5041	-24.3262	-0.5068	-24.4956	-0.5344	-10.0141
Xpsi	-0.0514	-9.1601	-0.0443	-7.0726	-0.0560	-6.1432
psi	3.74	2.9085	3.52	2.2666	3.41	1.9795
Delta logL	-	-	735.22	-		

The following code can be used to replicate the error matrices on page 9 of the supplementary material.

```
results_base <- read.csv(paste0(path, "/output/results_base_model_psi3.74.csv"),
                         stringsAsFactors = F)
results_3error <- read.csv(paste0(path, "/output/results_3error_psi3.52.csv"),
                           stringsAsFactors = F)

# two error components: sigma_zeta and sigma_epsilon
include_2error <- c(rep(1, 6), rep(0,6), rep(1,6))
# length of type
p <- 3
par_2error <- results_base$param_est[1:sum(include_2error==1)]
tmp <- matrix(0, p, p)
L.zeta <- tmp
L.zeta[lower.tri(tmp, diag = T)] <- par_2error[1:(p*(p+1)/2)]
diag(L.zeta) <- exp(diag(L.zeta))
sigmazeta_2error <-  tcrossprod(L.zeta)

L.eps <- tmp
L.eps[lower.tri(tmp, diag = T)] <- par_2error[(p*(p+1)/2+1):(2*(p*(p+1)/2))]
diag(L.eps) <- exp(diag(L.eps))
sigmaeps_2error <- tcrossprod(L.eps)

# three error components:  sigma_zeta, sigma_eta, and sigma_epsilon
include_3error <- rep(1, 18)
# length of type
p <- 3
par_3error <- results_3error$param_est[1:sum(include_3error==1)]
tmp <- matrix(0, p, p)
L.zeta <- tmp
L.zeta[lower.tri(tmp, diag = T)] <- par_3error[1:(p*(p+1)/2)]
diag(L.zeta) <- exp(diag(L.zeta))
sigmazeta_3error <-  tcrossprod(L.zeta)

L.eta <- tmp
L.eta[lower.tri(tmp, diag = T)] <- par_3error[(p*(p+1)/2+1):(p*(p+1)/2*2)]
diag(L.eta) <- exp(diag(L.eta))
sigmaeta_3error <- tcrossprod(L.eta)

L.eps <- tmp
L.eps[lower.tri(tmp, diag = T)] <- par_3error[(p*(p+1)/2*2+1):(p*(p+1)/2*3)]
```

```
diag(L.eps) <- exp(diag(L.eps))
sigmaeps_3error <- tcrossprod(L.eps)
```

Table 13: Estimated Sigma zeta under the base model, two error components, trace=0.129

0.0203	0.0135	-0.0005
0.0135	0.0238	-0.0048
-0.0005	-0.0048	0.0849

Table 14: Estimated Sigma eps under the base model, two error components, trace=0.4075

0.1251	0.0030	0.0008
0.0030	0.1360	0.0007
0.0008	0.0007	0.1464

Table 15: Estimated Sigma zeta under the base model, three error components, trace=0.1287

0.0205	0.0136	-0.0006
0.0136	0.0238	-0.0047
-0.0006	-0.0047	0.0844

Table 16: Estimated Sigma eta under the base model, three error components, trace=0.0018

6e-04	6e-04	0e+00
6e-04	8e-04	-1e-04
0e+00	-1e-04	4e-04

Table 17: Estimated Sigma eps under the base model, three error components, trace=0.4058

