A Spatial-Temporal Analysis of Corporate Bankruptcies

[redacted]

1 Introductory Comments

Typically, corporations and individuals that lend money to a firm receive close to nothing back if that firm goes bankrupt. Those who had lent out money to the investment bank Lehman Brothers, for example, received only 17 cents for every dollar they gave to Lehman[1] - Lehman's bankruptcy was one of the largest in US history. Notice, consequently, that if a corporation lends money to another one, and the recipient of those funds never pays back the funds it owes, that corporation (i.e., the lender/creditor) may have difficulties in meeting its financial obligations itself, and thus may also go bankrupt - especially if there are limited sources of funds available to such firm. Thus, one bankruptcy can trigger another one, and so on, much like earthquakes can. And also, much like earthquakes, bankruptcies are somewhat "rare" and unexpected. When taken together, these facts suggest that point process models, which are commonly used as frameworks for analyzing events such as earthquakes and wildfires, may also shed light on the occurrence of bankruptcies - in particular, they can help us in measuring how these events can trigger others through time.

Asset managers, furthermore, frequently think about an abstract asset allocation space where they, on a daily basis, invest the money/funds entrusted to them by investors. Instead of thinking about our usual spatial dimensions, latitude and longitude, however, they often instead think about two other dimensions: (i) market value and (ii) liquidation value. Market value measures how much something costs. In the context of corporations, it measures how much one would have to pay to buy off all of the shares of a firm (this is called market capitalization). The other dimension, liquidation value, measures what would be left off to the owner of a firm if he or she sold off all of its assets and paid off all of its liabilities (i.e. if the firm is liquidated) - this can be proxied through the accounting measure book-value-of-equity that can be found in the balance sheet statement of any firm.

[1] See http://nypost.com/2014/08/15/lehman-bondholders-to-receive-17-cents-on-dollar/.

The conditional intensity associated with bankruptcies, which hereafter we refer to as λ , may be high in some regions of this space, and low in others. So much like an individual might seek to avoid going to a crime-prone neighborhood and be robbed, an asset manager may want to avoid investing in, or "going to", a region of this "asset allocation space" where λ is high. Figure 1 presents a diagram that depicts this space. The y axis in this graph shows the variable market value, or market capitalization; the x axis represents liquidation value, or book-value-of-equity. Asset managers often think that the bankruptcy intensity is low for valuable firms (area in yellow), and high for firms with low market value (area in blue). Also, they have reasons to believe that λ is high below the 45 degree line shown in this figure, and low above it[2]. One can reasonably conjecture, therefore, that there is inhomogeneity in this space - which point process models can help us analyze.

In this study, we make use of the aforementioned two-dimensional abstract asset allocation space - our coordinates are, as we alluded to above, market value (market capitalization[3]) and liquidation value (proxied via book-value-of-equity[4]); and proceed to model bankruptcies of publicly-traded firms as purely spatial and as spatial-temporal point processes. By modeling the data thorough an epidemic-type aftershock sequence model, we hope to better understand the temporal triggering functions associated with these rare events, and gain further insights into their productivity. We argue that such knowledge can have economic/policy implications. A purely spatial analysis, moreover, helps to further elucidate how the conditional intensity may vary across our spatial dimensions.

2 Brief Description of Data

The data we analyze is composed of 456 points, each of which represents the bankruptcy of a publicly-traded firm[5]. Our observation time-window goes from January 1971 to December 2007[6]. Each point τ_i ($i \in \{1,...,456\}$) is associated with two coordinates in space, x_i and y_i, which represent, respectively, the book-value-of-equity and market capitalization of a firm filing for bankruptcy. Each point also has a

[2]Below the 45 degree line we have firms whose liquidation values are higher than their market values. This indicates that such firms possibly face greater liquidation risk than firms that are in the region situated above the 45 degree line - which have market value above liquidation value and are thus seen as ongoing concerns.

[3]We'll refer to the terms market capitalization and market value interchangeably.

[4]We'll also refer to book-value-of-equity and liquidation value interchangeably going forward.

[5]Traded on the New York Stock Exchange, American Stock Exchange or NASDAQ; and recorded in the CRSP US Stock database under delisting code 574.

[6]Note that we have, purposely, left out of our analysis the period corresponding to the financial crisis. During such period, the US Government engaged in an unprecedented effort to provide funds to firms that were close to experiencing financial distress. If we were to include this period, our parameter estimates might underestimate the impact of one bankruptcy on future ones, in normal times - i.e., without government interventions.

time coordinate, t_i, which denotes the time/day when the bankruptcy occurred; as well as a mark, m_i, that contains the amount of debt each bankrupt firm had when it became insolvent. The more debt a firm has when it goes bankrupt, the more bankruptcies it will likely trigger - debt-levels, therefore, are analogous to earthquake magnitudes.

The market capitalization of a firm filing for bankruptcy was computed as the product of the share-price of such firm on the day prior to the filing[7], and the total number of shares outstanding for such firm on that same day. Prices and numbers of shares, used in that computation, come from the CRSP (Center for Research in Security Prices) US Stock Database. Debt-levels[8] and data on book-values-of-equity come from the Compustat Annual database. Compustat compiles accounting data that is presented in annual financial reports filed by corporations.

A time covariate was also used in our study - we refer to it as the default spread, or simply as spread. It measures the difference between the return (or yield) promised by high risk corporate bonds (rated BAA by the credit rating agency Moody's), and the return promised by safe corporate bonds (rated AAA by the same agency). This covariate proxies for the state of the economy. When the economy is booming this variable tends to be low, and in periods of economic contraction it tends to be high. Data for this covariate was obtained from the website of the Board of Governors of the Federal Reserve System.

3 Analysis

3.1 Overview of Point Pattern

Figure 2 presents the point pattern we analyze in this study. Each circle in this diagram represents one of the points in our sample - these, as we argued, denote in turn bankruptcies of publicly-traded firms. Circle sizes are proportional to the outstanding debt-levels of bankrupt firms, i.e., they are proportional to the values of marks associated with each point (debt-levels are in billions of US dollars). Lighter colors represent events/points occurring in more recent times; darker colors, conversely,

[7] If share-price on the day prior to the filing was unavailable, we used instead share-price on the most recent date prior to the filing, provided that date was at most 7 working days apart from the bankruptcy filing date.

[8] The debt-level of a firm filing for bankruptcy was defined as the total debt of such firm, in billions of dollars, as shown in Compustat. If this information was not available in that database, we used instead long-term debt plus current liabilities to represent the debt-level of an insolvent firm. If current liabilities were also unavailable, we then used instead long-term debt. In the rare occurrences were neither total debt nor long-term debt were available, current liabilities were used to proxy for debt-levels.

denote bankruptcy filings occurring in older periods. This figure illustrates that some firms filing for bankruptcy had market values in excess of US $500 million when they became insolvent[9], indicating that some of these events may have come as a shocking surprise to market participants. Figure 2 also suggests the potential occurrence of "temporal clustering" in our data - the colors shown in this figure imply that a good portion of bankruptcy events occurred in the early 2000's.

In Figure 3, we show the spatial region with greatest activity. We can see here that most bankruptcies occurred in the region denoted by market capitalizations of 0 to 250 million dollars, and book values of -2 to 6 billion dollars.

Figure 4 shows a histogram containing bankruptcy events per year of occurrence. We have periods with few bankruptcies followed by periods with many, again suggesting some type of temporal triggering in our data.

In the analyses that follow, our x, y and t coordinates were standardized so as to go from 0 to 1. Similarly, all marks (debt-levels) were standardized so as to be within that same interval.

3.2 Purely Spatial Analysis

We start our analysis by conducting a purely spatial assessment of our data. Figure 5 shows the F, G and J functions, all of which indicate potential clustering or inhomogeneity in the data. The F function shows, for example, that there is roughly a 22% probability that the distance of a randomly chosen location to its nearest point is less than or equal to 0.075 - a value much smaller than we would expect if the underlying process generating these points in space were a stationary Poisson process. This, in turn, indicates the prevalence of "a good deal of empty space" in our $[0, 1] \times [0, 1]$ spatial window, suggesting thus clustering or simply inhomogeneity. The G function shows that the probability that a randomly chosen point is less than or equal to, say, 0.025, is almost 1 - this suggests that points are much closer than what we would expect if the underlying process was a stationary Poisson process - which again indicates possible clustering, or inhomogeneity. Lastly, the J function lies below 1 for various distances, also confirming the conclusions we drew from the F and G functions about the process generating our points.

Figure 6 depicts the K function, and Figure 7 the L function. Both of these also indicate the potential occurrence of clustering or inhomogeneity. Figure 8 shows the values of our marked G function. This

[9] These corresponded to the bankruptcies of Mirant Corporation (an energy producer), Delphi (a producer of auto parts), Worldcom (a telecommunications company) and Enron (an energy, commodities and services firm).

diagram shows the probability that a point τ_i with mark $m_i \leq 0.5$ is within distance r of another point τ_j with mark $m_j \geq 0.8$, $i \neq j$. This graph indicates that there is a probability of almost 0 that a bankruptcy with low magnitude ($m \leq 0.5$) will be very close to one of large magnitude ($m \geq 0.8$) in our asset allocation space.

Figure 9 contains the results of our kernel estimation of the Papangelou intensity, when a bandwidth of 0.8 is used (other bandwidth levels were used as well)[10].The estimates we obtained with this bandwidth (and others) seem to be arguably too low for the region where most points are: a quick visual inspection of this graph shows that if we were integrate the estimated rate over, say, the $[0, 0.2] \times [0, 0.2]$ square, we would get an estimated expected number of points in this region much lower than the actual number of points observed there.

