2019 Pitching Spatial Data Analysis

Azim Ali

06/01/2022

```
# load library
library(ggplot2)
library(gridExtra)
library(tidyverse)
library(xts)
library(dplyr)
library(spatstat)
library(sf)
library(units)
```

Data source and parsing method:

The data was taken from 2019 MLB pitching data (https://www.kaggle.com/pschale/mlb-pitch-data-20152018 (https://www.kaggle.com/pschale/mlb-pitch-data-20152018) by Paul Schale), which was scraped from http://gd2.mlb.com/components/game/mlb/ (http://gd2.mlb.com/components/game/mlb/), that has information (speed, break angle, pitch call [called strike, ball, foul, etc.], etc.), pitch location (px and pz), and categories (pitch type) of all pitches thrown in 2019.

Dataset filtering explanation:

I filtered the data to show all umpire called strike pitches (batter did not swing), high break angle (≥ 30), distinct at bat ID, and within the average strike zone window ($px = (-0.8, 0.8)$ and $pz = (1.5, 3.5)$). In reality, the strike zone is adjusted for all batters due to the definition of the strike zone being dependent on the shoulder height to waist height. Also, umpires vary their called strikes across the strike zone. These are the reasons why the data was filtered for called strike pitches.

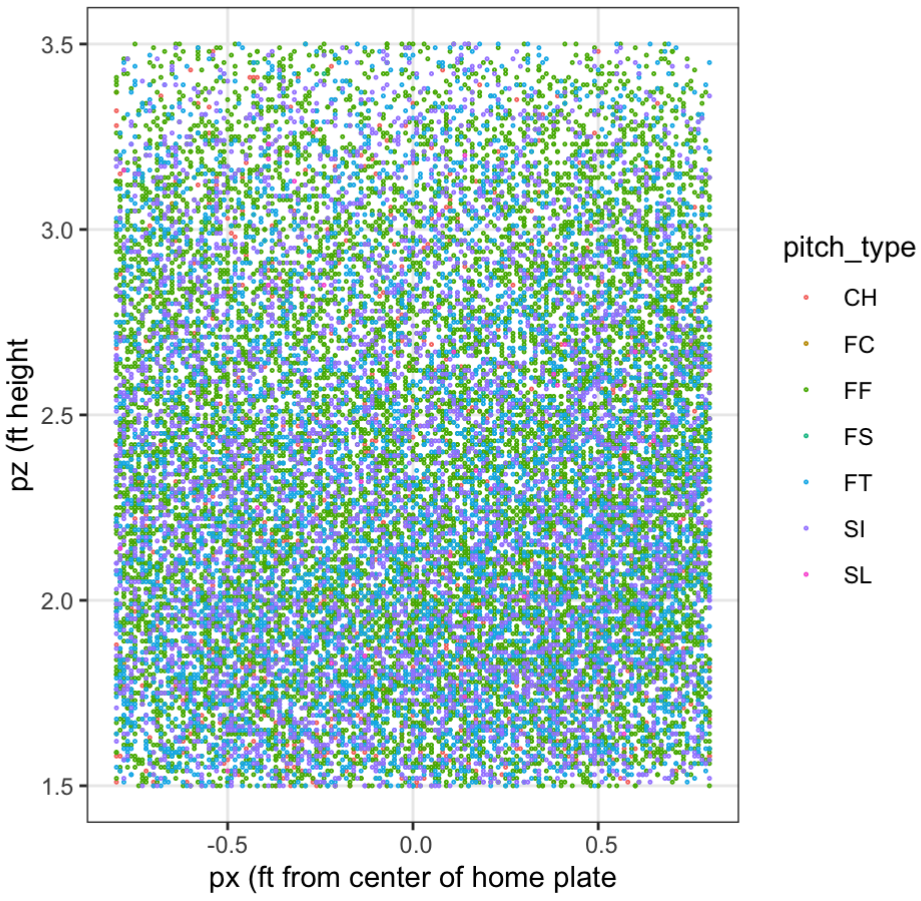
```
# Import 2019 MLB pitches data
pitches2019 <- read.csv("2019_pitches.csv")
```

```
# Filter the data for average strike zone (-0.80, 0.80 px), (1.5, 3.5 pz)
pitches2019avgstrikzone <- pitches2019 %>%
  dplyr::select(px, pz, start_speed, end_speed, break_angle, break_length, code, type, p
itch_type, ab_id) %>%
  filter(px>=-0.80 & px <= 0.80, pz <= 3.5 & pz >= 1.5, type=="C", break_angle >= 30) %
>% # Filter for high break angle, window, and called strike pitches
  drop_na(px, pz, pitch_type, type, code) %>%
  add_count(ab_id) %>%
  filter(n==1) # Filter for unique at bats
pitches2019avgstrikzone_sf <- st_as_sf(pitches2019avgstrikzone, coords = c("px", "pz"))

pitches2019avgstrikzone2 <- pitches2019avgstrikzone %>%
  dplyr::select(px, pz)
W<-owin( c(-0.8, 0.8), c(1.5, 3.5) )
pp1 <- as.ppp(pitches2019avgstrikzone2, W = W)
```

```
ggplot() + geom_sf(data = pitches2019avgstrikzone_sf, aes(col=pitch_type), size=0.2, pch
=21) +
  ggtitle("Figure 1. High break angle strike zone pitch locations.") +
  xlab("px (ft from center of home plate") +
  ylab("pz (ft height") +
  theme_bw()
```