0.1246	0.0024	0.0009
0.0024	0.1353	0.0008
0.0009	0.0008	0.1459

Importance ordering and decomposition of risk premia

The following code can be used to replicate Tables C6 of the supplementary material.

```
results_base <- read.csv(paste0(path, "/output/results_base_model_psi3.74.csv"),
                         stringsAsFactors = F)
psi_estimate <- 3.74

# get list of coefficients
coef <- as.data.frame(t(results_base$coef))
colnames(coef) <- results_base$var
# replace missing values with 0
```

```r
individual_data[is.na(individual_data)] <- 0
# prepare data
individual_data <- individual_data %>%
  mutate(
    Xpsi = mvecr::prelec(Xpsi_obj, psi = psi_estimate),
    log.price.real = log.price - log.CPI,
    City = factor(City,
                  levels = c(
                    'Tokyo', 'Osaka', 'Nagoya', 'Fukuoka', 'Sapporo'
                  )),
    Type = factor(
      Type,
      levels = c(
        'Residential Land(Land and Building)',
        'Residential Land(Land Only)',
        'Pre-owned Condominiums, etc.'
      )
    ),
    inf_constant = Type_LandBldg * coef$constant_LandBldg +
      Type_LandOnly * coef$constant_LandOnly +
      Type_Condo * coef$constant_Condo,
    inf_distance = distance.num * coef$distance.num,
    inf_area = area.m2.num * coef$area.m2.num,
    inf_floorarea = total.floor.area.m2.num * coef$total.floor.area.m2.num,
    inf_age = building.age * coef$building.age,
    inf_bldgyr = built.1981_2000 * coef$built.1981_2000 +
      built.after2000 * coef$built.after2000,
    inf_bldg_RC = LandBldg_RC * coef$LandBldg_RC,
    inf_bldg_S = LandBldg_S * coef$LandBldg_S,
    inf_bldg_W = LandBldg_W * coef$LandBldg_W,
    inf_bldgstructure = LandBldg_RC * coef$LandBldg_RC +
      LandBldg_S * coef$LandBldg_S +
      LandBldg_W * coef$LandBldg_W,
    inf_UC = Urban_Control * coef$Urban_Control,
    inf_bcr = max.building.coverage.ratio * coef$max.building.coverage.ratio,
    inf_far = max.floor.area.ratio * coef$max.floor.area.ratio,
    inf_city = City_Fukuoka * coef$City_Fukuoka +
      City_Nagoya * coef$City_Nagoya +
      City_Osaka * coef$City_Osaka +
      City_Sapporo * coef$City_Sapporo,
    inf_GDP = log.nGDP * coef$log.nGDP,
    inf_CPI_real = log.CPI * (coef$log.CPI - 1),
    inf_immi = PctImmi * coef$PctImmi,
    inf_crime = Ncrime * coef$Ncrime,
    inf_unemp = PctUnemploy * coef$PctUnemploy,
    inf_exec = PctExec * coef$PctExec,
    inf_lr4555 = JSHIS_I45_55 * coef$JSHIS_I45_55,
    inf_lr55 = JSHIS_I55 * coef$JSHIS_I55,
    inf_sr = Xpsi * coef$Xpsi,
    inf_total_real = abs(inf_constant) + abs(inf_city) +
      abs(inf_immi) + abs(inf_crime) + abs(inf_unemp) +
      abs(inf_exec) + abs(inf_GDP) + abs(inf_CPI_real) +
      abs(inf_area) + abs(inf_floorarea) + abs(inf_distance) + abs(inf_age) +
```

```
      abs(inf_bldgyr) + abs(inf_bldgstructure) + abs(inf_UC) +
      abs(inf_bcr) + abs(inf_far) + abs(inf_lr4555) + abs(inf_lr55) +
      abs(inf_sr) ,
    pct_intercept_real = abs(inf_constant) / inf_total_real ,
    pct_city_real = abs(inf_city) / inf_total_real ,
    pct_int_city_real = pct_intercept_real + pct_city_real,
    pct_ward_real = (abs(inf_immi) + abs(inf_crime) + abs(inf_unemp) +
                       abs(inf_exec)) / inf_total_real ,
    pct_macro_real = (abs(inf_GDP) + abs(inf_CPI_real)) / inf_total_real ,
    pct_prop_basic_real = (
      abs(inf_area) + abs(inf_floorarea) +
        abs(inf_distance) + abs(inf_age)
    ) / inf_total_real ,
    pct_prop_ext_real = (
      abs(inf_bldgyr) + abs(inf_bldgstructure) + abs(inf_UC) +
        abs(inf_bcr) + abs(inf_far)
    ) / inf_total_real ,
    pct_prop_all_real = pct_prop_basic_real + pct_prop_ext_real,
    pct_lr_real = (abs(inf_lr4555) + abs(inf_lr55)) / inf_total_real ,
    pct_sr_real = abs(inf_sr) / inf_total_real

  )

individual_data <- individual_data %>%
  mutate(prediction_real = inf_constant + inf_city +
           inf_immi + inf_crime + inf_unemp + inf_exec +
           inf_GDP + inf_CPI_real +
          inf_area + inf_floorarea +
           inf_bldgyr + inf_bldg_RC + inf_bldg_S + inf_far +
