THE POWER TO KNOW.

SAS/STAT® 15.1 User's Guide

The SPP Procedure

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 15.1 User's Guide

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

Chapter 111
The SPP Procedure

Contents

Overview: SPP Procedure

The SPP procedure performs analysis for spatial point patterns in two dimensions. You can specify the point process rectangular window or rely on the input data set coordinates. Summary descriptions are available through the F, G, J, K functions, which compare the empirical function distributions to the theoretical homogeneous Poisson functions.

The SPP procedure uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For more information about the graphics available in PROC SPP, see the section "ODS Graphics" on page 9312.

Classes of Spatial Data

There are three broad classes of spatial data:

- *Point-referenced data* are values that are sampled at specific locations within an area of a predefined size. An example is air temperatures that are measured where weather monitoring instruments are located. The stochastic nature of spatial processes can be described by using spatial random fields (SRFs). A set of point-referenced data can be seen as a realization of a continuous SRF that takes values over the entire study area. The values at unsampled locations are unknown but can be predicted by means of geostatistical analysis. You can analyze point-referenced data by using the SAS/STAT procedures VARIOGRAM, KRIGE2D, and SIM2D.

- *Areal (lattice) data* are values for a fixed number of areal units within a particular area. These data differ the point-referenced data in that one areal observation is assigned to a whole areal unit instead of to a specific location. An example is crime rates that are aggregated over counties within a state.

- *Point pattern data* are a collection of locations of single events of a spatial process. In this category, the study area can have a variable size and observations might have associated covariates, but the main interest is in their spatial patterns of occurrence. Examples include locations of tree growth, locations of petty crimes, and so on. A set of point-pattern data can be seen as a realization of a discrete SRF that has values only at the event locations (Illian et al. 2008, p. 44). A collection of this type of data is known as a spatial *point pattern*. Point pattern analysis usually does not refer to the SRF concept. The applied techniques in point patterns differ from the geostatistical approach, although both types of analysis share corresponding measures to describe correlation among the data. You can use the SPP procedure to analyze point pattern data.

Introduction to Point Pattern Analysis

In point pattern analysis, you want to describe characteristics of the *events* (observations) that compose the pattern. The events are manifestations of a phenomenon or process at random locations. Therefore, your analysis goal is to investigate underlying connections among these events that could explain the phenomenon.

In some cases, events might have additional attributes, known as *marks*. If a point pattern has a mark, then it is called a *marked point pattern*. There can be *continuous marks* or *categorical marks*, depending on whether the mark attribute takes continuous values or values from a list of discrete levels, respectively. A marked point pattern that has a categorical mark attribute is known as a *multitype point pattern*. A multitype point pattern is also called a *multivariate point pattern* because you can view it as a collection of point patterns, one for each type.

To study the events, you use the concepts of the *study region* (also called a *study window*) to represent the area where the point pattern is defined. The window selection can be a subjective choice, and it definitely can affect the analysis. When the window is a subregion of a larger region where the point process operates, you might need to account for *edge effects*. This term describes discrepancies that can appear in the analysis, depending on whether you consider that events close to the window edges have neighbors outside the window area.

Point pattern analysis often focuses on whether interaction exists among the observations in a spatial point pattern. That is, you test whether the points are spread evenly around the study region with no particular pattern, or alternatively whether there tends to be more or less clumping of points than you would expect purely from randomness. To this end, you usually test the hypothesis of complete spatial randomness (CSR) in the point pattern. According to CSR, the events follow a Poisson distribution with constant mean, and they have no interactions. A point pattern can follow CSR, in which case it is known as a *homogeneous Poisson process*. Alternatively, a point pattern can demonstrate event interaction or clustering.

You can test CSR by using heuristic approaches that use *sparse sampling methods* in exploratory and summary analysis. Two general approaches to this are as follows:

- *distance methods*, where you compare the empirical distribution function (EDF) of distance between events with an EDF that is based on the CSR assumption
- *quadrats*, where you partition the spatial framework into smaller subregions and study the number of events (also known as the *quadrat count*) in each subregion

The SPP procedure provides options for implementing both of these approaches. For more information, see the sections "Testing for Complete Spatial Randomness" on page 9293 and "Statistics Based on Second-Order Characteristics" on page 9295.

You can tell a lot about the behavior of a point pattern if you have an expression for the point pattern *intensity*, which shows the number of events per unit area. A simple way to estimate intensity from the point pattern events is to produce kernel density estimates. You can also model the intensity by maximizing suitable pseudolikelihood expressions for the logarithmic intensity. Intensity models can also incorporate information about covariate variables; together with distance methods, they enable you to examine whether a covariate plays a significant role in the underlying process.

A SAS/STAT procedure that compares to PROC SPP is the KDE procedure, which fits the special case of Gaussian bivariate kernels for the purpose of nonparametric density estimation. PROC SPP enables you to

perform much more extensive nonparametric intensity estimation by using different types of kernels, and it provides support for adaptive kernel estimation. In addition, PROC SPP enables you to fit parametric inhomogeneous poisson process models and use a variety of residual diagnostics to perform model validation.

Getting Started: SPP Procedure

This example uses forestry data, which are shown in Figure 111.4, to show how you can use PROC SPP to fit a model for the first-order intensity of a spatial point pattern. The Sashelp.BEI data set contains the locations of 3,604 trees in tropical rain forests. A study window of 1,000 × 500 square kilometers is appropriate. The data set also contains covariates that are represented by the variables Gradient and Elevation, which are collected at 20,301 locations on a regular grid across the study region. The variable Trees distinguishes the event observations in the data set. These data are a part of a much larger data set, which contains the positions of hundreds of thousands of trees that belong to thousands of species (Condit 1998; Hubbell and Foster 1983; Condit, Hubbell, and Foster 1996).[1] The Sashelp.BEI data set contains five variables:

- X and Y: the X and Y coordinates for locations of trees and for measurements of the height and slope of the study area
- Trees: a 0/1 variable that indicates which observation corresponds to locations of trees: 1 indicates the presence of a tree, and 0 indicates absence
- Elevation: which measures how far the study area is above sea level
- Gradient: which measures the slope of the study area

The following statements produce a plot of the event observations (which is shown in Figure 111.4) and plots of the covariates (which are shown in Figure 111.5 and Figure 111.6).

```
ods graphics on;
proc spp data=sashelp.bei plots(equate)=(trends observations);
   process trees = (x, y /area=(0,0,1000,500) Event=Trees);
   trend grad = field(x,y, gradient);
   trend elev = field(x,y, elevation);
run;
```

In addition, the preceding statements produce three tables, which are shown in Figure 111.1, Figure 111.2, and Figure 111.3. The number of observations in the combined data set is shown in Figure 111.1; it includes both the number of event observations and the number of covariate observations.