Next, we fit an inhomogeneous Poisson model to our data using the pseudo-loglikelihood method. The model we use is of the form $\lambda(x, y) = \mu + ax + by + cxy$. The results of our estimation are shown in Figure 10. We see here that, for example, a one unit increase in y (which corresponds to an increase of approximately US$ 700 million in market value) promotes an increase in the Papangelou intensity of $\frac{\partial\lambda(x,y)}{\partial y} = b + cx = 50.27 - 341.42x$. Hence, the impact of an increase in y on the Papangelou intensity depends on the value of the x coordinate in our diagram. A similar analysis can be applied to interpret how changes in x affect $\lambda(x, y)$. A spatial-temporal approach is in order.

3.3 ETAS Model with Covariate

The ETAS model we fit is presented below. We have:

$$\lambda(x, y, t) = \mu(x, y, t) + \sum_{\{t'|'t'<t\}} g^*(x, y) g(t - t^{'}) h(m')$$

Where:

$$\mu(t, x, y) = \bar{\mu} + ax + by + ct + dspread_t$$

[10] We choose to use a "large" bandwidth level so as to "squash" more our points and thus get estimates for the Papangelou intensity that would be non-zero across a large range of the $[0, 1] \times [0, 1]$ spatial window.

And:

$$g(t) = \alpha e^{-\alpha(t)},$$
$$h(m') = Ke^{\beta m'}$$
$$g^*(x,y) = \frac{1_{\{x \leq 0.2, y \leq 0.2)}}{0.2 \times 0.2}$$

Note that the background rate, here denoted as $\mu(x,y,t)$, is somewhat different from the one initially proposed by Ogata. We allow for a time trend to influence μ; moreover, the background rate depends here on our spread covariate, which proxies for the state of the economy; and we specify μ as a linear function of x, y, t and our covariate - this facilitates interpreting the parameters that affect μ.

The temporal triggering function is exponential. The impact function, h, is as usual, with m_0 set to 0 - so we consider all shocks/bankruptcies in our study.

Notice that the spatial triggering function that we use, $g^*(x,y)$, is a density: it integrates to one over all space. Differently from the spatial triggering functions we studied, however, $g^*(x,y)$ takes on a constant value over a small 0.2 by 0.2 square, which in our model represents the most "active spatial region". The idea here is that bankruptcies, wherever they happen, will lead to more bankruptcies in exactly this active region, but not in other places. Firms in this region are more fragile. They will usually have accumulated many losses, and have limited access to capital markets. Firms in other regions will likely be less affected by previous bankruptcies given they may either have enough cash available or have access to outside funding sources. Thus, this active region can be thought of as "a death zone".

Our parameter estimates are shown in Table 1 - they were obtained via maximum likelihood[11]. The background rate falls as x or y increases. An increase of one unit in y (or roughly US$ 700 million) is associated, for example, with a decrease of 19.78 in the background rate. The a and c coefficients can be interpreted in a similar fashion[12].

Moreover, a one unit increase in the spread covariate is associated with a 16.67 increase in the

[11] We fit our model using optim in R - to arrive at our results, we ran optim twice. This procedure took 5 minutes and 45 seconds in total. We also tried using simulated annealing to maximize our log-likelihood function (using the function GenSA, from the GenSA package). We allowed GenSA to also run for 5 minutes and 45 seconds. Despite having performed well in a number of global optimization tests performed by Prof. Kathreen Mullen (see "Continuous Global Optimization in R", 2014, Journal of Statistical Software), GenSA yielded a lower log-likelihood than optim.

[12] Notice that even though our standard errors are high for many parameters, these are only relevant to the extent that we have found an optimal solution to the global maximization problem at hand, which may not be the case. We should thus take these with a grain of salt.

background rate (this covariate oscillated between 0.55 and 2.69 within our sample)[13].

The expected number of first generation aftershocks is given by $E[h(m)]$, which we estimate to be equal to 0.9753, indicating that bankruptcies are fairly persistent events.

The temporal triggering function, $g(t)$, is shown in Figure 11. Put simply, this function suggests that bankruptcies can help trigger bankruptcies up to approximately 20 months into the future.

Lastly, an analysis of the performance of our model is shown in Figures 12 and 13. Superthinning (Figure 12) shows that our ETAS model overpredicts (λ is too high) in the active 0.2 by 0.2 region. One could potentially deal with this by simply reducing the size of the "death zone" region in the model, where triggering takes place. The G function of the superthinned points (shown in Figure 13) shows, nevertheless, that the calibrated model does a reasonably good job in explaining our data.

Our model has investment implications, as well as implications for regulatory practices/policies. We discuss these below in our concluding remarks.

4 Final Analysis and Concluding Remarks

Firstly, note that our model has implications for the regulatory environment in which firms operate. What is too big to fail? If the expected value of the impact function, $h(m)$, is greater than 1, the process we model becomes explosive. Thus, government could potentially consider passing legislation stating for example to no firm could have debt levels that made $h(m)$ be greater than 1. This would preclude the expectation of $h(m)$ from ever being above 1. We can solve for the value of m tied to this rule. By doing so, we get that the maximum debt that should be allowed, according to such policy, should be about 26 billion dollars. But some firms have more than 600 billion dollars in debt today! This suggest that some firms in operation may, indeed, be too big to fail.

Our model also has investment implications. As we move farther to the right in our spatial diagram (i.e., as x increases), the background rate decreases. The region were x is "high" is also unaffected by triggering in our model. The conditional intensity, λ, is thus relatively low for high values of x, indicating that the right-most region of our asset allocation space is safer, from a bankruptcy perspective. Suppose one invests in stocks in this region, which also have liquidation value (x coordinate) above their market value (y coordinate). Assume that these stocks, moreover, also have "good fundamentals",

[13] The spread covariate was not standardized in our analysis.

in the sense that their earnings are non-decreasing[14]. These companies are somewhat unlikely to go bankrupt, as they are situated in a region where x is relatively high. They are also unlikely to move to the left-most region of the spatial diagram, considering they have positive earnings - earnings tend to be the main driver of changes in book-values-of-equity. But notice that if they do not go bankrupt, then in the long run we would expect their market values to rise and become at least as large as their liquidation values. This could potentially thus be a "winning" strategy. Interestingly, one could argue that such strategy is very much in line with the one Warren Buffet seems to have implemented in the later stages of his life: buying stocks of firms that look "cheap" (i.e. which have liquidation value above market value); that are large (have high liquidation/book-values); and which have "good fundamentals" (and thus potentially have positive earnings); and holding on to them for long periods of time[15].

Note, however, that our study is not devoid of limitations. We have not considered the impact of bankruptcies of private firms on publicly traded ones - having more time, we would like to find databases which contain information on bankruptcy events associated with such firms so as to incorporate these into our analysis. Other covariates, particularly those associated with the availability of credit in the economy, could also impact our estimates.

Lastly, we could also have defined the asset allocation space differently, using "network theories" to create measures of proximity between firms. The temporal and spatial triggering functions, moreover, could also be estimated non-parametrically. We intend to implement these extensions in future studies.

[14] Decreases in book-values-of-equity (x coordinate) tend to occur mostly due to negative earnings.

[15] In a recent documentary conducted by HBO, "Becoming Warren Buffet", Buffet states that early on in his life he implemented a strategy similar to this one - however, instead of investing in the region were x is high, he instead invested in small firms (which would tend to have low liquidation/book values). He apparently later on realized that this strategy was "too risky", and decided to move his investments towards larger firms, which would tend to be farther to the right in our diagram.

5 APPENDIX

5.1 Figures and Tables

Figure 1: Abstract Asset Allocation Space: Yellow - large-cap stocks; green - mid-cap stocks; blue - small-cap stocks. Conditional intensity is expected to be high in blue region, and in region below the 45 degree line.

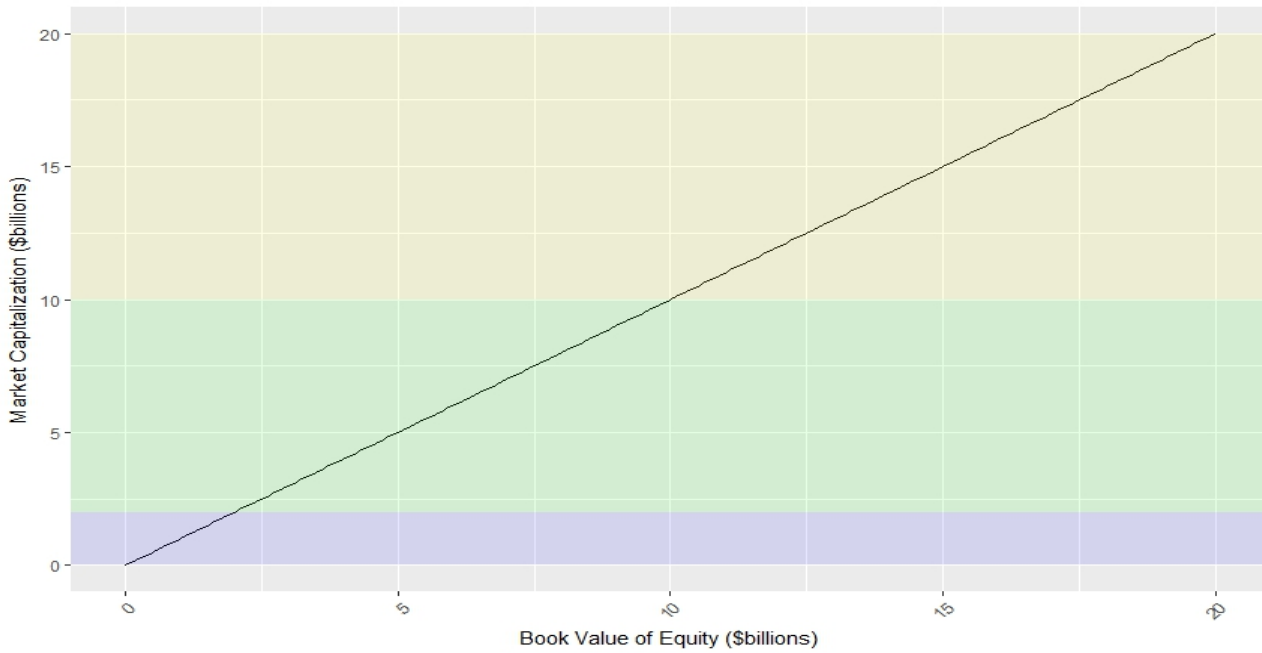


Figure 2: Bankruptcies in Space-Time: most companies going bankrupt have market values below $ 200 million dollars; and book values below $5 billion.