           inf_distance + inf_age +
           inf_bldg_W + inf_UC + inf_bcr+
           inf_lr4555 + inf_lr55 + inf_sr) %>%
  mutate(
    pct_type_real = inf_constant / prediction_real,
    pct_city_real = inf_city / prediction_real,
    pct_intercept_real = pct_type_real + pct_city_real,
    pct_ward_positive_real = (inf_immi + inf_exec) / prediction_real,
    pct_ward_negative_real = (inf_crime + inf_unemp) / prediction_real,
    pct_macro_real = (inf_GDP + inf_CPI_real) / prediction_real,
    pct_prop_positive_real =
      (inf_area + inf_floorarea +
         inf_bldgyr + inf_bldg_RC + inf_bldg_S + inf_far) / prediction_real,
    pct_prop_negative_real = (inf_distance + inf_age + inf_bldg_W + inf_UC +
                            inf_bcr ) / prediction_real,
    pct_lr_real = (inf_lr4555 + inf_lr55) / prediction_real,
    pct_sr_real = inf_sr / prediction_real
  )
```

```
sumstat.inf1 <- individual_data %>%
  group_by(Type) %>%
  summarise(
    intercept = median(pct_intercept_real),
    `W_+` = median(pct_ward_positive_real),
    `W_-` = median(pct_ward_negative_real),
```

```
    M = median(pct_macro_real),
    `P_+` = median(pct_prop_positive_real),
    `P_-` = median(pct_prop_negative_real),
    longrun = median(pct_lr_real),
    shortrun = median(pct_sr_real),
    total1 = median(pct_intercept_real+pct_ward_positive_real+pct_ward_negative_real+
                      pct_macro_real+pct_prop_positive_real+pct_prop_negative_real+
                      pct_lr_real+pct_sr_real)
  ) %>%
  mutate(total = intercept +
           `W_+` + `W_-` + M +
           `P_+` + `P_-` +
           longrun + shortrun) %>%
  mutate_at(
    vars(-Type),
    ~sprintf("%1.2f%%", 100*.)
  ) %>%
  mutate(Type = c('land bldg', 'land only', 'condo'))
sumstat.inf2 <- individual_data %>%
  group_by(City) %>%
  summarise(
    intercept = median(pct_intercept_real),
    `W_+` = median(pct_ward_positive_real),
    `W_-` = median(pct_ward_negative_real),
    M = median(pct_macro_real),
    `P_+` = median(pct_prop_positive_real),
    `P_-` = median(pct_prop_negative_real),
    longrun = median(pct_lr_real),
    shortrun = median(pct_sr_real)
  ) %>%
  mutate(total = intercept +
           `W_+` + `W_-` + M +
           `P_+` + `P_-` +
           longrun + shortrun) %>%
  mutate_at(
    vars(-City),
    ~sprintf("%1.2f%%", 100*.)
  )

sumstat.inf3 <- individual_data %>%
  group_by(Type) %>%
  summarise(
    intercept = quantile(pct_intercept_real, 0.75) - quantile(pct_intercept_real, 0.25),
    `W_+` = quantile(pct_ward_positive_real, 0.75) - quantile(pct_ward_positive_real, 0.25),
    `W_-` = quantile(pct_ward_negative_real, 0.75) - quantile(pct_ward_negative_real, 0.25),
    M = quantile(pct_macro_real, 0.75) - quantile(pct_macro_real, 0.25),
    `P_+` = quantile(pct_prop_positive_real, 0.75) - quantile(pct_prop_positive_real, 0.25),
    `P_-` = quantile(pct_prop_negative_real, 0.75) - quantile(pct_prop_negative_real, 0.25),
    longrun = quantile(pct_lr_real, 0.75) - quantile(pct_lr_real, 0.25),
    shortrun = quantile(pct_sr_real, 0.75) - quantile(pct_sr_real, 0.25)
  ) %>%
   mutate_at(
    vars(-Type),
```

```
    ~sprintf("%1.2f%%", 100*.)
  )%>%
  mutate(Type = c('land bldg', 'land only', 'condo'))

sumstat.inf4 <- individual_data %>%
  group_by(City) %>%
  summarise(
    intercept = quantile(pct_intercept_real, 0.75) -
      quantile(pct_intercept_real, 0.25),
    `W_+` = quantile(pct_ward_positive_real, 0.75) -
      quantile(pct_ward_positive_real, 0.25),
    `W_-` = quantile(pct_ward_negative_real, 0.75) -
      quantile(pct_ward_negative_real, 0.25),
    M = quantile(pct_macro_real, 0.75) - quantile(pct_macro_real, 0.25),
    `P_+` = quantile(pct_prop_positive_real, 0.75) -
      quantile(pct_prop_positive_real, 0.25),
    `P_-` = quantile(pct_prop_negative_real, 0.75) -
      quantile(pct_prop_negative_real, 0.25),
    longrun = quantile(pct_lr_real, 0.75) -
      quantile(pct_lr_real, 0.25),
    shortrun = quantile(pct_sr_real, 0.75) -
      quantile(pct_sr_real, 0.25)
  ) %>%
   mutate_at(
    vars(-City),
    ~sprintf("%1.2f%%", 100*.)
  )
```