Figure 1. High break angle strike zone pitch locations.



Discussion stoichastic process of the pitching data:

This set of points can be thought of as being generated from a stochastic process because a pitch can theoretically be pitched at any location within the average strike zone. That means that a high bank angle pitch can theoretically be pitched at any coordinate location in the average strike zone and thus the data is a SPP.

Thoughts on second-order dispersion interaction between pitches in the dataset:

It is known that pitches in the same at bat can have a dispersion interaction property (second-order) associated with previous pitch locations due to the fact that a pitcher does not want to consistently pitch in the same location to keep the current batter guessing. Under the same logic, clustering or interactive property (second-order) can exist when a pitch has a high break angle pitch that is working very well and a pitcher wants to keep attempting similar pitches and pitch locations under the same at bat. Thus, the data was filtered to only include 1 pitch per at bat that has a high break angle. This should help to result in little to no interactions in the locations of pitches in the strike zone. This means that for each high break angle pitch location in the strike zone in the filtered data it should have no bearing to a previous pitch's location on the strike zone.

Theoretically, assuming a pitcher that has a high break angle pitch, it should not affect the location of strikes in the strike zone which should result in a homogeneous poisson process, and under the null hypothesis, a spatial variation (first-order) in intensity is that it should be expected that when a pitcher throws a higher break angle pitch it should result in a pitch location in the strike zone that is nearly constant in all areas of the strike zone (homogeneous poisson process). A secondary hypothesis is that when a pitcher knows that they will be throwing a higher break angle pitch, then it is more difficult to command the location of the pitch in the upper strike zone due to a lesser ability to know how high to throw if aiming for the upper strike zone.

Spatial statistical analysis on spatial dataset:

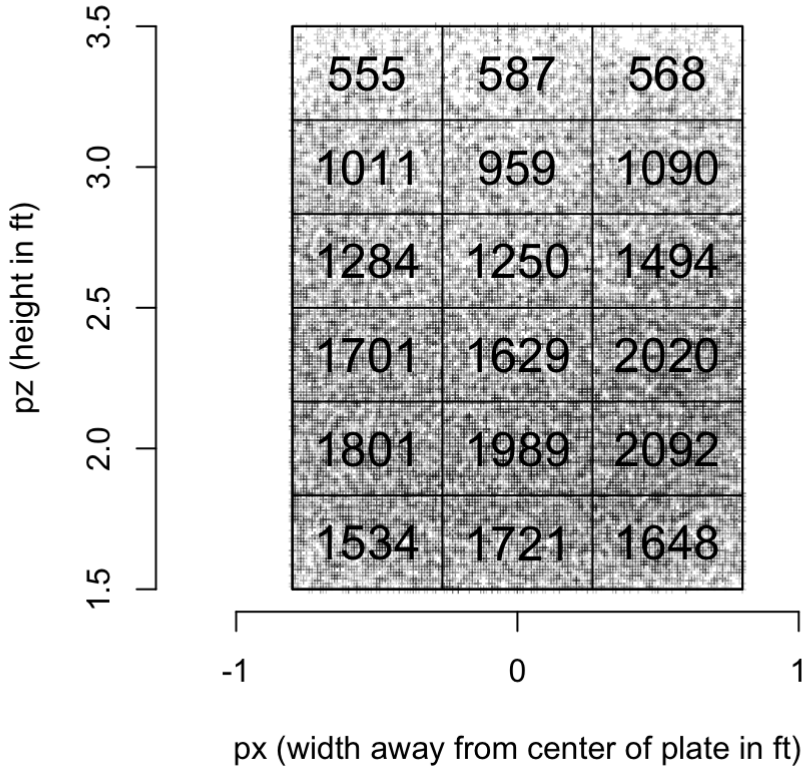
A formal chi-squared test is completed for Complete Spatial Randomness (CSR) model to determine if the high break angle pitching location data in the strike zone can be rejected. The pitch data is split up into 3 x 6 quadrants in the below plot with the ppp data generated from parts a and b. Also, a table of the observed pitches in each quadrant is given below.

```
# exploratory analysis -- quadrant count
Q <- quadratcount(pp1,
                  nx = 3,
                  ny = 6)
Q
```

```
##              x
## y             [-0.8,-0.267) [-0.267,0.267) [0.267,0.8]
##   [3.17,3.5]            555             587         568
##   [2.83,3.17)          1011             959        1090
##   [2.5,2.83)           1284            1250        1494
##   [2.17,2.5)           1701            1629        2020
##   [1.83,2.17)          1801            1989        2092
##   [1.5,1.83)           1534            1721        1648
```

```
# Plot the observed number of high break angle strike zone pitch locations in the strike
zone with 3 x 6 quadrants.
plot(pp1,
     cex = 0.5,
     pch = "+",
     main=str_wrap("Figure 2. High break angle strike zone pitch locations split into 3x
6 quadrants for a CSR chi^2 analysis.", 75))
plot(Q,
     add = TRUE,
     cex = 1.5)
title(xlab="px (width away from center of plate in ft)")
title(ylab="pz (height in ft)", line = -2)
axis(1, at = c(-1, 0, 1))
axis(2, line = -5)
```

Figure 2. High break angle strike zone pitch locations split into 3x6 quadrants for a CSR chi^2 analysis.



Using the total number of observed counts of pitches, the expected number of pitches per quadrant is calculated based on the total number of pitches divided by the number of quadrants (1385.167). Then for each quadrant, the squared difference between number of observed pitches by the number of expected pitches is divided by the number of expected pitches. The statistic is summed for all quadrants and is reported as the χ^2 statistic. The following code chunk below shows the calculation of the χ^2 statistic using the quadrat.test() function and the iterative method calculation as described. Both methods result in a χ^2 statistic of 3065.7. With 17 degrees of freedom (18 quadrants - 1) and χ^2 of 3065.7, the p-value is presumed to be extremely small and basically 0. With a tiny p-value (the p-value is significantly smaller than $\alpha = 0.05$ or 0.01), this means that the high break angle pitching location strike zone data can emphatically reject the null hypothesis of a CSR pattern.

```
# Using the quadrat.test function, calculate the chi^2 test of whether the spatial data
 is generated from a null CSR process or not.
Qtest <- quadrat.test(pp1, nx = 3, ny = 6, method = "Chisq")
Qtest
```

```
##
##  Chi-squared test of CSR using quadrat counts
##
## data:  pp1
## X2 = 3065.7, df = 17, p-value < 2.2e-16
## alternative hypothesis: two.sided
##
## Quadrats: 3 by 6 grid of tiles
```

```
Qtest$expected # Expected number of pitches per quadrant.
```

```
##  [1] 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167
##  [9] 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167 1385.167
## [17] 1385.167 1385.167
```

```
Qtest$observed # Observed number of pitches per quadrant.
```

```
##  [1]  555  587  568 1011  959 1090 1284 1250 1494 1701 1629 2020 1801 1989 2092
## [16] 1534 1721 1648
```

```
# The following is the calculation of the chi^2 statistic, the p-value is obtained from
 a table with 17 degrees of freedom, which identifies that the p-value is basically 0 (e
xtremely small p-value) and that the null hypothesis is emphatically rejected for this s
patial data being generated from a CSR process.
total = 0
for (i in 1:18){
total = ((as.numeric(Qtest$observed[i]) - as.numeric(Qtest$expected[i]))^2)/as.numeric(Q
test$expected[i]) + total
}
total
```

```
## [1] 3065.682
```

```
pchisq(total, df=17, lower.tail = FALSE)
```

```
## [1] 0
```