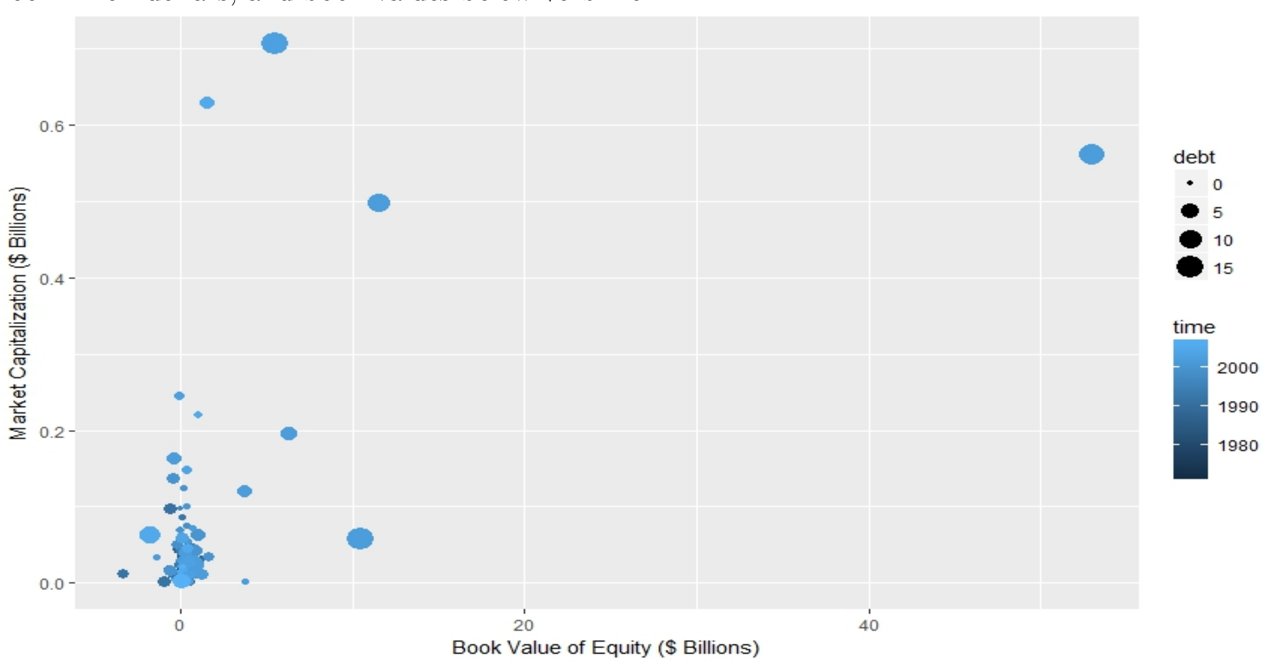


Figure 3: Bankruptcies in Region with Greatest Activity.

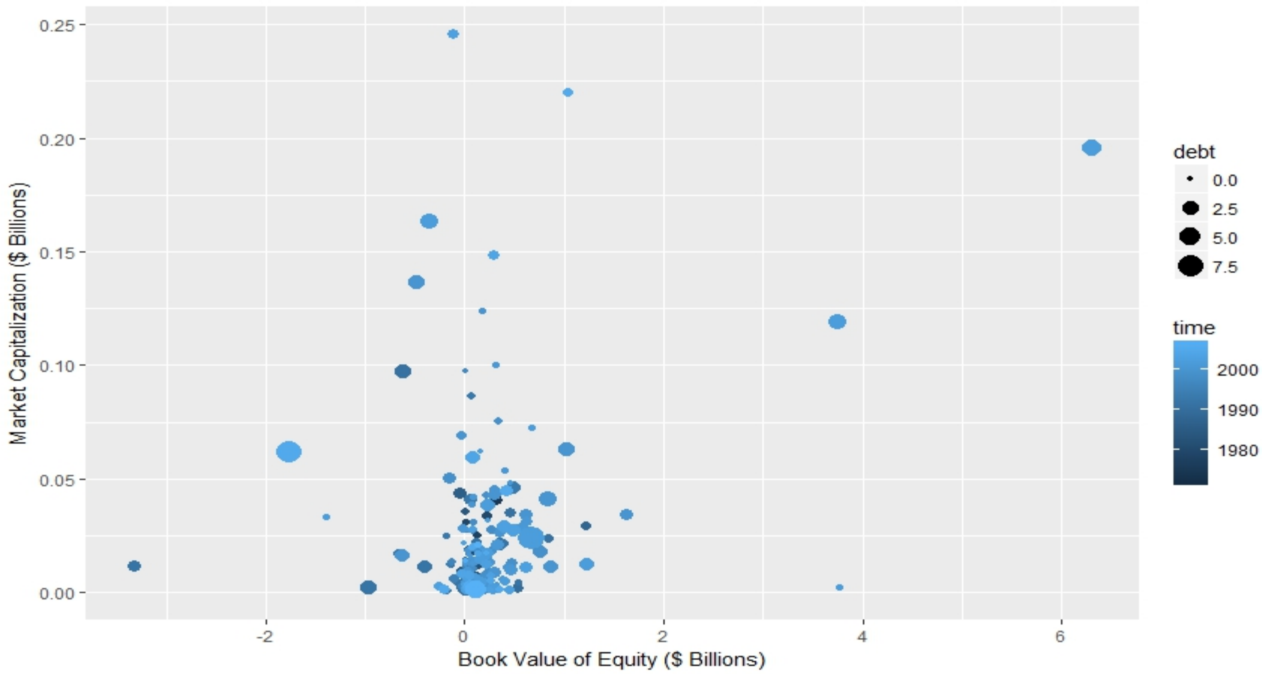


Figure 4: Histogram showing corporate bankruptcies: periods of "low" activity are followed by periods of "high" activity.

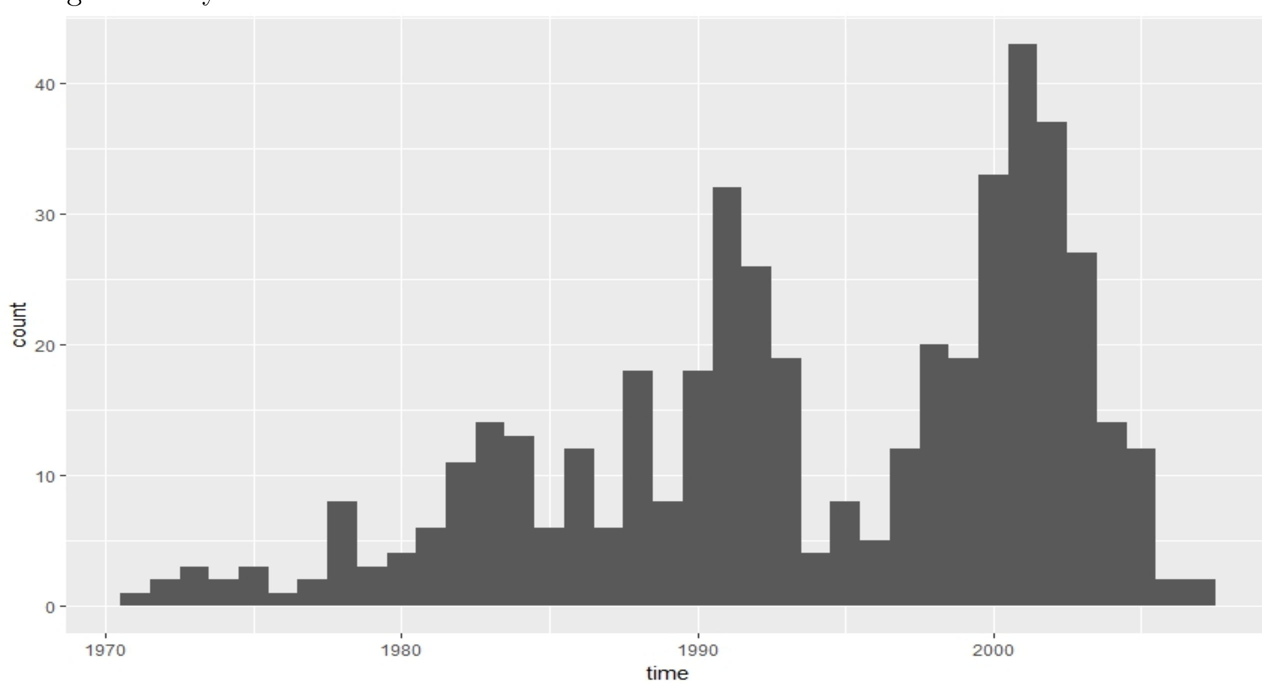


Figure 5: F, G and J functions indicate clustering or inhomogeneity. Data: red line; Poisson: green line.