[1] This data set is used with kind permission from Professor S. Hubbell, with acknowledgment of the support of the Center for Tropical Forest Science of the Smithsonian Tropical Research Institute and the primary granting agencies that have supported the BCI plot. The BCI forest dynamics research project was made possible by National Science Foundation grants to Stephen P. Hubbell: DEB-0640386, DEB-0425651, DEB-0346488, DEB-0129874, DEB-00753102, DEB-9909347, DEB-9615226, DEB-9615226, DEB-9405933, DEB-9221033, DEB-9100058, DEB-8906869, DEB-8605042, DEB-8206992, DEB-7922197, support from the Center for Tropical Forest Science, the Smithsonian Tropical Research Institute, the John D. and Catherine T. MacArthur Foundation, the Mellon Foundation, the Small World Institute Fund, and numerous private individuals, and through the hard work of over 100 people from 10 countries over the past two decades. The plot project is part of the Center for Tropical Forest Science, a global network of large-scale demographic tree plots.

Figure 111.1 Number of Events and Number of Covariate Observations

The SPP Procedure

Observations Read	24205
Observations Used	23905
Event Observations Read	3604
Event Observations Used	3604
Gradient Observations Read	20301
Gradient Observations Used	20301
Elevation Observations Read	20301
Elevation Observations Used	20301

Figure 111.2 provides some summary information about the point pattern, including the average intensity or the number of events per unit area.

Figure 111.2 Exploratory Information about the Point Pattern

Summary of Point Pattern	
Data Type	Point Pattern
Pattern Name	trees
Region Type	User Defined Window
Region X Range	[0,1000] Units
Region Y Range	[0,500] Units
Region X Size	1000 Units
Region Y Size	500 Units
Region Area	500000 Square Units
Observations in Window	3604
Average Intensity	0.007208
Grid Nodes in X	50
Grid Nodes in Y	50
Grid Nodes in Window	2500
Quadrat Dimension in X	10
Quadrat Dimension in Y	10

Figure 111.3 provides the results of a default 10×10 quadrat-based Pearson chi-square test for CSR.

Figure 111.3 Pearson Chi-Square Test for CSR

Pearson Chi-Square Test for CSR				
Expected Frequency	**DF**	**Dispersion Index**	**Chi-Square**	**Pr > ChiSq**
36.04	99	33.222	3288.95	<.0001

Figure 111.4 Spatial Point Pattern of Tropical Rain Forest Trees

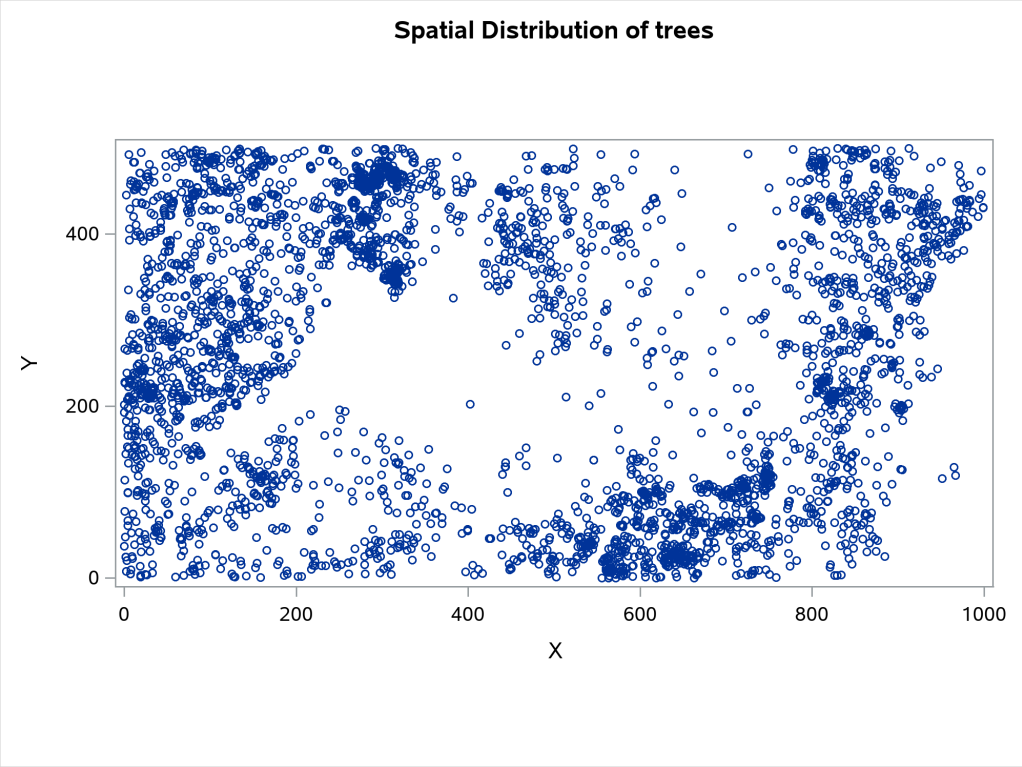


Figure 111.5 Spatial Covariate Gradient

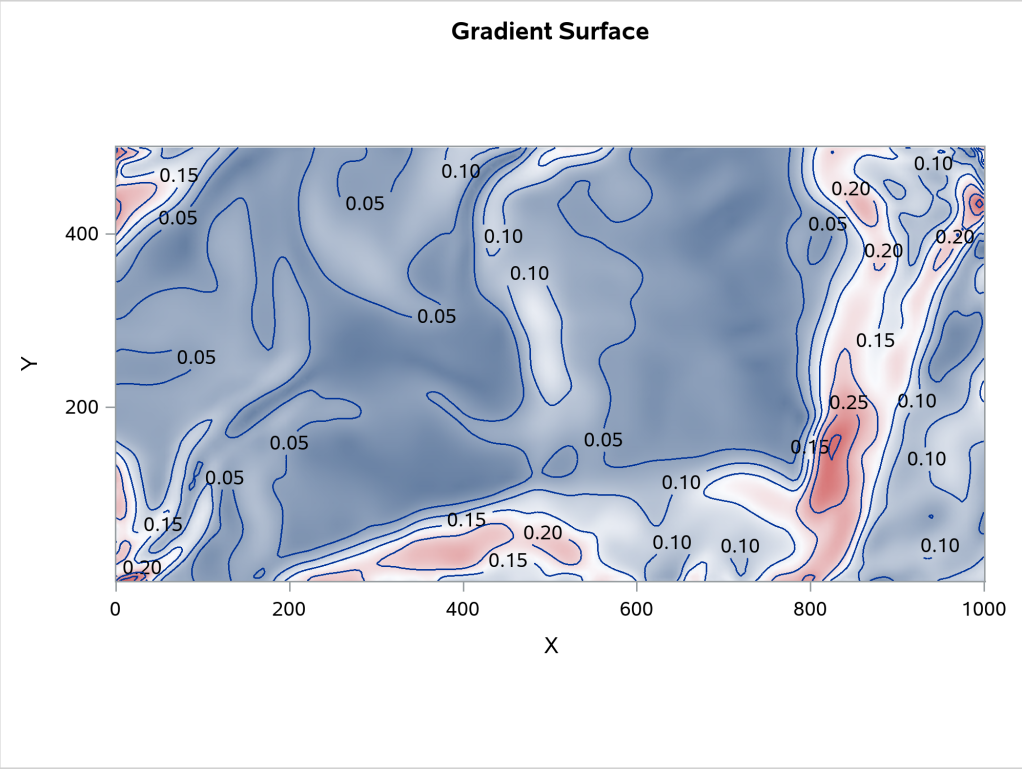
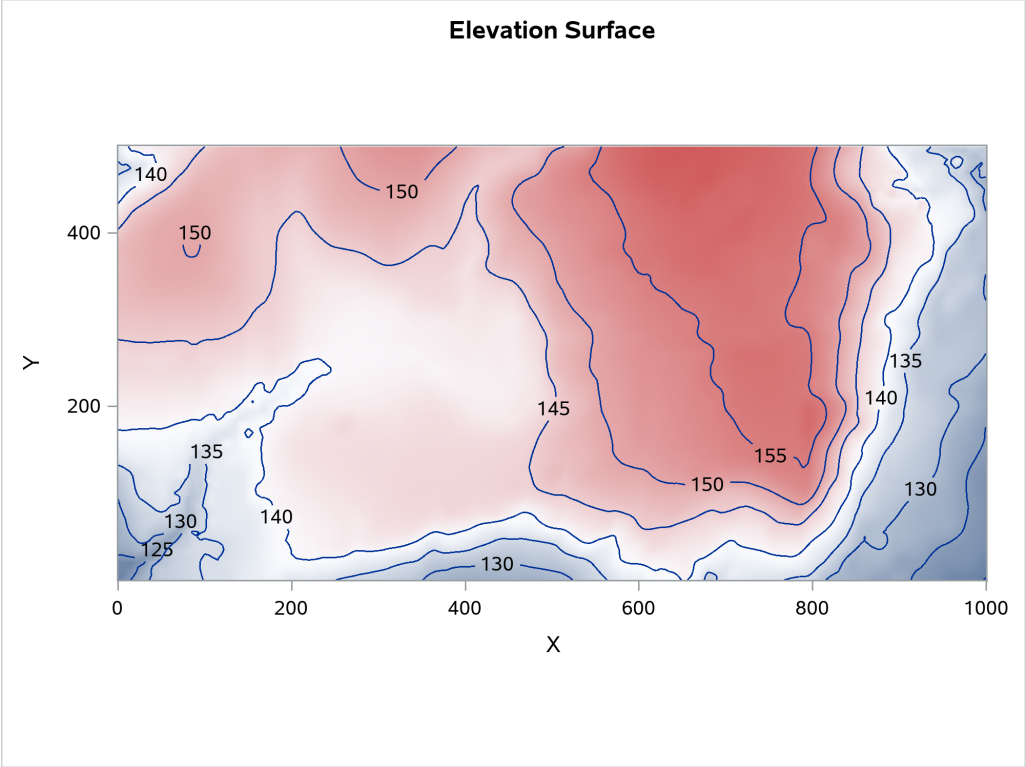