Further analysis was completed on the high break angle pitching data to determine which model is potentially the best first-order model of the pitching location. It is shown below that with an ANOVA table between three models (constant trend, linear trend, and quadratic trend) shows that either the linear trend or quadratic trend models can

both be used to reject the null hypothesis of the CSR model both with p-values close to 0. Both the quadratic and linear models show that there is an inhomogeneous first-order spatial point pattern where the intensity of high break angle pitches are closer to the bottom of the strike zone than the top of the strike zone.

```
# Further analysis of the inhomogeneous mean (first-order) strike pitches data

# constant trend
fit1 <- ppm(pp1, ~ 1)
summary(fit1)
```

```
## Point process model
## Fitting method: maximum likelihood
## Model was fitted analytically
## Call:
## ppm.ppp(Q = pp1, trend = ~1)
## Edge correction: "border"
##  [border correction distance r = 0 ]
## --------------------------------------------------------------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
## Data pattern:
## Planar point pattern:  24933 points
## Average intensity 7790 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
##
## Dummy quadrature points:
##      320 x 320 grid of dummy points, plus 4 corner points
##      dummy spacing: 0.00500 x 0.00625 units
##
## Original dummy parameters: =
## Planar point pattern:  102404 points
## Average intensity 32000 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
## Quadrature weights:
##      (counting weights based on 320 x 320 array of rectangular tiles)
## All weights:
##  range: [3.91e-06, 3.12e-05] total: 3.2
## Weights on data points:
##  range: [3.91e-06, 1.56e-05] total: 0.299
## Weights on dummy points:
##  range: [3.91e-06, 3.12e-05] total: 2.9
## --------------------------------------------------------------------------------
## FITTED MODEL:
##
## Stationary Poisson process
##
## ---- Intensity: ----
##
##
## Uniform intensity:
## [1] 7791.563
##
##             Estimate        S.E.  CI95.lo  CI95.hi Ztest     Zval
## log(lambda) 8.960797 0.006333047 8.948384 8.973209   *** 1414.927
##
## ----------- gory details -----
##
## Fitted regular parameters (theta):
```

```
## log(lambda)
##    8.960797
##
## Fitted exp(theta):
## log(lambda)
##    7791.563
```

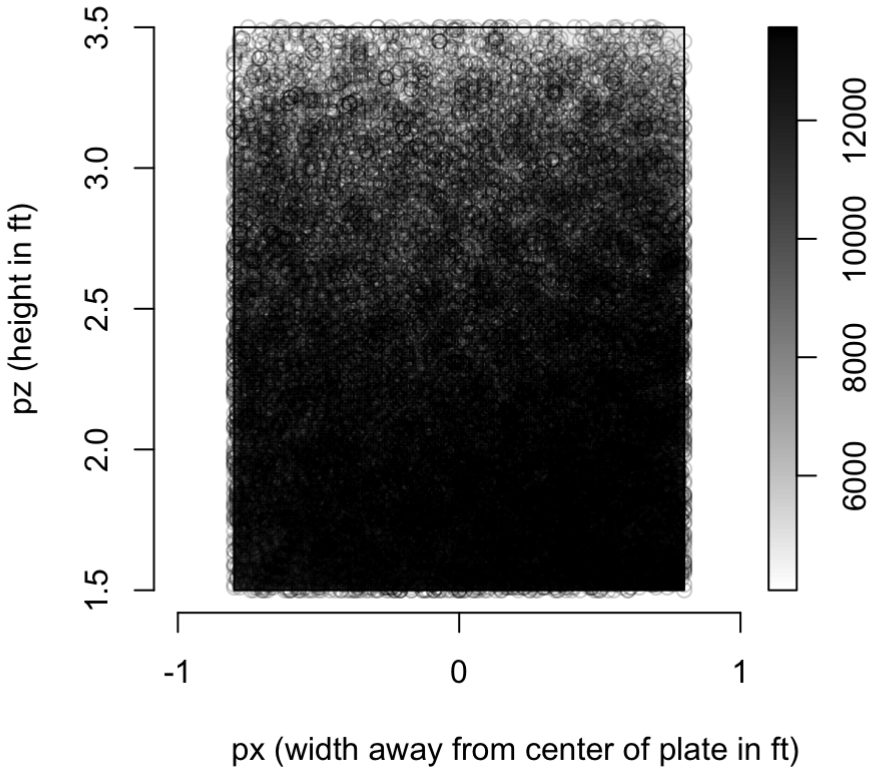