Table 18: Median of influences of each component, by type

Type	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun	total1	total
land bldg	32.54%	4.97%	-2.85%	65.33%	6.00%	-3.54%	-1.87%	-0.14%	100.00%	100.43%
land only	31.52%	5.14%	-2.84%	66.37%	3.60%	-1.91%	-1.80%	-0.00%	100.00%	100.08%
condo	29.36%	6.13%	-3.02%	69.11%	4.03%	-3.34%	-1.96%	-0.18%	100.00%	100.13%

Table 19: Median of influences of each component, by city

City	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun	total
Tokyo	31.19%	5.82%	-2.53%	65.98%	4.52%	-2.85%	-1.86%	-0.26%	100.02%
Osaka	30.95%	4.57%	-4.64%	68.99%	4.91%	-3.23%	-2.28%	-0.00%	99.27%
Nagoya	30.35%	4.98%	-2.69%	68.08%	5.18%	-3.13%	-2.80%	-0.00%	99.97%
Fukuoka	25.19%	5.35%	-3.24%	70.44%	4.87%	-3.63%	-0.61%	-0.00%	98.37%
Sapporo	24.42%	5.34%	-3.18%	71.45%	5.32%	-3.62%	-0.33%	-0.00%	99.40%

Table 20: IQR of influences of each component, by type

Type	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun
land bldg	3.20%	1.62%	1.04%	4.36%	2.73%	2.24%	0.85%	0.25%
land only	3.94%	1.59%	0.95%	4.04%	2.96%	0.87%	0.85%	0.24%
condo	2.17%	2.26%	1.08%	4.90%	2.16%	2.19%	0.76%	0.31%

Table 21: IQR of influences of each component, by city

City	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun
Tokyo	3.16%	1.77%	0.79%	3.82%	2.70%	1.94%	0.76%	0.15%
Osaka	3.54%	3.06%	1.02%	4.79%	3.02%	2.47%	0.50%	0.00%
Nagoya	2.66%	1.52%	1.03%	4.37%	3.32%	2.08%	0.87%	0.00%
Fukuoka	2.92%	2.43%	0.74%	5.90%	3.52%	2.40%	0.21%	0.00%
Sapporo	3.08%	1.16%	0.40%	5.56%	3.75%	2.55%	0.32%	0.00%