Figure 111.6 Spatial Covariate Elevation



The variables Gradient and Elevation are both continuous functions, because any arbitrary point that is chosen in the study area has a value for both these variables. However, these variables are sampled at select points where measuring them is easy. In spatial analysis and geographic information systems (GISs), such variables are termed *field* variables and are associated with a spatial trend. You can include such variables in the SPP procedure by using the TREND statement.

The sashelp.bei data contains combined information for both the point pattern and the spatial covariates. However, the SPP procedure requires you to identify the point pattern event identifier separately. This is done by using the EVENT= option in the PROCESS statement to specify that the variable Trees identifies the event.

It is natural to suppose that tree growth is affected by the gradient and elevation of the surrounding land. Hence, you can use the gradient and elevation in a parametric model to model the intensity of tree growth in the study area. Such a model is an inhomogeneous Poisson process (Baddeley 2010, p. 354), whose first-order intensity, $\lambda(s)$, is log linear in the covariates. You can use the MODEL statement to compose models for a point pattern's intensity. In the MODEL statement, you specify the response pattern on the left side. The response pattern is a process that you define before you specify the MODEL statement. You can specify any covariates that are likely to influence the target point pattern on the right side of the MODEL statement syntax.

To obtain a plot of the model-based intensity estimate, you specify the PLOTS=INTENSITY option. In addition, if you want to request residual diagnostics, you can specify the PLOTS=RESIDUAL option. If you want to specify a response grid to obtain the intensity estimates, you can use the GRID option in the MODEL statement. The following statements explore the influence of the covariates Elevation and Gradient on the intensity of Tree presence:

```
proc spp data=sashelp.bei plots(equate)=(residual intensity);
   process trees = (x,y /area=(0,0,1000,500) event=Trees);
   trend elev = field(x,y,elevation);
   trend grad = field(x,y,gradient);
   model trees = elev grad / grid(64,64) residual(B=70) ;
run;
```

In addition to the tables shown in previous figures, these statements produce a table that contains the parameter estimates (Figure 111.7) and a fit summary table (Figure 111.8). The parameter estimates designate the intercept value and the values of the factors of the model terms. The relative values of the parameter estimates indicate how much each factor contributes to the model. In this case, Gradient is much more important in modeling where trees grow than Elevation, although both are highly significant.

Figure 111.7 Parameter Estimates Table

The SPP Procedure

Poisson Parameter Estimates

Parameter	Estimate	Standard Error	z Value	Approx Pr > |z|
Intercept	-8.5672	0.3415	-25.08	<.0001
Elevation	0.02146	0.002291	9.37	<.0001
Gradient	5.8616	0.2567	22.83	<.0001

The fit summary table in Figure 111.8 shows the model fit statistics. You can use these values to compare multiple fits from different models and to select an optimal model in your study.

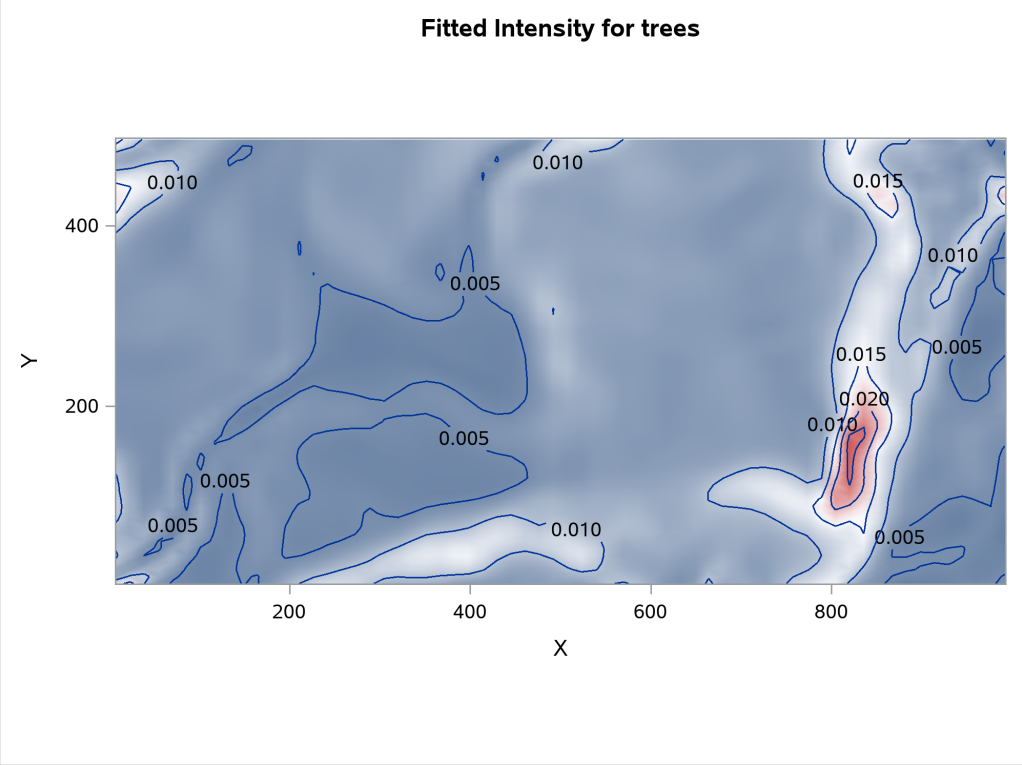
Figure 111.8 Fit Summary Table

Fit Statistics

Criterion	Value
-2 Log Likelihood	42290.0
AIC (smaller is better)	42296.0
BIC (smaller is better)	42316.8