```
# linear trend in x and y
fit2 <- ppm(pp1, ~ x + y)
summary(fit2)
```

```
## Point process model
## Fitting method: maximum likelihood (Berman-Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
## ppm.ppp(Q = pp1, trend = ~x + y)
## Edge correction: "border"
##  [border correction distance r = 0 ]
## --------------------------------------------------------------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
## Data pattern:
## Planar point pattern:  24933 points
## Average intensity 7790 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
##
## Dummy quadrature points:
##      320 x 320 grid of dummy points, plus 4 corner points
##      dummy spacing: 0.00500 x 0.00625 units
##
## Original dummy parameters: =
## Planar point pattern:  102404 points
## Average intensity 32000 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
## Quadrature weights:
##      (counting weights based on 320 x 320 array of rectangular tiles)
## All weights:
##  range: [3.91e-06, 3.12e-05] total: 3.2
## Weights on data points:
##  range: [3.91e-06, 1.56e-05] total: 0.299
## Weights on dummy points:
##  range: [3.91e-06, 3.12e-05] total: 2.9
## --------------------------------------------------------------------------------
## FITTED MODEL:
##
## Nonstationary Poisson process
##
## ---- Intensity: ----
##
## Log intensity: ~x + y
##
## Fitted trend coefficients:
## (Intercept)           x           y
##  10.2385371   0.1099098  -0.5301780
##
##               Estimate       S.E.     CI95.lo    CI95.hi Ztest       Zval
## (Intercept) 10.2385371 0.02699053 10.18563660 10.2914375   *** 379.338194
## x            0.1099098 0.01371871  0.08302168  0.1367980   ***   8.011678
```

```
## y           -0.5301780 0.01127651 -0.55227958 -0.5080765   *** -47.016162
##
## ----------- gory details -----
##
## Fitted regular parameters (theta):
## (Intercept)           x           y
##  10.2385371   0.1099098  -0.5301780
##
## Fitted exp(theta):
##  (Intercept)            x            y
## 2.796019e+04 1.116177e+00 5.885002e-01
```

```
plot(fit2,
     how = "image",
     se = FALSE,
     col = grey(seq(1,0,length=128)),
     main = str_wrap("Figure 3. Linear trend (inhomogeneous) fit model on high break ang
le pitching data", 60))
title(xlab="px (width away from center of plate in ft)")
title(ylab="pz (height in ft)", line = -2)
axis(1, at = c(-1, 0, 1))
axis(2, line = -5)
```