We can also compute these influences per quarter, in particular the quarter after the Tohoku earthquake (2011/Q2).

```
sumstat.inf1.tohoku <- individual_data %>%
  filter(Q_after_Fukushima==1) %>%
  group_by(Type) %>%
  summarise(
    intercept = median(pct_intercept_real),
    `W_+` = median(pct_ward_positive_real),
    `W_-` = median(pct_ward_negative_real),
    M = median(pct_macro_real),
    `P_+` = median(pct_prop_positive_real),
    `P_-` = median(pct_prop_negative_real),
    longrun = median(pct_lr_real),
    shortrun = median(pct_sr_real),
    total1 = median(pct_intercept_real+pct_ward_positive_real+pct_ward_negative_real+
                      pct_macro_real+pct_prop_positive_real+pct_prop_negative_real+
                      pct_lr_real+pct_sr_real)
  ) %>%
  mutate(total = intercept +
           `W_+` + `W_-` + M +
           `P_+` + `P_-` +
           longrun + shortrun) %>%
  mutate_at(
    vars(-Type),
    ~sprintf("%1.2f%%", 100*.)
  ) %>%
  mutate(Type = c('land bldg', 'land only', 'condo'))

sumstat.inf2.tohoku <- individual_data %>%
  filter(Q_after_Fukushima==1) %>%
  group_by(City) %>%
```

```
  summarise(
    intercept = median(pct_intercept_real),
    `W_+` = median(pct_ward_positive_real),
    `W_-` = median(pct_ward_negative_real),
    M = median(pct_macro_real),
    `P_+` = median(pct_prop_positive_real),
    `P_-` = median(pct_prop_negative_real),
    longrun = median(pct_lr_real),
    shortrun = median(pct_sr_real)
  ) %>%
  mutate(total = intercept +
           `W_+` + `W_-` + M +
           `P_+` + `P_-` +
           longrun + shortrun) %>%
  mutate_at(
    vars(-City),
    ~sprintf("%1.2f%%", 100*.)
  )
```

Table 22: Median of influences of each component, by type, 2011Q2

Type	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun	total1	total
land bldg	32.70%	4.96%	-2.90%	65.58%	5.95%	-3.55%	-1.90%	-0.00%	100.00%	100.84%
land only	31.60%	5.14%	-2.85%	66.50%	3.47%	-1.94%	-1.79%	-0.00%	100.00%	100.13%
condo	29.33%	6.19%	-3.08%	69.33%	3.96%	-3.32%	-1.90%	-0.40%	100.00%	100.10%

Table 23: Median of influences of each component, by city, 2011Q2

City	intercept	W_+	W_-	M	P_+	P_-	longrun	shortrun	total
Tokyo	31.31%	5.88%	-2.55%	66.01%	4.30%	-2.80%	-1.86%	-0.40%	99.89%
Osaka	31.14%	4.60%	-4.69%	68.94%	4.77%	-3.18%	-2.30%	-0.00%	99.28%
Nagoya	30.48%	4.93%	-2.68%	67.99%	4.91%	-3.12%	-2.79%	-0.00%	99.71%
Fukuoka	25.24%	5.43%	-3.31%	70.73%	4.51%	-3.61%	-0.61%	-0.00%	98.38%
Sapporo	23.92%	5.40%	-3.21%	71.94%	4.81%	-3.64%	-0.33%	-0.00%	98.88%