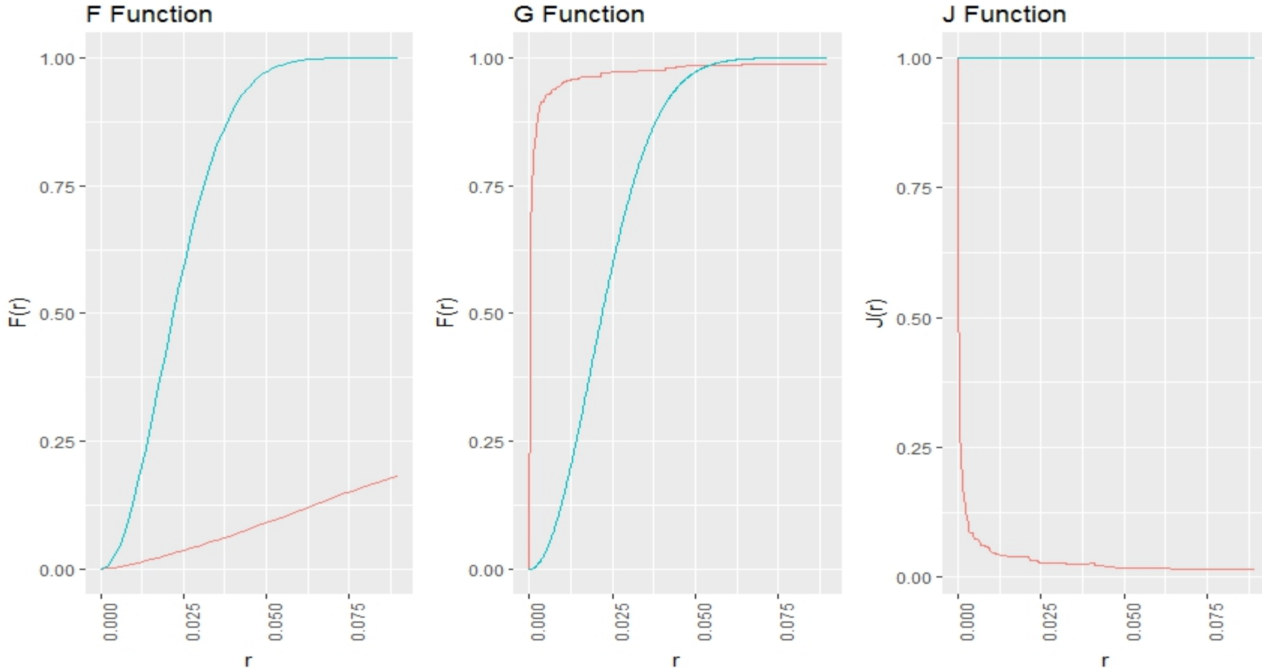


Figure 6: K function also indicates clustering or inhomogeneity. Data: red line; Poisson: green line.

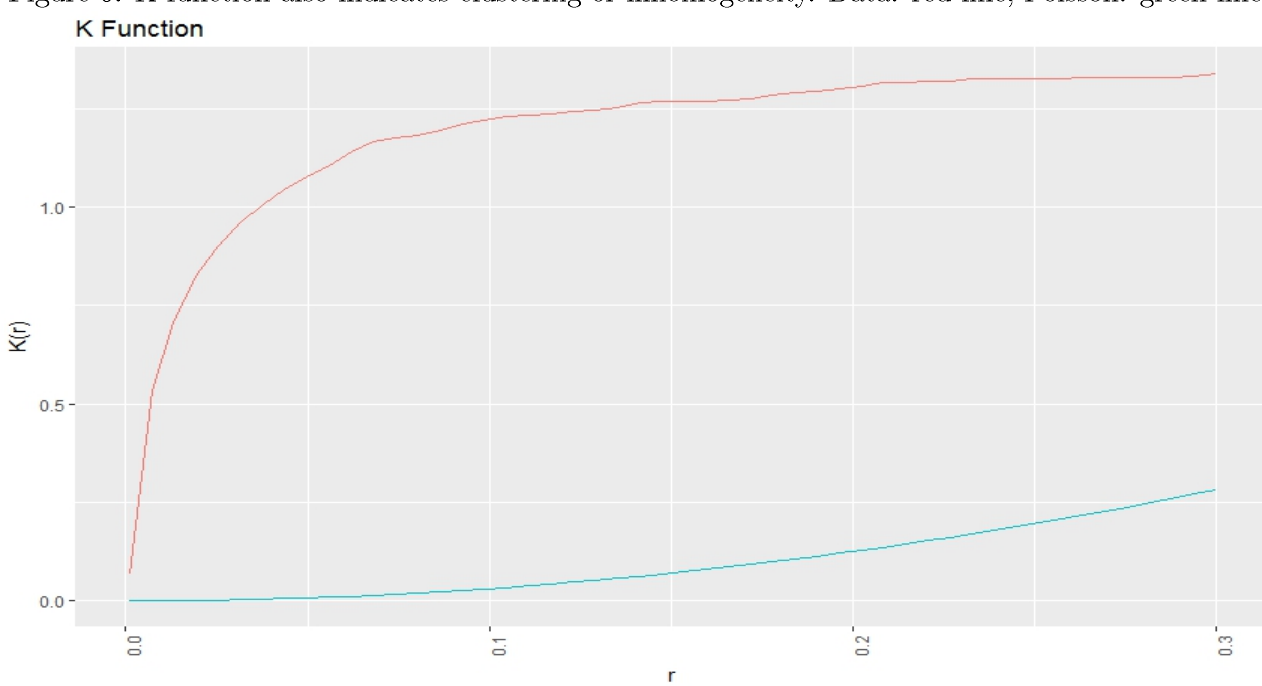


Figure 7: ...and so does the L function. Data: red line.

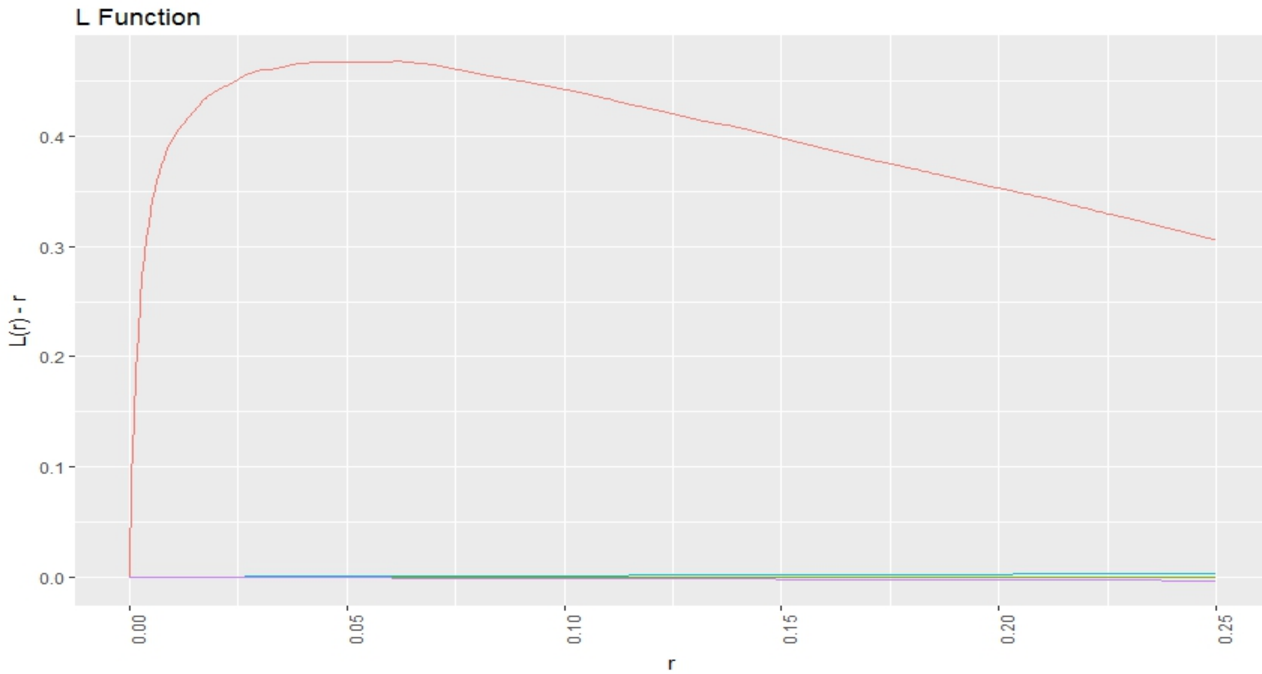


Figure 8: Marked G Function: Probability that a point with $m \leq 0.5$ is within distance r of point with $m \geq 0.8$. Data: Red; Poisson: Green.