Figure 3. Linear trend (inhomogeneous) fit model on high break angle pitching data



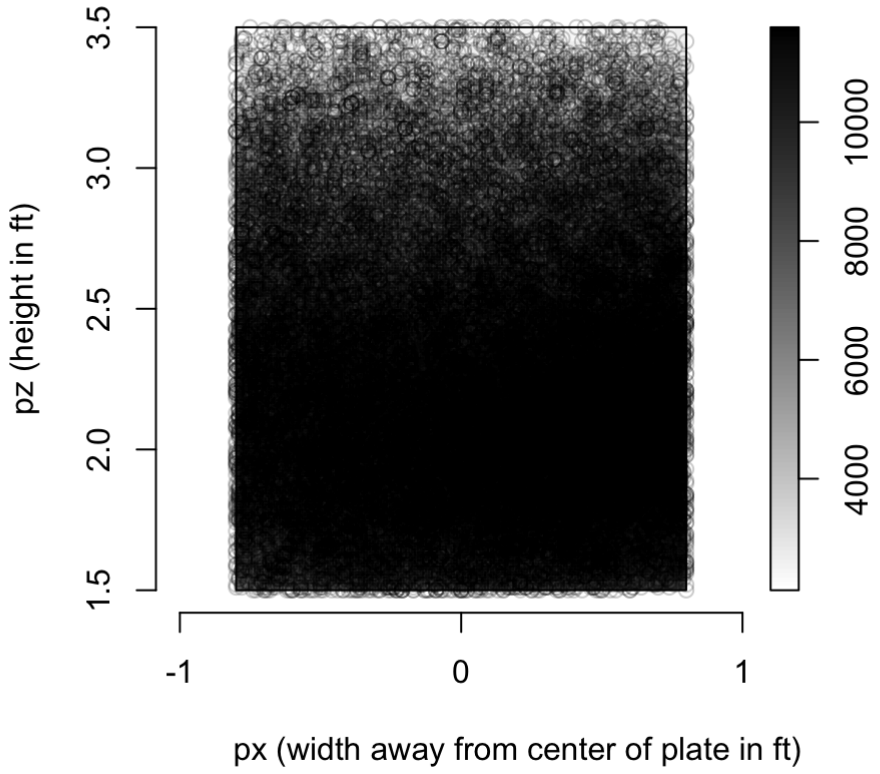
```
# quadratic trend in x and y
fit3 <- ppm(pp1, ~ polynom(x, y, 2))
summary(fit3)
```

```
## Point process model
## Fitting method: maximum likelihood (Berman-Turner approximation)
## Model was fitted using glm()
## Algorithm converged
## Call:
## ppm.ppp(Q = pp1, trend = ~polynom(x, y, 2))
## Edge correction: "border"
##  [border correction distance r = 0 ]
## --------------------------------------------------------------------------------
## Quadrature scheme (Berman-Turner) = data + dummy + weights
##
## Data pattern:
## Planar point pattern:  24933 points
## Average intensity 7790 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
##
## Dummy quadrature points:
##      320 x 320 grid of dummy points, plus 4 corner points
##      dummy spacing: 0.00500 x 0.00625 units
##
## Original dummy parameters: =
## Planar point pattern:  102404 points
## Average intensity 32000 points per square unit
## Window: rectangle = [-0.8, 0.8] x [1.5, 3.5] units
##                     (1.6 x 2 units)
## Window area = 3.2 square units
## Quadrature weights:
##      (counting weights based on 320 x 320 array of rectangular tiles)
## All weights:
##  range: [3.91e-06, 3.12e-05] total: 3.2
## Weights on data points:
##  range: [3.91e-06, 1.56e-05] total: 0.299
## Weights on dummy points:
##  range: [3.91e-06, 3.12e-05] total: 2.9
## --------------------------------------------------------------------------------
## FITTED MODEL:
##
## Nonstationary Poisson process
##
## ---- Intensity: ----
##
## Log intensity: ~x + y + I(x^2) + I(x * y) + I(y^2)
##
## Fitted trend coefficients:
## (Intercept)           x           y      I(x^2)    I(x * y)      I(y^2)
##  5.84186280  0.12446260  3.29745269 -0.01164587 -0.00616989 -0.79099414
##
##                Estimate       S.E.      CI95.lo     CI95.hi Ztest        Zval
## (Intercept)  5.84186280 0.13653644  5.574256298  6.10946930   ***  42.7861079
## x            0.12446260 0.06484303 -0.002627403  0.25155260         1.9194445
```

```
## y            3.29745269 0.11629590  3.069516904  3.52538847   ***  28.3539884
## I(x^2)      -0.01164587 0.03321632 -0.076748654  0.05345692        -0.3506067
## I(x * y)    -0.00616989 0.02723729 -0.059553996  0.04721422        -0.2265236
## I(y^2)      -0.79099414 0.02399044 -0.838014541 -0.74397374   *** -32.9712197
##
## ----------- gory details -----
##
## Fitted regular parameters (theta):
## (Intercept)           x           y      I(x^2)    I(x * y)      I(y^2)
##  5.84186280  0.12446260  3.29745269 -0.01164587 -0.00616989 -0.79099414
##
## Fitted exp(theta):
## (Intercept)           x           y      I(x^2)    I(x * y)      I(y^2)
## 344.4203299   1.1325397  27.0436624   0.9884217   0.9938491   0.4533938
```

```
plot(fit3,
     how = "image",
     se = FALSE,
     col = grey(seq(1,0,length=128)),
     main = str_wrap("Figure 4. Quadratic trend (inhomogeneous) fit model on high break
 angle pitching data", 60))
title(xlab="px (width away from center of plate in ft)")
title(ylab="pz (height in ft)", line = -2)
axis(1, at = c(-1, 0, 1))
axis(2, line = -5)
```

Figure 4. Quadratic trend (inhomogeneous) fit model on high break angle pitching data



```r
# Likelihood ratio test of
#     H0:  homogeneous PP (CSR) vs.
#     H1:  inhomogeneous PP with intensity that is a loglinear function of the xy
#     H2:  inhomogeneous PP with intensity that is a logquadratic function of the xy
anova.ppm(fit1, fit2, fit3, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: ~1   Poisson
## Model 2: ~x + y   Poisson
## Model 3: ~x + y + I(x^2) + I(x * y) + I(y^2)      Poisson
##   Npar Df Deviance  Pr(>Chi)
## 1    1
## 2    3  2   2336.7 < 2.2e-16 ***
## 3    6  3   1167.8 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```