The following code can be used to replicate Table C7 of the supplementary material.

```
# risk premia
individual_data <- individual_data %>%
  mutate(prediction =
           Type_LandBldg * coef$constant_LandBldg +
           Type_LandOnly * coef$constant_LandOnly +
           Type_Condo * coef$constant_Condo +
           City_Fukuoka * coef$City_Fukuoka +
           City_Nagoya * coef$City_Nagoya +
           City_Osaka * coef$City_Osaka +
           City_Sapporo * coef$City_Sapporo +
           PctImmi * coef$PctImmi +
           Ncrime * coef$Ncrime +
           PctUnemploy * coef$PctUnemploy +
```

```
    PctExec * coef$PctExec +
    log.CPI * coef$log.CPI +
    log.nGDP * coef$log.nGDP +
    distance.num * coef$distance.num +
    area.m2.num * coef$area.m2.num +
    total.floor.area.m2.num * coef$total.floor.area.m2.num +
    building.age * coef$building.age +
    built.1981_2000 * coef$built.1981_2000 +
    built.after2000 * coef$built.after2000 +
    LandBldg_RC * coef$LandBldg_RC +
    LandBldg_S * coef$LandBldg_S +
    LandBldg_W * coef$LandBldg_W +
    Urban_Control * coef$Urban_Control +
    max.building.coverage.ratio * coef$max.building.coverage.ratio +
    max.floor.area.ratio * coef$max.floor.area.ratio +
    JSHIS_I45_55 * coef$JSHIS_I45_55 +
    JSHIS_I55 * coef$JSHIS_I55 +
    Xpsi * coef$Xpsi,
  prediction_real = prediction - log.CPI * 1,
  m0 = Type_LandBldg * coef$constant_LandBldg +
    Type_LandOnly * coef$constant_LandOnly +
    Type_Condo * coef$constant_Condo +
    City_Fukuoka * coef$City_Fukuoka +
    City_Nagoya * coef$City_Nagoya +
    City_Osaka * coef$City_Osaka +
    City_Sapporo * coef$City_Sapporo +
    PctImmi * coef$PctImmi +
    Ncrime * coef$Ncrime +
    PctUnemploy * coef$PctUnemploy +
    PctExec * coef$PctExec +
    log.CPI * coef$log.CPI +
    log.nGDP * coef$log.nGDP +
    distance.num * coef$distance.num +
    area.m2.num * coef$area.m2.num +
    total.floor.area.m2.num * coef$total.floor.area.m2.num +
    building.age * coef$building.age +
    built.1981_2000 * coef$built.1981_2000 +
    built.after2000 * coef$built.after2000 +
    LandBldg_RC * coef$LandBldg_RC +
    LandBldg_S * coef$LandBldg_S +
    LandBldg_W * coef$LandBldg_W +
    Urban_Control * coef$Urban_Control +
    max.building.coverage.ratio * coef$max.building.coverage.ratio +
    max.floor.area.ratio * coef$max.floor.area.ratio,
  m1 = Type_LandBldg * coef$constant_LandBldg +
    Type_LandOnly * coef$constant_LandOnly +
    Type_Condo * coef$constant_Condo +
    City_Fukuoka * coef$City_Fukuoka +
    City_Nagoya * coef$City_Nagoya +
    City_Osaka * coef$City_Osaka +
    City_Sapporo * coef$City_Sapporo +
    PctImmi * coef$PctImmi +
    Ncrime * coef$Ncrime +
```

```
          PctUnemploy * coef$PctUnemploy +
          PctExec * coef$PctExec +
          log.CPI * coef$log.CPI +
          log.nGDP * coef$log.nGDP +
          distance.num * coef$distance.num +
          area.m2.num * coef$area.m2.num +
          total.floor.area.m2.num * coef$total.floor.area.m2.num +
          building.age * coef$building.age +
          built.1981_2000 * coef$built.1981_2000 +
          built.after2000 * coef$built.after2000 +
          LandBldg_RC * coef$LandBldg_RC +
          LandBldg_S * coef$LandBldg_S +
          LandBldg_W * coef$LandBldg_W +
          Urban_Control * coef$Urban_Control +
          max.building.coverage.ratio * coef$max.building.coverage.ratio +
          max.floor.area.ratio * coef$max.floor.area.ratio +
          JSHIS_I45_55 * coef$JSHIS_I45_55 +
          JSHIS_I55 * coef$JSHIS_I55,
        m2 = Type_LandBldg * coef$constant_LandBldg +
          Type_LandOnly * coef$constant_LandOnly +
          Type_Condo * coef$constant_Condo +
          City_Fukuoka * coef$City_Fukuoka +
          City_Nagoya * coef$City_Nagoya +
          City_Osaka * coef$City_Osaka +
          City_Sapporo * coef$City_Sapporo +
          PctImmi * coef$PctImmi +
          Ncrime * coef$Ncrime +
          PctUnemploy * coef$PctUnemploy +
          PctExec * coef$PctExec +
          log.CPI * coef$log.CPI +
          log.nGDP * coef$log.nGDP +
          distance.num * coef$distance.num +
          area.m2.num * coef$area.m2.num +
          total.floor.area.m2.num * coef$total.floor.area.m2.num +
          building.age * coef$building.age +
          built.1981_2000 * coef$built.1981_2000 +
          built.after2000 * coef$built.after2000 +
          LandBldg_RC * coef$LandBldg_RC +
          LandBldg_S * coef$LandBldg_S +
          LandBldg_W * coef$LandBldg_W +
          Urban_Control * coef$Urban_Control +
          max.building.coverage.ratio * coef$max.building.coverage.ratio +
          max.floor.area.ratio * coef$max.floor.area.ratio +
          JSHIS_I45_55 * coef$JSHIS_I45_55 +
          JSHIS_I55 * coef$JSHIS_I55 +
          Xpsi_obj * coef$Xpsi,
        m3 = prediction
        )

sumstat.rp <- individual_data %>%
  group_by(Type, City) %>%
  summarise(median_log_price = median(log.price),
                    pred.m0 = median(m0),
```

```
                  pred.m1 = median(m1),
                  pred.m2 = median(m2),
                  pred.m3 = median(m3)) %>%
mutate(premium_lr = pred.m1 - pred.m0,
       premium_sr_obj = pred.m2 - pred.m1,
       premium_sr_sub = pred.m3 - pred.m2) %>%
mutate_at( c(
    'median_log_price',
    'premium_lr',
    'premium_sr_obj',
    'premium_sr_sub'
  ),
  formatC,
  format = "f",
  digits = 4) %>%
select(Type, City, median_log_price, premium_lr, premium_sr_obj, premium_sr_sub)
```