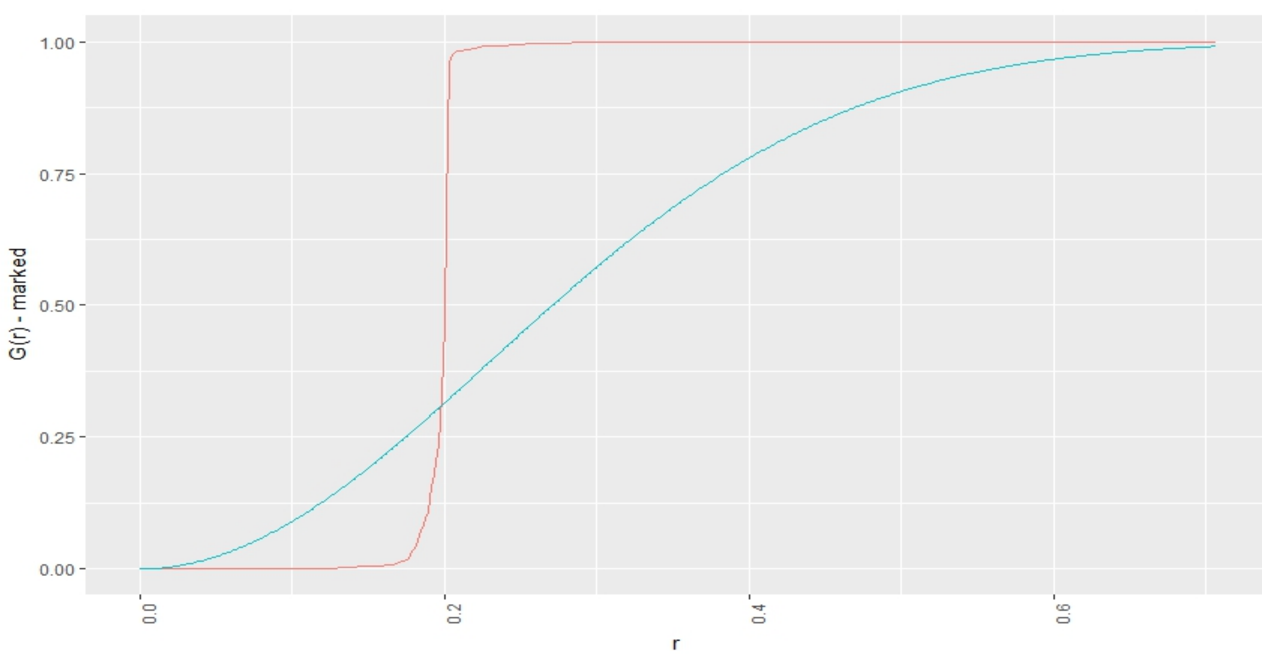


Figure 9: Kernel estimation of the Papangelou intensity: intensity is too small in "active region".

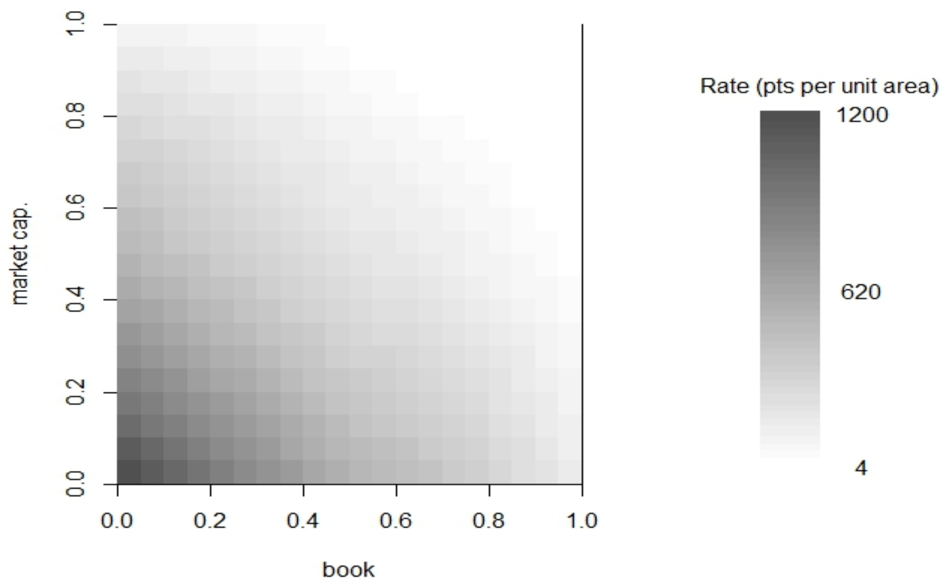


Figure 10: Papangelou Intensity estimated via Pseudo-Loglikelihood. Model: $\lambda(x, y) = 539.17 - 50.27x + 50.27y - 341.42xy$; SE = [98.44,1553.06,385.40,1977.14]

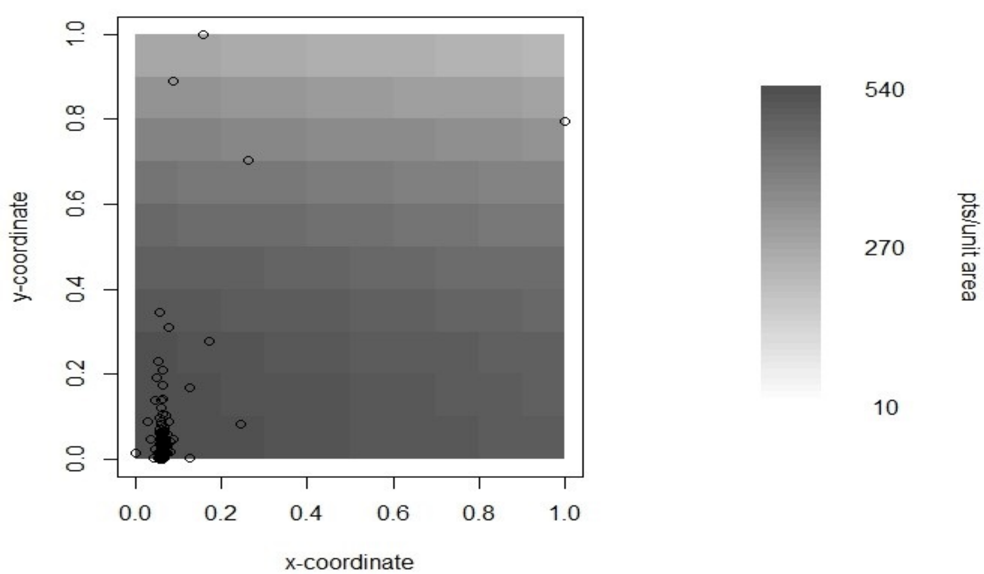


Table 1: Coefficient estimates and standard errors for ETAS Model. The background rate decreases with market capitalization and book value, and increases with time. The expected number of first generation aftershocks is $E[h(m)]$, which we estimate to be 0.9753.

$\bar{\mu}$	a	b	c	d	α	β	K
2.052	-11.450	-19.775	18.837	16.667	62.990	0.016	0.975
(60.252)	(24.660)	(24.257)	(38.759)	(39.229)	(8.974)	(0.849)	(0.052)

Figure 11: Temporal Triggering Function - g(t). Bankruptcies help trigger bankruptcies up to 20 months apart.

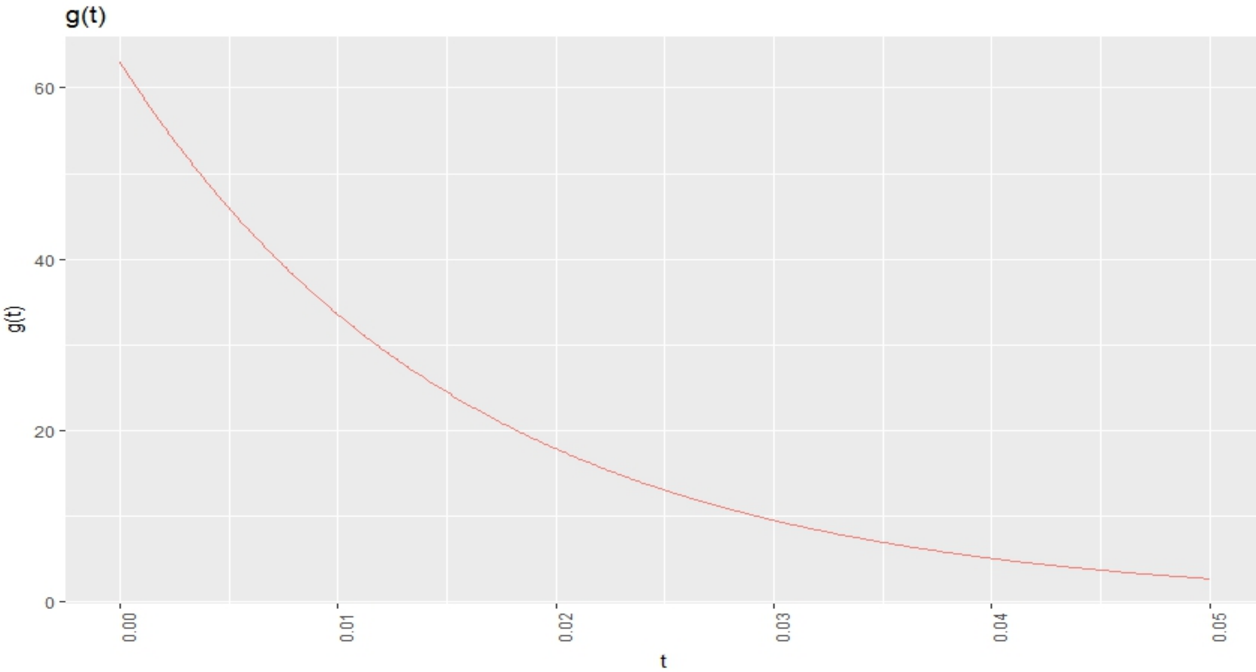


Figure 12: Superthinning - Model "overpredicts" in [0,2]x[0,2]. Green points: original data.