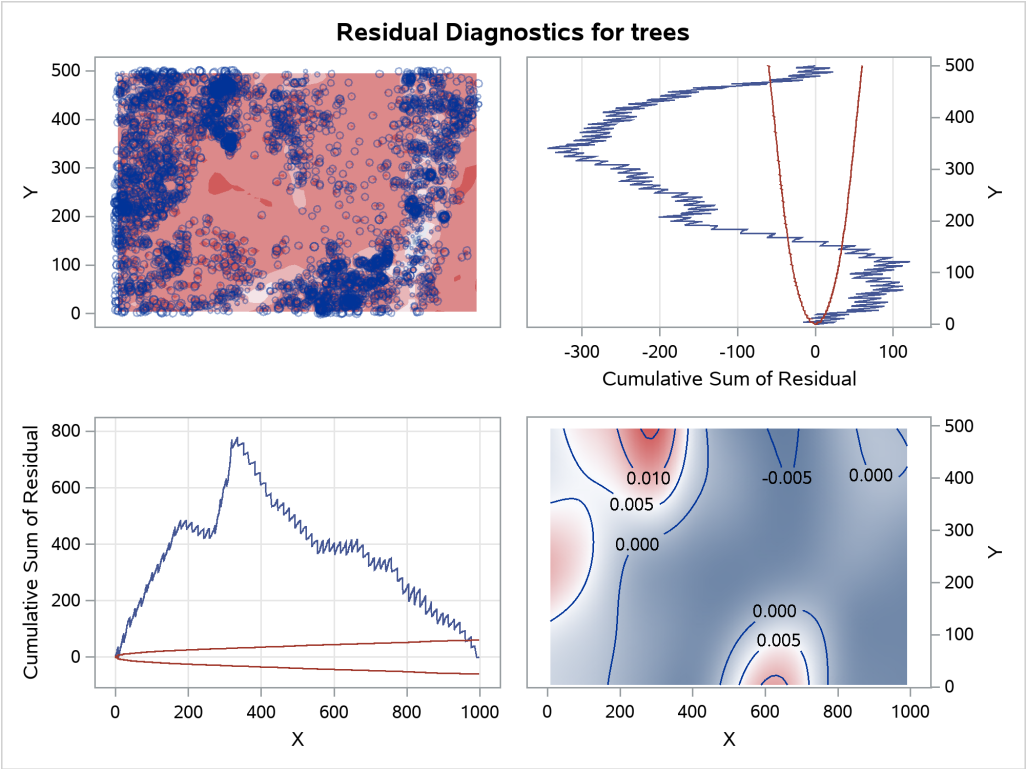
The corresponding fitted intensity is shown in Figure 111.9.

Figure 111.9 Intensity Estimates of Tree Presence in Study Area



The resulting residual diagnostics are shown in Figure 111.10.

Figure 111.10 Residual Diagnostics for Fitted Log-Intensity Model



The residual diagnostics plot in Figure 111.10 provides an informal assessment of the fitted parametric model. In particular, the smoothed residual plot in the right bottom corner reveals a trend in the residual that is not accounted for by the model. In addition, the lurking variable plots with respect to the coordinate variables show significant deviation from the 2σ limits, indicating that the model does not account for a variation in intensity with respect to these variables.

Syntax: SPP Procedure

The following statements are available in PROC SPP:

PROC SPP *options* **;**
BY *variables* **;**
PROCESS *name* **=** (*variables* < */pattern-options* >)< */process-options* < *distance-function-options* > >
;
TREND *name* **= FIELD(***field-definition* **) ;**
COVTEST *process-name* **=** *trend-name* < *trend-name, . . .* >< */options* > **;**
MODEL *process-name* **=** < *trend-name, . . .* >< */model-options* > **;**
PARMS *value-list* < / **PARMSDATA**=*SAS-data set* > **;**
NLOPTIONS < *options* > **;**

You must specify at least one PROCESS statement. The MODEL statement and the COVTEST statements must have one process variable on the left side and can have one or more processes or trends on the right side. When you specify the PARMS and NLOPTIONS statements, they must be preceded by the MODEL statement.

The following sections describe the PROC SPP statement and then describe the other statements in alphabetical order.

PROC SPP Statement

PROC SPP *options* **;**

The PROC SPP statement invokes the SPP procedure. Table 111.1 summarizes the options available in the PROC SPP statement.

Table 111.1 Options Available in the PROC SPP Statement

Option	Description
DATA=	Specifies the input data set
EDGECORR=	Requests edge correction in the analysis
NODUP	Specifies inclusion or exclusion of collocated observations
NOPRINT	Suppresses normal display of results
PLOTS	Specifies the plot display and options
SEED=	Specifies the seed value for the random number generator

You can specify the following *options* in the PROC SPP statement.

DATA=*SAS-data-set*
specifies a SAS data set that contains the x and y coordinate variables of one or more point patterns, associated mark variables, and event identifiers. Mark variables and event identifiers are specified using MARK= and EVENT= options, respectively, in the PROCESS statement. If your analysis involves covariates, you must also include them in the DATA= data set. When you include covariates, you must identify individual point patterns by specifying the EVENT= option in the PROCESS statement. You must specify a DATA=*SAS-data-set*; there is no default.

EDGECORR=ON | OFF
specifies whether you want to correct edge effects in the distance function computations and kernel density estimation. Edge correction is not applicable for the J function. For more information about how SPP implements edge correction, see the section "Border Edge Correction for Distance Functions" on page 9298. By default, EDGECORR=ON.

NODUP=*nodup-option*
specifies whether to eliminate multiple records of data that have the same pairs of coordinates in the DATA= data set. When multiple such records exist among observations of the event, or among observations of the same covariate variable, they are known as duplicates. For example, if two or more event records feature the same coordinates, then your data contain duplicates. However, if your data contain a record of an event and a record of a covariate that happen to be sampled at the same coordinates, then they are not duplicates.

The analysis of a spatial point pattern usually requires that no two events can share the same location. If your data include such duplicates, this option enables you to deal with them in different ways. You can specify the following values:

TRUE < (*true-suboption***) >**
removes duplicates from the analysis. You can also specify the following *true-suboption*:

KEEP=AVG | ONE
specifies how to treat removal of duplicate records. You can specify the following values:

AVG removes all but one record from a set of records that contain duplicate coordinates. In addition, if the duplicates are records of a numeric mark or covariate, then the average attribute value of all duplicate records is assigned to the single record that is retained. If any of the duplicate records has a missing value for the numeric mark or covariate, then it does not contribute to the average. Character variables ignore the KEEP=AVG suboption and retain only the last value in any series of collocated records.