Table 24: Decomposition of the premia for earthquake risk per type and city

Type	City	median_log_price	premium_lr	premium_sr_obj	premium_sr_sub
Residential Land(Land and Building)	Tokyo	17.7275	-0.2620	-0.0246	-0.0092
Residential Land(Land and Building)	Osaka	17.2495	-0.2783	-0.0049	0.0049
Residential Land(Land and Building)	Nagoya	17.4264	-0.3393	-0.0076	0.0076
Residential Land(Land and Building)	Fukuoka	17.2812	-0.0630	-0.0043	0.0043
Residential Land(Land and Building)	Sapporo	17.0736	-0.0558	-0.0016	0.0016
Residential Land(Land Only)	Tokyo	17.7073	-0.2409	-0.0241	-0.0087
Residential Land(Land Only)	Osaka	17.2167	-0.2691	-0.0048	0.0048
Residential Land(Land Only)	Nagoya	17.1113	-0.3293	-0.0077	0.0077
Residential Land(Land Only)	Fukuoka	16.9066	-0.0658	-0.0046	0.0046
Residential Land(Land Only)	Sapporo	16.3805	-0.0517	-0.0016	0.0016
Pre-owned Condominiums, etc.	Tokyo	17.0344	-0.2621	-0.0246	-0.0093
Pre-owned Condominiums, etc.	Osaka	16.5881	-0.2677	-0.0051	0.0051
Pre-owned Condominiums, etc.	Nagoya	16.5236	-0.3175	-0.0079	0.0079
Pre-owned Condominiums, etc.	Fukuoka	16.2134	-0.0740	-0.0042	0.0042
Pre-owned Condominiums, etc.	Sapporo	16.2134	-0.0457	-0.0015	0.0015