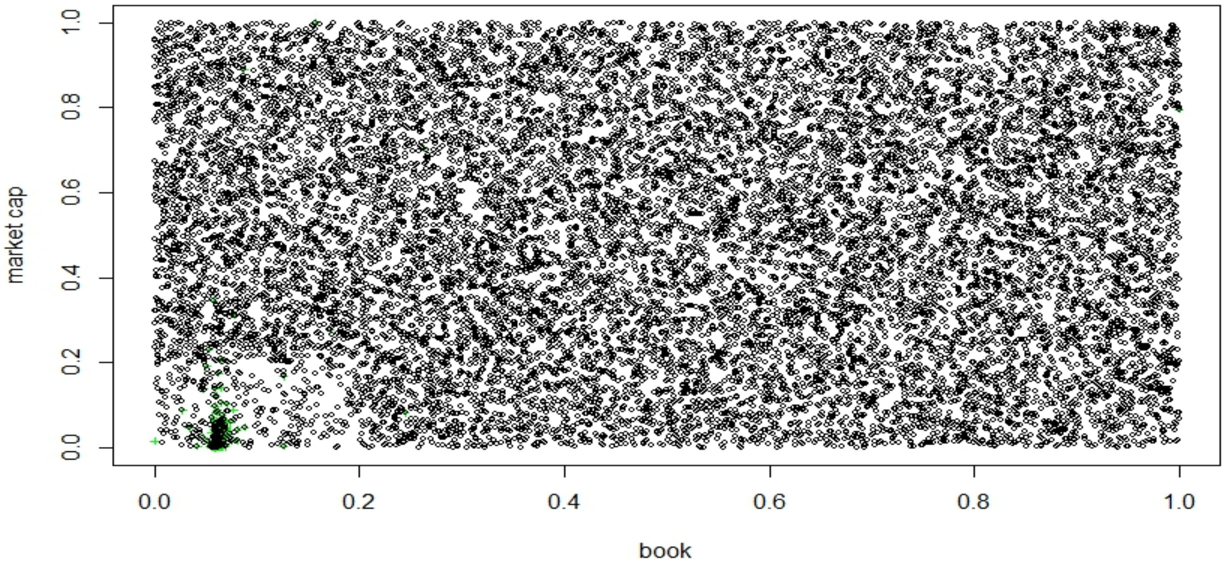
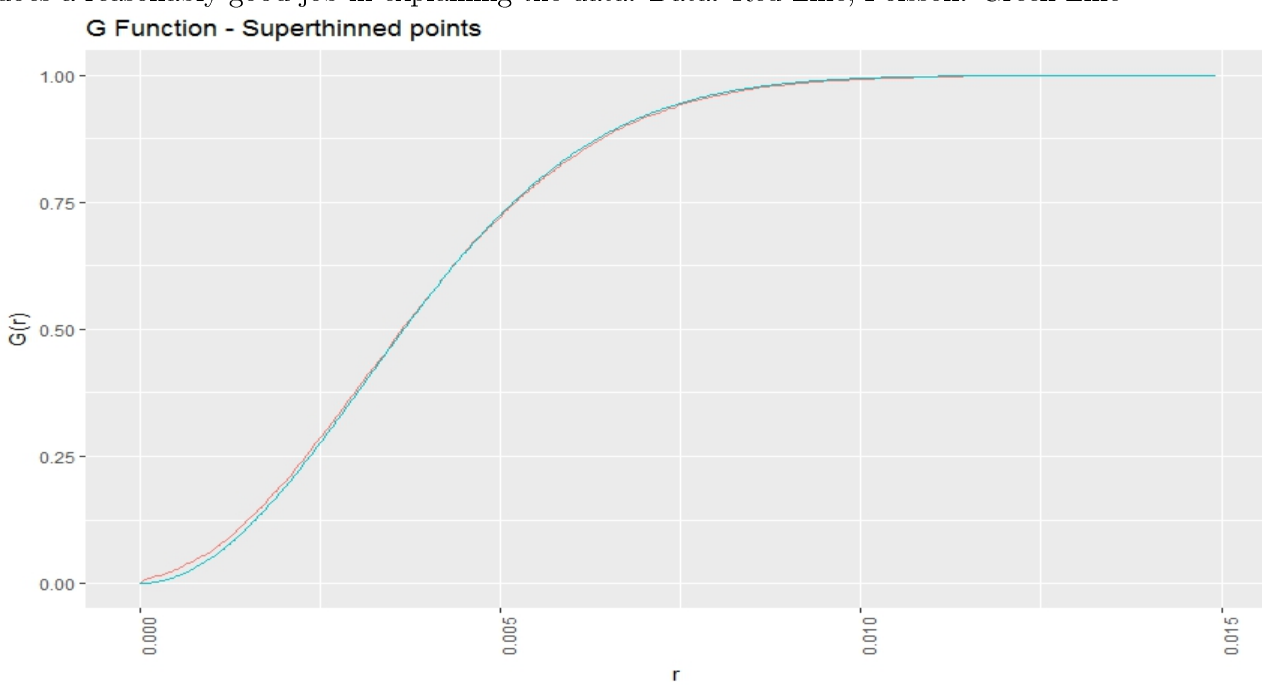


Figure 13: G Function of Superthinned Points - the G function suggests here that our calibrated model does a reasonably good job in explaining the data. Data: Red Line; Poisson: Green Line



5.2 R Code

```r
library(data.table) # manipulate data
library(plyr) # manipulate data
library(DataCombine) # manipulate data
library(reshape2) # manipulate data
library(stargazer) # make nice-looking tables
library(ggplot2) # create nice-looking graphs
library(splancs)
library(spatstat)
library(gridExtra) # plot multiple graphs with ggplot (like par(mfrow) command)
# library(pracma)
# library(moments)

######READ IN DATA## #####################################
##########################################################

# READING IN CRSP DATA#####################################

setwd("C:/users/rafae/desktop/pdf/stats 222")

coalesce <- function(...) {
  apply(cbind(...), 1, function(x) {
    x[!is.na(x)][1]
  })
}

# read the data
crsp <- read.csv("all_crsp.csv", header=TRUE)

if (is.null(crsp$date)) {
  crsp$date <- crsp$DATE
  crsp$DATE <- NULL
}

# separate YYYYMMDD dates into years, months and days
crsp$year <- as.numeric(substr(as.character(crsp$date), 1, 4))

# keep only stocks listed on NYSE (1), AMEX (2) or NASDAQ (3)
crsp <- crsp[crsp$EXCHCD >= 1 & crsp$EXCHCD <= 3 & !is.na(crsp$EXCHCD),]

# remove any duplicate observations for a given stock in a given month
crsp <- crsp[!duplicated(crsp[, c("date", "PERMNO")]),]

# keep only ordinary stocks (share codes 10 and 11)
crsp <- crsp[(crsp$SHRCD == 11 |crsp$SHRCD == 10)  & !is.na(crsp$SHRCD),]

crsp$PRC = abs(crsp$PRC)
crsp$SHROUT = abs(crsp$SHROUT)

crsp$SHROUT[crsp$SHROUT==0] = NA # if shares outstanding are 0, we replace the number with NA

crsp$DLSTCD[is.na(crsp$DLSTCD)] = 0

crsp_bankrupt = crsp[crsp$DLSTCD == 574,]

ids = matrix(unique(crsp_bankrupt$PERMNO),ncol = 1)