ONE keeps only a single record out of multiple records that have the same duplicate coordinates. When you specify KEEP=ONE, PROC SPP retains the last record in any series of collocated records.

By default, KEEP=ONE.

FALSE
retains and uses all duplicates in the analysis.

If mark or covariate variables are included in the analysis, the NODUP= option specification applies the same mode of action to each individual variable. If PROC SPP finds duplicates, then it issues a note. By default, NODUP=TRUE(KEEP=ONE).

NOPRINT
suppresses the normal display of results. This option is useful when you want only to create one or more output data sets with the procedure.
NOTE: This option temporarily disables the Output Delivery System (ODS). For more information, see the section "ODS Graphics" on page 9312.

PLOTS < (*global-plot-options***) > < = *plot-request* < *(options)* > >**

PLOTS < (*global-plot-options***) > < = (***plot-request* < *(options)* > **< ...** *plot-request* < *(options)* > **>) >**
controls the plots that are produced through ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=observ
plots=(observ intensity)
plots(unpack)=observ
plots=(observ(attr=mark) observ(attr=event))
```

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 623 in Chapter 21, "Statistical Graphics Using ODS."

You can specify the following *global-plot-options*:

EQUATE
produces all plots that have coordinates so that the axes coordinates have equal size units. This option is ignored for panel plots.

ONLY
suppresses the default plots. Only plots that are specifically requested are displayed.

UNPACKPANEL
UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack the default plots. You can also unpack individual panel plots by specifying the UNP suboption in the FFUN, GFUN, KFUN, LFUN, and OBSERVATIONS(LEVEL=(SPLIT)) plot options.

You can specify the following individual *plot-requests* and *options*:

ALL
produces all appropriate plots. You can also specify other *options* with ALL.

CSRKSTEST
produces a plot for the Kolmogorov-Smirnov weighted EDF test for complete spatial randomness in the presence of covariates. To request this plot, you must specify the COVTEST statement and include trends on the right side of the COVTEST statement.

EMPTYSPACE < (*emptyspace-plot-options***) >**
produces a plot of the nearest-neighbor distance for every grid node in the window. You can specify the following *emptyspace-plot-options*:

FILL=ON | OFF
specifies whether to produce a surface plot of the nearest neighbor distances. By default, FILL=ON.

LINE=ON | OFF
specifies whether to produce a contour line plot of the nearest neighbor distances. By default, LINE=OFF.

OBS=ON | OFF
specifies whether to produce an overlaid scatter plot of the observations in addition to nearest neighbor distances. By default, OBS=OFF.

F < (UNPACK) >
requests that a panel of diagnostics for the empty space function F be produced. The F function is the empirical distribution of observed distances to the nearest observation from any location in the point pattern window. The panel contains four plots: an EDF plot that shows simulation envelopes for CSR, an EDF-CSR difference plot, a PP plot that compares the EDF of the summary statistic, and a confidence interval plot that shows envelopes for the confidence intervals of the summary statistic. If you specify the PLOTS=F option without requesting any distance function calculations in the PROCESS statement, then it is ignored. You can specify the following option:

UNPACK
suppresses paneling of the F function plots and produces each constituent plot in the panel separately.

The F plot is produced when you specify the F option in the PROCESS statement.

G < (UNPACK) >
produces a panel of diagnostics for the nearest-neighbor distance function G. The G function is the empirical distribution function of observed distances to the nearest observation from any other observation in the point pattern window. The panel contains four plots: an EDF plot that shows simulation envelopes for CSR, an EDF-CSR difference plot, a PP plot that compares the EDF of the summary statistic, and a confidence interval plot that shows envelopes for the confidence intervals of the summary statistic. If you specify PLOTS=G option without requesting any distance function calculations in the PROCESS statement, then it is ignored.

You can specify the following option:

UNPACK
suppresses paneling of the G function plots and produces each constituent plot in the panel separately.

The G plot is produced when you specify the G if you specify the G function option in the PROCESS statement.

INTENSITY < (*intensity-plot-options*) >
produces a plot of the estimated intensity function for every grid node in the window. You can specify the following *intensity-plot-options*:

EST=KERNEL | FIT
specifies the source to use for the intensity estimate. You can specify the following values:

KERNEL produces a plot of the intensity kernel density estimate. This suboption is incompatible with requests for standard error in the FILL= and LINE= intensity plot options. If you specify EST=KERNEL and either the FILL=SE suboption or the LINE=SE suboption, then intensity plot request is ignored.

FIT produces a plot of the estimated intensity on the basis of a model fit when you fit an intensity model by specifying the MODEL statement.

FILL=INTENSITY | NONE | SE
specifies which type of surface plot to produce. You can specify the following values:

INTENSITY	produces an estimated intensity surface plot.
NONE	produces no surface plot.
SE	produces a standard errors surface plot.

The default behavior depends on the LINE suboption as follows: If you specify LINE=NONE or entirely omit the LINE suboption, then FILL=INTENSTIY. If you specify LINE=INTENSITY or LINE=SE, then the FILL= suboption is set to the same value as the LINE suboption.

LINE=INTENSITY | NONE | SE
specifies which type of plot to produce. You can specify the following values:

INTENSITY	produces an estimated intensity contour line plot.
NONE	produces no contour line plot.
SE	produces a standard errors contour line plot.

If you omit the LINE suboption, the behavior depends on the FILL suboption as follows: If you specify FILL=NONE or entirely omit the FILL= suboption, then LINE=INTENSITY. If you specify FILL=INTENSITY or FILL=SE, then the LINE suboption is set to the same value as the FILL suboption.

OBS=ON | OFF
specifies whether to produce an overlaid scatter plot of the observations in addition to the intensity plot. By default, OBS=OFF.

You can specify multiple instances of the INTENSITY plot option to produce intensity plots that have different characteristics. If you specify multiple instances of any of the FILL=, LINE=, or OBS= suboptions in the same INTENSITY plot request, then one plot is produced that honors the last value specified for any of these suboptions. If you explicitly specify (or the suboptions imply) the combination FILL=NONE and LINE=NONE, then the intensity plot is not produced.

J < (UNPACK) >
produces a combined plot of the J function. The J function is the ratio of transformations of the F and G nearest-neighbor functions. The combined plot shows both the confidence intervals for the summary statistic and the simulation envelope for comparison with CSR. You can specify the following option:

UNPACK
produces each constituent J plot separately.