```

```r
ids = data.frame(PERMNO = matrix(unique(crsp_bankrupt$PERMNO),ncol = 1))

crsp_subset = merge(ids,crsp, allx = TRUE)
crsp_subset = arrange(crsp_subset,PERMNO,date)

# remove any characters from key values

destring <- function(x, columns=names(crsp)) {
  tmp <- x
  tmp[, columns] <- suppressWarnings(lapply(lapply(x[, columns], as.character), as.numeric))
  return (tmp)
}

crsp_subset <- destring(crsp_subset, c("PRC", "SHROUT","DLSTCD"))

# obtain most recent price, prior to bankruptcy filing

crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -1,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -2,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -3,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -4,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -5,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -6,keepInvalid = TRUE)
crsp_subset = slide(crsp_subset, Var = "PRC", GroupVar = "PERMNO", slideBy = -7,keepInvalid = TRUE)

crsp_subset = slide(crsp_subset, Var = "SHROUT", GroupVar = "PERMNO", slideBy = -1,keepInvalid = TRUE)

crsp_clean = crsp_subset[crsp_subset$DLSTCD == 574,]
crsp_clean$PRICE = coalesce(crsp_clean[,"PRC-1"],crsp_clean[,"PRC-2"],crsp_clean[,"PRC-3"],
                            crsp_clean[,"PRC-4"],crsp_clean[,"PRC-5"],crsp_clean[,"PRC-6"],
                            crsp_clean[,"PRC-7"])

crsp_clean$SHARES = crsp_clean[,"SHROUT-1"]


# keep only variables we will use

col_select = c("PRICE","SHARES","date","PERMNO","year","HSICCD","HSICMG","COMNAM")
crsp_clean = crsp_clean[,col_select]

remove(crsp)
remove(crsp_subset)


##########################################################
## Read-in Accounting Data from COMPUSTAT Database ########

compustat <- read.csv("compustat_project.csv", header=TRUE)

# book value and debt
compustat$book = coalesce(compustat$se,compustat$ce)
compustat$debt = coalesce(compustat$dt,compustat$dltt + compustat$lct,compustat$dltt,compustat$lct)

compustat$date = compustat$datadate
compustat$datadate = NULL

# separate YYYYMMDD dates into years, months and days
compustat$year <- as.numeric(substr(as.character(compustat$date), 1, 4))
```

```
compustat$month <- as.numeric(substr(as.character(compustat$date), 5, 6))
compustat$day <- as.numeric(substr(as.character(compustat$date), 7, 8))

# remove any duplicate observations for a given stock in a given year
compustat <- compustat[!duplicated(compustat[, c("date", "LPERMNO")]),]

# keep only variables we will use
col_select = c("book","LPERMNO","date","year","debt")
compustat_subset = compustat[,col_select]
remove(compustat)

compustat_subset = rename(compustat_subset,replace = c("LPERMNO"= "PERMNO"))

compustat_subset$year = compustat_subset$year +2

########## Merge CRSP and Compustat data##################

merged = merge(crsp_clean,compustat_subset,all.x = TRUE, by = c("PERMNO","year"))
merged <- merged[!is.na(merged$date.y),] # no match in compustat database was found

merged$date.y = NULL
merged = rename(merged,replace = c("date.x" = "date"))
merged$mkt_cap = merged$PRICE*merged$SHARES/1000000 # market caps are in $ billions of dollars
merged$book = merged$book/1000 # book value is in $ billions of dollars as well

merged = arrange(merged,desc(debt))

data = merged
data = data[is.na(data$book) == FALSE & is.na(data$mkt_cap) == FALSE &
              is.na(data$debt) == FALSE,]
data$debt = data$debt/1000

data$month = as.numeric(substr(as.character(data$date), 5, 6))

dates = data$date

year = as.numeric(substr(as.character(dates), 1, 4))
month = as.numeric(substr(as.character(dates), 5, 6))
day = as.numeric(substr(as.character(dates), 7, 8))

toDate <- function(year, month, day) {
  ISOdate(year, month, day)
}

t = as.numeric(toDate(year,month,day))
MINT = min(t)
MAXT = max(t)
t = (t-MINT)/(MAXT-MINT)
data$t = t

data = arrange(data,t)

############################################################
########Adding time and spread covariate (spread)##########

book = data$book
market = data$mkt_cap
debt = data$debt
```

```
m = (debt-min(debt))/(max(debt)-min(debt))
time = data$year
t = data$t
T = 1

setwd("C:/users/rafae/desktop/ATC")

AAA = read.csv("AAA.csv", header=TRUE)
BAA = read.csv("BAA.csv", header=TRUE)
def = data.frame(cbind(AAA[,1],BAA[,-1]-AAA[,-1]))
def[,1] = as.Date(def[,1],origin="1899-12-30")
def = slide(def, Var = "X2", slideBy = -1)
def = def[def$X1>= "1971-01-01" & def$X1 <= "2007-12-31",]
def[,2]=NULL

def$year = as.numeric(substr(as.character(def[,1]), 1, 4))
def$month = as.numeric(substr(as.character(def[,1]), 6, 7))
def[,1] = NULL

colnames(def) = c("spread","year","month")

data = merge(data,def,all.x = TRUE, by = c("year","month"))

spread = data$spread
n = length(book)


###ANALYSIS################################################
##########################################################
##########################################################
##########################################################


### Creating market cap x book plots

# Dividing-up data
window1 = data.frame(Beggining = 0, End = 2)
window2 = data.frame(Beggining = 2, End = 10)
window3 = data.frame(Beggining = 10, End = 20)

dataplot = data.frame(book = seq(0,20,0.1), market = seq(0,20,0.1))

# Creating Figure 1
plot = ggplot(dataplot)  + geom_line(aes(x=book, y=market)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_rect(data=window1, aes(ymin=Beggining, ymax=End, xmin=-Inf, xmax=+Inf),
            fill='blue', alpha=0.1)  +
  geom_rect(data=window2, aes(ymin=Beggining, ymax=End, xmin=-Inf, xmax=+Inf),
            fill='green', alpha=0.1)  +
  geom_rect(data=window3, aes(ymin=Beggining, ymax=End, xmin=-Inf, xmax=+Inf),
            fill='yellow', alpha=0.1) +
  ylab("Market Capitalization ($billions)") + xlab("Book Value of Equity ($billions)")


### (Purely) Spatial Analysis #############################



# Creating Figure 2
```

```
data_plot = ggplot(dataframe,aes(book,market))
data_plot + geom_point(aes(size = debt,colour = time)) +
  ylab("Market Capitalization ($ Billions)") +
  xlab("Book Value of Equity ($ Billions)")


# Creating Figure 3
slice = (book<= 10 & market <= 0.3)
book_slice = book[slice]
market_slice = market[slice]
time_slice = time[slice]
debt_slice = debt[slice]

dataslice = data.frame(book = book_slice,market = market_slice, debt = debt_slice,
                       time = time_slice)

data_plot = ggplot(dataslice,aes(book,market))
data_plot + geom_point(aes(size = debt,colour = time)) +
  ylab("Market Capitalization ($ Billions)") +
  xlab("Book Value of Equity ($ Billions)")


# Creating Figure 4
dataframe = data.frame(book = book,market = market, debt = debt,
                       time = time)

ggplot(dataframe, aes(time, fill = debt)) +
  geom_histogram(binwidth = 1)


# Re-scaling our data
book = (book-min(book))/(max(book)-min(book))
market = (market-min(market))/(max(market)-min(market))

###  Kernel smoothing
bdw = 0.8
b1 = as.points(book,market)
bdry = matrix(c(0,0,1,0,1,1,0,1,0,0),ncol=2,byrow=T)
z = kernel2d(b1,bdry,bdw)
par(mfrow=c(1,2))
image(z,col=gray((64:20)/64),xlab="book",ylab="market cap.")
# points(b1)
x4 = (0:100)/100*(max(z$z)-min(z$z))+min(z$z)
plot(c(0,10),c(.8*min(x4),1.2*max(x4)),type="n",axes=F,xlab="",ylab="")
image(c(-1:1),x4,matrix(rep(x4,2),ncol=101,byrow=T),add=T,col=gray((64:20)/64))
text(2,min(x4),as.character(signif(min(x4),2)),cex=1)
text(2,(max(x4)+min(x4))/2,as.character(signif((max(x4)+min(x4))/2,2)),cex=1)
text(2,max(x4),as.character(signif(max(x4),2)),cex=1)
mtext(s=3,l=-3,at=1,"Rate (pts per unit area)")

# Creating F,G,J,K and L functions

#### F-function (empty-space function):
b2 = as.ppp(b1, W = c(0,1,0,1))
## the above convert the points into a "ppp" object,
## using as a window [0,1] x [0,1]

f = Fest(b2,correction = 'none')
```

```
data.plot <- data.frame(r = f$r,Data = f$raw,Poisson = f$theo)

# Plotting f function
data_long <- melt(data.plot, id= "r")

plot1 = ggplot(data=data_long,
       aes(x=r, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="F Function") +ylab("F(r)") +xlab("r")

#### G-function:
g = Gest(b2,correction = "none")
data.plot <- data.frame(r = g$r,Data = g$raw,Poisson = g$theo)

data_long <- melt(data.plot, id= "r")

plot2 = ggplot(data=data_long,
       aes(x=r, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="G Function") +ylab("F(r)") +xlab("r")

#### J-function:
j = Jest(b2,correction = 'none')
data.plot <- data.frame(r = j$r,Data = j$un,Poisson = j$theo)

data_long <- melt(data.plot, id= "r")

plot3 = ggplot(data=data_long,
       aes(x=r, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="J Function") +ylab("J(r)") +xlab("r")


grid.arrange(plot1, plot2,plot3,nrow = 1, ncol=3)

# Plotting K an L Functions ##############################

k = Kest(b2,correction = "none")

data.plot <- data.frame(distance = k$r,Data = k$un,poisson = K$theo)

data_long <- melt(data.plot, id= "distance")

ggplot(data=data_long,
               aes(x=distance, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="K Function") +ylab("K(r)") +xlab("r")

l = Lest(b2,correction = "none")
s = seq(.001,.3,length=length(l$r))

Lupper = 1.96 * sqrt(2*pi*1*1) * s / n
Llower = -1.0 * Lupper