J plots are produced when you specify the J option in the PROCESS statement. If you specify PLOTS=J without specifying the J option in the PROCESS statement, then PLOTS=J is ignored.

K <(UNPACK)>
produces a panel of Ripley's K function. The K function is the ratio of the expected number of point pattern observations within distance *r* of any other observation divided by the average intensity value of the point pattern. The panel contains four plots: an EDF plot that shows simulation envelopes for CSR, an EDF-CSR difference plot, a PP plot that compares the EDF of the summary statistic, and a confidence interval plot that shows envelopes for the confidence intervals of the summary statistic.

The K plot is produced when you specify the K option in the PROCESS statement. If you specify PLOTS=K without specifying the K option in the PROCESS statement, then PLOTS=K is ignored. You can specify the following option:

UNPACK
suppresses paneling of the K function plots and produces each constituent plot separately.

L <(UNPACK)>
produces a panel of the L function, which is a transformation of the K function. The panel contains four plots: an EDF plot with simulation envelopes for CSR, an EDF-CSR difference plot, a PP plot that compares the EDF of the summary statistic, and a confidence interval plot that shows envelopes for the confidence intervals of the summary statistic.

The L plot is produced when you specify the L option in the PROCESS statement. If you specify the PLOTS=L option without requesting the L option in the PROCESS statement, then PLOTS=L is ignored. You can specify the following option:

UNPACK
suppresses paneling of the L function plots and produces each constituent plot separately.

LURKING <(*lurking-plot-options*)>
requests lurking variable plots, which show the cumulative raw residual with respect to the covariates or the coordinate variables or both. By default, PROC SPP computes lurking variable panel plots with respect to both covariates and coordinates. You can specify the following *lurking-plot-options*:

ALL
creates lurking variable plots of the model's covariates and of the coordinate variables that are specified in the PROCESS statement.

COORD
creates lurking variable plots only of the coordinate variables that are specified in the PROCESS statement.

COVAR
creates lurking variable plots only of the covariates and does not create plots with respect to the coordinate variables X and Y.

UNPACK
unpacks the lurking variable panel plots into individual lurking variable plots.

The default is LURKING(ALL).

NONE
suppresses all plots.

OBSERVATIONS < (*observations-plot-option*) >
OBSERV < (*observations-plot-option*) >
OBS < (*observations-plot-option*) >
produces the observed data plot. You can specify the following *observations-plot-options*:

ATTR=EVENT | MARK
specifies the observations attribute that you want to plot. You can specify the following values:

EVENT
specifies a plot of the locations of the point-pattern event observations.

MARK
specifies a plot of the locations and the mark values of the point-pattern event observations. If you do not specify OBS(MARK) or if the analysis skips the specified mark variable, then the observations plot request is ignored.

PCF < (UNPACK) >
produces a combined plot of the pair correlation function, g. The combined plot shows both the confidence intervals for the summary statistic and the simulation envelope for comparison with CSR.

The PCF plot is produced when you specify the PCF option in the PROCESS statement. If you specify the PLOTS=PCF option without specifying the PCF option in the PROCESS statement, then PLOTS=PCF is ignored. You can specify the following option:

UNPACK
suppresses the combination of different PCF plots into a single plot and produces each constituent plot separately.

RESIDUAL < (*residual-plot-options*) >
produces a plot of the residual diagnostics. By default, the SPP procedure produces a panel plot that contains smoothed raw residuals, raw residuals, and lurking variable plots with respect to the X and Y coordinates. In addition, you can specify the following *residual-plot-options*:

TYPE=CUM | RES
specifies the type of residual to be plotted in the lurking variable plots of the coordinate variables. You can specify the following values:

CUM	plots the cumulative residual
RES	plots a noncumulative residual as a scatter plot.

UNPACK
unpacks the panel plot, which contains smoothed raw residuals, raw residuals, and a lurking variable plot, into four separate plots.

SEED=*seed-value*
specifies the seed to use for the random number generator. The SEED= value has to be an integer.

TRENDS
produces a plot of all trend covariates. This option is ignored if no trend covariates are specified in the TREND statement.

BY Statement

BY *variables* **;**

You can specify a BY statement in PROC SPP to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement in the SPP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

COVTEST Statement

COVTEST *process-name* **=** *trend-name* < *trend-name, . . .* >< */options* > **;**

You use the COVTEST statement to perform covariate dependency tests that are based on an empirical distribution function (EDF). The COVTEST statement contains two essential parts: a *process-name* (which must be declared in a PROCESS statement that precedes the COVTEST statement) that you specify on the left side and a list of *trend-names* (which must be defined in a preceding TREND statement within the same PROC SPP call) that you specify on the right side. The procedure performs separate EDF tests for every *trend-name* that is specified in the right side of the COVTEST statement. When you include a trend on the right side of the COVTEST statement, PROC SPP performs a weighted EDF test. Performing EDF tests involves computing EDF statistics, for which the SPP procedure calculates the Kolmogorov-Smirnov D statistic by default. PROC SPP also produces an EDF plot for the Kolmogorov-Smirnov D statistic, which you can request by using the TEST=D option. If you are interested instead in the Cramér–von Mises W^2 statistic, you need to request it via the following *covtest-option*:

TEST=CM | D

requests the test statistics for any type of requested weighted tests. This option applies only if you have specified a trend on the right side of the COVTEST statement. The requested weighted test statistic is applied to every trend that is specified on the right side of the COVTEST statement. You can specify the following values:

D requests the Kolmogorov-Smirnov D statistic.

CM requests the Cramér–von Mises W^2 statistic.

By default, TEST=D.

MODEL Statement

MODEL *process-name* **=** < *trend-name, . . .* > < */model-options* > **;**

The MODEL statement enables you to fit an inhomogeneous Poisson process model. You must specify a *process-name* as the dependent variable. In addition, the MODEL statement enables you to specify multiple trends as covariates. If you do not specify any trends as covariates in the MODEL statement, PROC SPP fits a second-degree polynomial. The *process-name* must be defined in a preceding PROCESS statement, and each *trend-name* must be defined in a preceding TREND statement. Table 111.2 summarizes the *model-options* that you can specify.

Table 111.2 MODEL Statement Options

Option	Description
CENSCALE	Displays optimization centering and scaling information