```

```
data.plot <- data.frame(distance = l$r,Data = l$un - l$r,poisson = l$theo - l$r,lupper = Lupper,llower = Llower)

data_long <- melt(data.plot, id= "distance")

ggplot(data=data_long,
       aes(x=distance, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="L Function") +ylab("L(r) - r") +xlab("r")

### Fitting Pseudo-Likelihood model.
# The model we fit is lambda(x,y) = mu + ax +by +cxy

n1 = length(book)

x1 = book
y1 = market

f = function(p){
  ## returns the negative pseudo log-likelihood
  ## p = (mu,a,b,c)

  if(p[1] < 0) return(99999)
  if(p[1] + p[2] < 0) return(99999)
  if(p[1] + p[3] < 0) return(99999)
  if(p[1] + p[4] < 0) return(99999)
  if(p[2] + p[3] < 0) return(99999)
  if(p[1] + p[3] < 0) return(99999)
  if(p[1] + p[3] < 0) return(99999)
  if(p[1] + p[2] + p[3] + p[4] < 0) return(99999)
  if(p[1] + p[2] + p[3] < 0) return(99999)

  lam = p[1] + p[2] * x1 + p[3] * y1 +p[4]*x1*y1

  #lam = p[1] + p[2] * x1 + p[3] * y1

  if (min(lam) < 0) return (99999)

  int2 = p[1] + p[2]/2 + p[3]/2 +p[4]/4
 # int2 = p[1] + p[2]/2 + p[3]/2

  cat("integral = ",int2," negative loglikelihood = ",
      int2-sum(log(lam)), "\n"," p = ",p,"\n")
  ## integral should be roughly n when it's done
  return(int2-sum(log(lam)))
}



pstart = c(0.1, 0.1, 0.1, 0.1)

fit1 = optim(pstart,f,control=list(maxit=200),hessian = TRUE)
pend = fit1$par
b3 = sqrt(diag(solve(fit1$hess)))


### Plot the Model's Background Rate
par(mfrow=c(1,2))
```

```
plot(c(0,1),c(0,1),type="n",xlab="x-coordinate",ylab="y-coordinate",
     main="background rate")
x2 = seq(0.05,0.95,length=10)
y2 = seq(0.05,0.95,length=10)
z2 = matrix(rep(0,(10*10)),ncol=10)
#z3 = matrix(rep(0,(10*10)),ncol=10)
for(i in 1:10){
  for(j in 1:10){
    z2[i,j] = pend[1] + pend[2]*x2[i] + pend[3]*y2[j] +pend[4]*y2[j]*x2[j]
 #   z3[i,j] = pstart[1] + pstart[2]*x2[i] + pstart[3]*y2[j]
  }}
#zmin = min(c(z2,z3))
#zmax = max(c(z2,z3))
zmin = min(z2)
zmax = max(z2)


image(x2,y2,z2,col=gray((64:20)/64),zlim=c(zmin,zmax),add=T)
points(x1,y1)

######### LEGEND:
zrng = zmax - zmin
zmid = zmin + zrng/2
plot(c(0,10),c(zmid-2*zrng/3,zmid+2*zrng/3),type="n",axes=F,xlab="",ylab="")
zgrid = seq(zmin,zmax,length=100)
## zgrid = vector of 100 equally-spaced numbers spanning range of the values.
image(c(-1:1),zgrid,matrix(rep(zgrid,2),ncol=100,byrow=T),add=T,col=gray((64:20)/64))
text(2.5,zmin,as.character(signif(zmin,2)),cex=1)
text(2.5,zmax,as.character(signif(zmax,2)),cex=1)
text(2.5,zmid,as.character(signif(zmid,2)),cex=1)
text(4.5,zmid,"pts/unit area",srt=-90)

# plot(c(0,1),c(0,1),type="n",xlab="-coordinate",ylab="y-coordinate",
#     main="original guess")
#image(x2,y2,z3,col=gray((64:20)/64),zlim=c(zmin,zmax),add=T)
#points(x1,y1)


### Marked G-function:

b222 = as.ppp(cbind(x1,y1), W = c(0,1,0,1))
b222$marks = m
b222$n = n1

## the above convert the points into a "marked ppp" object,
## using as a window [0,1] x [0,1]

par(mfrow=c(1,1))
gm = Gmulti(b222, b222$marks < 0.5, b222$marks > 0.8,correction = "none")

data.plot <- data.frame(r = gm$r,Data = gm$raw,Poisson = gm$theo)

# Plotting marked G function
data_long <- melt(data.plot, id= "r")

plot1 = ggplot(data=data_long,
               aes(x=r, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
```

```
  geom_line() +
  labs(title="") +ylab("G(r) - marked") +xlab("r")
```

```

###FITTING ETAS MODEL################################################################
library(GenSA)

m3 = function(x) signif(x,3)

x = book
y = market

logl = function(theta){

  mu_bar = theta[1]; a = theta[2]; b = theta[3];c = theta[4];
  d = theta[5]; alpha = theta[6];
  beta = theta[7]; K = theta[8]

  cat("\n mu_bar = ",m3(mu_bar),", K = ",m3(K),", a = ",m3(a),", b = ",m3(b),", c = ",m3(c),", d = ",m3(d),
      ", alpha = ",m3(alpha),", beta = ",m3(beta),".\n")

  if(min(min(mu_bar+a*x+b*y+c*t+d*spread),K,alpha,beta,d)<0.000000001) return(99999)
  if(min(mu_bar +a*1 +c*t +d*spread) < 0.000000001) return(99999)
  if((K*mean(exp(beta*m)))>.99999) return(99999)

  mu_xyt = mu_bar + a*x + b*y +c*t +d*spread

  intlam = mu_bar*T +a/2 +b/2 +c/2 +d*mean(spread) + K*sum(exp(beta*(m)))

  g = vector(length = n)
  lam = vector(length = n)
  lam[1] = mu_xyt[1]

  for(j in 2:n){

    gij = 0

    if(x[j]<= 0.2 & y[j]<= 0.2){


    for(i in 1:(j-1)){

      gij = gij + alpha*exp(-alpha*(t[j]-t[i]))*K*exp(beta*(m[i]))

    }

    }

    lamj = mu_xyt[j] + gij*(1/(0.2*0.2))

    if(lamj <= 0){
      cat("lambda ",j," is less than 0.")
      return(99999)
    }

    lam[j] = lamj

  }
```

```

  sumlog = sum(log(lam))

  loglik = sumlog - intlam
  cat("loglike is ", loglik, ". sumlog = ", sumlog,". integral = ", intlam,".\n")

  return(-1.0*loglik)
}


theta1 = rep(0.3,8)

seed = 3000


#FITTING MODEL##########################################################
# Start the clock!
ptm <- proc.time()

b1 = optim(theta1,logl)

# Stop the clock
proc.time() - ptm

# Start the clock!
ptm <- proc.time()

b2 = optim(b1$par,logl,hessian=T)

# Stop the clock
proc.time() - ptm

theta2 = b2$par
sqrt(diag(solve(b2$hess))) ## for SEs

# COMPARE FIT WITH THAT GIVEN BY SIMULATED ANNEALING####################
b1_new = GenSA(theta1, fn = logl, lower = c(rep(-1000,8)), upper = c(rep(1000,8)),
               control=list(max.time = 345.25))
```

```

########Superthinning################################################

z = data.frame(lat = market,lon= book,t= t,m= m,spread= spread,n= n )

theta = b2$par

mu_bar = theta[1]; a = theta[2]; b = theta[3];c = theta[4];
d = theta[5]; alpha = theta[6];
beta = theta[7]; K = theta[8]


find_spread = function(candt){
# this function retrieves the value of the spread covariate on any date

  t_actual = MINT +candt*(MAXT - MINT)

  date_obs = as.Date(t_actual/86400, origin = "1970-01-01")
  year_obs = year(date_obs)
```

```
  month_obs = month(date_obs)

  t_actual = data.frame(year = year_obs,month = month_obs)
  covariate = merge(t_actual, def, allx = TRUE,by = c("year","month"))

  return(covariate$spread)

}



compute_lambda_data = function(t,x,y,z,spread){
  # This function computes the value of lambda
  # for each point in our data

  mu_xyt = mu_bar + a*x + b*y +c*t +d*spread

  g = vector(length = n)
  lam = vector(length = n)
  lam[1] = mu_xyt[1]

  for(j in 2:n){

    gij = 0

    if(x[j]<= 0.2 & y[j]<= 0.2){


    for(i in 1:(j-1)){

      gij = gij + alpha*exp(-alpha*(t[j]-t[i]))*K*exp(beta*(m[i]))

    }

    }

    lamj = mu_xyt[j] + gij*(1/(0.2*0.2))

    if(lamj <= 0){
      cat("lambda ",j," is less than 0.")
      return(99999)
    }

    lam[j] = lamj

  }

  return(lam)
}


fun_comp = function(cant,canx,cany,spread,z){
  # compute lambda(t,x,y) given data, z,
  # for a given candidate point.

  gij = 0
  j = 0
  if(cant > z$t[1]) j = max(c(1:z$n[1])[z$t<cant])
```

```
  if(j>0 & canx <=0.2 & cany<=0.2) for(i in 1:j){


    gij = gij + alpha*exp(-alpha*(cant-z$t[i]))*K*exp(beta*(z$m[i]))

    #gij = gij + exp(-beta*(t-z$t[i])-alpha*r2)
  }

  return(mu_bar + a*canx + b*cany +c*cant +d*spread +gij*(1/(0.2*0.2)))


}



supthin = function(z,lambda,fun,thresh=median(lambda)){
  ## z = data, lambda = conditional intensity at pts,
  # fun = function to compute lambda,
  ## and thresh = resulting rate.
  ## First thin, then superpose

  keepz = list()
  for(i in 1:z$n[1]){
    if(runif(1) < thresh/lambda[i]){
      keepz$t = c(keepz$t,z$t[i])
      keepz$lon = c(keepz$lon,z$lon[i])
      keepz$lat = c(keepz$lat,z$lat[i])
    }
  }
  candn = rpois(1,thresh*X1*Y1*T)
  candt = sort(runif(candn)*T)
  candx = runif(candn)*X1
  candy = runif(candn)*Y1

  cov = find_spread(candt)

  for(i in 1:candn){
    v = fun_comp(candt[i],candx[i],candy[i],cov[i],z)
    if(v < thresh){
      if(runif(1) < (thresh-v)/thresh){
        keepz$t = c(keepz$t,candt[i])
        keepz$lon = c(keepz$lon,candx[i])
        keepz$lat = c(keepz$lat,candy[i])

            }}
    }
  keepz$lon = keepz$lon[order(keepz$t)]
  keepz$lat = keepz$lat[order(keepz$t)]
  keepz$t = sort(keepz$t)
  keepz$n = length(keepz$t)
  keepz
}

lambda = compute_lambda_data(t,x,y,z,spread)

# Performing Superthinning ####################################
s = supthin(z,lambda,fun_comp)
par(mfrow=c(1,2))
```

```
plot(z$lon,z$lat,pch=3,cex=.5,xlab="book",ylab="market cap.",main="original pts.")
plot(s$lon,s$lat,pch=3,cex=.5,xlab="book",ylab="market cap",main="superthinned points")
par(mfrow=c(1,1))
plot(z$lon,z$lat,pch=3,cex=.5,xlab="book",ylab="market cap",col="green")
points(s$lon,s$lat,pch=1,cex=.5,xlab="book",ylab="market cap")
```

```

#Plotting g(t) - temporal triggering #################

g_func = function(alpha,t){

  alpha*exp(-alpha*t)
}


data.plot <- data.frame(distance = seq(0,0.05,0.0001),Data = g_func(alpha,seq(0,0.05,0.0001)))

data_long <- melt(data.plot, id= "distance")

ggplot(data=data_long,
       aes(x=distance, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="g(t)") +ylab("g(t)") +xlab("t")


#Plotting G function for superthinned points
s$lon #book
s$lat #market

#superthinned points

b11 = as.points(s$lon,s$lat)
b22 = as.ppp(b11, W = c(0,1,0,1))

g = Gest(b22,correction = "none")

data.plot <- data.frame(distance = g$r,Data = g$raw,poisson = g$theo)

data_long <- melt(data.plot, id= "distance")

ggplot(data=data_long,
       aes(x=distance, y=value, colour=variable)) +
  theme(legend.position="none",axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_line() +
  labs(title="G Function - Superthinned points") +ylab("G(r)") +xlab("r")
```