CORRB	Requests the approximate correlation matrix
COVB	Requests the approximate covariance matrix
CL	Constructs a *t*-type confidence interval
GOF	Performs a chi-square-based goodness-of-fit test
GRID	Specifies the intensity response GRID size
ITHIST	Requests the optimization iteration history
MTYPE	Requests a specific type of model to be fit
OUTINTENSITY	Specifies an output data set to store the intensity estimates
OUTSIM	Specifies an output data set to store the simulations from an intensity model
POLYNOMIAL	Requests an additional polynomial component to be included in the model fitting process
RESIDUAL	Requests residual computations and specifies the bandwidth for smoothed residuals
SOLUTION	Requests display of raw results

You can specify the following *model-options*:

CENSCALE
lists the centering and scaling (standardization) information for each coordinate and covariate in the model.

CL< (*alpha-value*) >
requests a *t*-type confidence interval for the estimated parameters. You can also specify the significance level via the *alpha-value*. The default *alpha-value* is 0.05, which corresponds to the default confidence level of 95%.

CORRB
requests the estimated correlation matrix for the parameter estimates. To request the estimated correlation matrix for the model parameters with respect to the standardized covariates, specify both this *model-option* and the SOLUTION *model-option*.

COVB
requests the estimated covariance matrix for the parameter estimates. To request the estimated covariance matrix for the model parameters with respect to the standardized covariates, specify this *model-option* and the SOLUTION *model-option*.

GOF(*num-simulations*)
requests a goodness-of-fit test for the fitted intensity model. You can specify the number of Monte Carlo simulation runs as an integer in *num-simulations*. By default, the SPP procedure performs 100 simulations when you specify this option. It is recommended that you specify a QUADRAT option in the definition of the response/dependent point pattern in the PROCESS statement. If you do not specify such an option, the SPP procedure uses a default 10×10 quadrat.

GRID(*value-NX*,*value-NY*)
specifies the grid resolution for model fitting, where *value-NX* specifies the number of grids in the horizontal direction and *value-NY* specifies the number of grids in the vertical direction. By default, the SPP procedure fits the model on a 128×128 grid.

ITHIST< (PARM) >
requests an iteration history table for the model-fitting optimization. Specify this option to produce additional levels of output detail. You can specify the following value:

PARM
includes the fitting parameters in the iteration history table.

MTYPE=POISSON | NEGBINOMIAL
specifies the type of inhomogeneous intensity model to be fit by PROC SPP. You fit a negative binomial model only in order to diagnose overdispersion, so in this case no fitted intensity is produced, and likewise none of the goodness-of-fit tests or residual diagnostics that are based on the intensity are produced. You can specify the following values:

POISSON fits a Poisson process model.

NEGBINOMIAL fits a negative binomial model.

By default, MTYPE=POISSON.

OUTINTENSITY=*SAS-data-set*
specifies a *SAS-data-set* in which to store the output intensity estimate.

OUTSIM<(*iter-value***)>=***SAS-data-set*
specifies a *SAS-data-set* in which to store a simulated point pattern from a fitted intensity model. Specify the number of iterations in <*iter-value*> to generate multiple point pattern data sets. By default, the number of simulation iterations is set to 1.

POLYNOMIAL|POLY<(*degree* **)>**
specifies a polynomial trend in the coordinates. You can also specify the *degree* of the polynomial component. If you do not specify the degree, PROC SPP procedure uses a second-degree polynomial by default.

RESIDUAL(B=*value***)**
requests residual diagnostics for the inhomogeneous Poisson process model. If you specify this option, you must also specify the residual bandwidth for computing smoothed residuals via the B= suboption.

SOLUTION
displays the parameter estimates table in a location- and scale-standardized space. For optimization purposes, any polynomial coordinates and covariates in the model are centered and scaled. The parameters and the approximate covariance and the correlation matrices are displayed by default in the untransformed, unstandardized space. This option causes the output to be displayed on the basis of the actual fitted parameters in the transformed space. If you also specify the COVB or CORRB *model-option* (or both), then PROC SPP also displays the estimated covariance or correlation matrix, respectively (or both), in the transformed space.

You can specify additional options that are related to the nonlinear optimization aspects of the MODEL fitting process via the NLOPTIONS statement.

NLOPTIONS Statement

NLOPTIONS <*options*> **;**

The NLOPTIONS statement specifies details about the nonlinear optimization technique that PROC SPP uses to maximize the log-likelihood function for the first-order intensity model. By default, PROC SPP uses the Newton-Raphson with ridging optimization technique. For more information about the NLOPTIONS statement, see the section "NLOPTIONS Statement" on page 499 in Chapter 19, "Shared Concepts and Topics."

PARMS Statement

PARMS *value-list* <**/ PARMSDATA**=*SAS-data-set*> **;**

The PARMS statement specifies initial values for the parameters in the MODEL statement. Alternatively, the PARMS statement can request a grid search over several values of these parameters. The PARMS statement is optional and must follow the associated MODEL statement.

Table 111.3 PARMS Statement Options

Option	Description
Component Options	
PARMSDATA=	Specifies an input data set that contains initial values for the model parameters

Specification of parameter values in the PARMS statement is ordered, but the order is unrelated to the order in which you specify covariates in the MODEL statement. In particular, you must specify the initial parameter values by starting with the intercept parameter. Depending on the terms you specify in the model, you must continue sequentially by specifying the initial values for each of the monomials in a polynomial, and finally specify the coefficients that correspond to plain covariate terms in the model. If you have no initial value for one or more of the model parameters, then you can specify missing values as initial values. You can specify the *value-list* in any of following forms:

m	a single value
$m_1, m_2, \ldots, m_n$	several values
m to n	a sequence in which m equals the starting value, n equals the ending value, and the increment equals 1
m to n by i	a sequence in which m equals the starting value, n equals the ending value, and i equals the increment
m_1, m_2 to m_3	mixed values and sequences

For example, suppose you are fitting an intensity model that consists of a polynomial of first degree in each of the coordinates x and y and a term with the covariate variable Elevation. You want to specify an initial value of –3.5 for the intercept, and an initial value of –5 for the covariate Elevation. In the PARMS statement, you specify initial values for the Intercept parameter and the parameter of the Elevation variable, and no initial values for the parameters of the polynomial terms x, xy, and y. The following SAS statements implement these specifications:

```
proc spp data=sashelp.bei plots(equate) = intensity;
   process trees = (x,y /area=(0,0,1000,500) event=Trees);
   trend grad = field(x,y,gradient);
   trend elev = field(x,y,elevation);
   model trees = elev / grid(50,25) poly(1);
   parms (-3.5) (.) (-5);
run;
```

If you specify more than one set of initial values, a grid of initial values sets is created. PROC SPP searches among the specified sets for the set that yields the lowest objective function value. Then, the procedure uses the initial values in the selected set for the optimization.

The results from the PARMS statement are the values of the parameters on the specified grid.

You can specify the following option after a slash (/) in the PARMS statement:

PARMSDATA=*SAS-data-set*

PDATA=*SAS-data-set*

specifies the SAS data set from which to read model parameter values. The data set should contain the values in the sequence that is required by the PARMS statement in either of the following two ways:

- Specify one single column under the variable Estimate (Est) that contains all the parameter values.
- Use one column for each parameter, and place the *n* columns under the Parm1–Parm*n* variables.

For example, the following two data sets are equivalent ways to specify initial values for a model that requires four parameters:

```
data parData1;
   input Estimate @@;
   datalines;
0.5 -2 0.03 -3.4
;


 data parData2;
   input Parm1 Parm2 Parm3 Parm4 Parm5 Parm6 Parm7;
   datalines;
0.5 -2 0.3 . . 1 -5
0.5 -2 0.1 . . 0.1 -5
;
```

You can specify more than one set of initial values in the *SAS-data-set*. PROC SPP seeks among the specified sets for the one that gives the lowest objective function value. Then, the procedure uses the initial values in the selected set for the fitting optimization.

You can either explicitly specify initial parameter values in the PARMS statement or use the PDATA= option, but you cannot use both at the same time.

PROCESS Statement

PROCESS *name* **=** **(***variables* < */pattern-options* > **)**< */process-options* < *distance-function-options* > > **;**

The PROCESS statement defines a point pattern for analysis. You must use a valid SAS variable *name* to define the process, and you can describe it by using *variables* that contain the x and y coordinates of the points within the point pattern. The *variables* must also be in the DATA= data set. You can specify only one PROCESS statement in PROC SPP.

The coordinates in spatial data can be spherical (represented as longitude and latitude) or projected (represented as Cartesian x and y coordinates). All the SAS/STAT procedures that analyze spatial data, including PROC SPP, assume that you are working with projected coordinates, for which Euclidean distance is appropriate. If your data consist of spherical coordinates, you are responsible for transforming the data to projected coordinates, such as by using PROC GPROJECT in SAS/GRAPH software. For more information about the spatial modeling issues that pertain to the use of geodetic versus simple Euclidean distance, see Banerjee (2005).

You can also specify *pattern-options* and *process-options*. The *pattern-options* are related to different attributes of the observed point pattern that is read from the DATA= data set. The *process-options* represent different analyses that are associated with a point pattern. These analyses are usually helpful in characterizing the underlying stochastic process that might have generated the point pattern. The PROCESS statement's *pattern-options* are listed in Table 111.4. The PROCESS statement's *process-options* are listed in Table 111.5.

Table 111.4 Point Pattern Definition Options

Option	Description
AREA=	Specifies a rectangular study window
EVENT=	Specifies an EVENT variable that identifies individual point pattern events
MARK=	Specifies the MARK variable for the point pattern

You can specify the following *pattern-options*, which enable you to describe various aspects of a point pattern data set:

AREA=(*xmin-number, ymin-number, xmax-number, ymax-number***)**
specifies parameters that define the study area bounds for the spatial point pattern. This option describes is a key attribute that governs the intensity estimates that are obtained by different methods in PROC SPP. When you specify this option, you must identify all the following area specifications:

- *xmin-number*, the lower left limit for the *x* coordinate
- *ymin-number*, the lower left limit for the *y* coordinate
- *xmax-number*, the upper right limit for the *x* coordinate
- *ymax-number*, the upper right limit for the *y* coordinate

If there are BY groups in the DATA= data set, then the explicit bounds remain the same across all BY groups. If you do not specify this option, then PROC SPP estimates a default area based on the Ripley-Rasson window estimator. For more information about the Ripley-Rasson window estimate, see the section "Ripley-Rasson Window Estimator" on page 9300.

EVENT=*variable-name*
specifies an event variable that is associated with instances (points) in this point pattern. If your DATA= data set also contains information about covariates, use this option to identify the events in the point pattern.

MARK=*variable-name*
specifies a character or quantitative variable from the DATA= data set as a mark variable. Character variable marks are used for requesting distance function summary statistics across different variable values.

Table 111.5 PROCESS Statement Options

Option	Description
F	Computes the empty-space F function
G	Computes the G function

Table 111.5 *continued*

Option	Description
J	Computes the J function
K	Computes the K function to test for complete spatial randomness (CSR)
KERNEL	Obtains a nonparametric intensity estimate of the point pattern
L	Computes the L function
OUTSIM	Specifies an output data set to store the simulated data sets in computation of distance functions
PCF	Computes the PCF function
QUADRAT	Performs a quadrat-based test for CSR

You can specify the following *process-options* to study the point pattern data set and the underlying spatial point process that is likely to have generated this pattern:

F< GRID(*value-NX***,***value-NY***) >**
performs a test for complete spatial randomness that is based on the empty-space F function. For more information about the F function and related functions see the section "Statistics Based on Second-Order Characteristics" on page 9295. You can specify the following suboption:

GRID(*value-NX***,** *value-NY***)**
specifies a reference grid for computing the empty-space F function, where *value-NX* represents the number of horizontal divisions and *value-NY* represents the number of vertical divisions. By default, the SPP procedure uses a 50×50 grid.

G
performs a test for complete spatial randomness that is based on the nearest-neighbor G function.

J< GRID(*value-NX***,** *value-NY***) >**
performs a test for complete spatial randomness that is based on the J function. You can specify the following suboption:

GRID(*value-NX***,** *value-NY***)**
specifies a reference grid for computing the J function, where *value-NX* represents the number of horizontal divisions and *value-NY* represents the number of vertical divisions. By default, the SPP procedure uses a 50×50 grid.

K
performs a test for complete spatial randomness that is based on the K function.

KERNEL< (*kernel-suboptions***) >**
produces a nonparametric estimate of the first-order intensity, or a nonparametric smoothed estimate of a quantitative mark variable of the point pattern, depending on the *kernel-suboptions*. When you do not specify the *kernel-suboptions*, PROC SPP computes a nonparametric intensity estimate that is based on a default bandwidth and uses a Gaussian kernel. You can specify the following *kernel-suboptions*.

TYPE=EPANECHNIKOV | GAUSSIAN | QUARTIC | TRIANGULAR | UNIFORM
specifies the kernel type for obtaining the nonparametric estimate. For more information about the different kernel types that PROC SPP supports, see the section "Nonparametric Intensity Estimation" on page 9302. By default, TYPE=GAUSSIAN.

B=*value*
specifies the *value* for the kernel bandwidth parameter. The bandwidth is a nonnegative number. By default, the SPP procedure uses a bandwidth of $0.1/\sqrt{\lambda}$, where λ is the CSR average intensity of the point pattern (Illian et al. 2008, p. 236).

ADAPTIVE
performs adaptive kernel estimation. Adaptive kernel estimation requires an initial bandwidth value to compute bandwidth estimates for each data point. If you specify a bandwidth in the B= *kernel-suboption*, then the SPP procedure uses this value as the initial bandwidth. Otherwise, it uses a default bandwidth value that is based on the suggestion by Illian et al. (2008, p.236). For more information about adaptive kernel estimation, see the section "Nonparametric Intensity Estimation" on page 9302.

OUT=*SAS-data-set*
specifies the name of a *SAS-data-set* to contain the kernel based nonparametric estimates.

GRID(*value-NX***,** *value-NY***)**
specifies a reference grid for computing the kernel estimate, where *value-NX* represents the number of horizontal divisions and *value-NY* represents the number of vertical divisions. By default, the SPP procedure uses a 50×50 grid.

L
performs a test for complete spatial randomness that is based on the L function.

OUTSIM=*SAS-data-set*
specifies the name of a *SAS-data-set* to contain the results of simulations in distance functions. This option is ignored unless one of the distance functions is specified in the PROCESS statement.

PCF< B=*value* **>**
performs a test for complete spatial randomness that is based on the pair correlation function (PCF) function. The pair correlation function is calculated only when you specify EDGECORR=ON in the PROC SPP statement. You can specify the following suboption:

B=*value*
specifies the bandwidth value to use in the kernel density estimation inside the pair correlation function. The *value* must be a nonnegative real number. Otherwise, it is assigned a default value of $0.1/\sqrt{\lambda}$, where λ is the CSR average intensity of the point pattern or of the current categorical mark type (Illian et al. 2008, p. 236).

QUADRAT< (< *value-NX*,*value-NY* **> < /DETAILS >) >**
performs a test for complete spatial randomness. You can specify *value-NX* and *value-NY* to provide a quadrat specification that includes the number of horizontal and vertical divisions. If you do not specify the number of horizontal and vertical divisions, PROC SPP computes a default quadrat of 10×10. By default, the QUADRAT option displays only the Pearson chi-square test for CSR. If you also specify the DETAILS suboption, then PROC SPP displays the quadrat count in addition to the Pearson residual information.

When you specify an F, G, J, K, L, or PCF *process-option* (shown in Table 111.5), you can also specify the *distance-function-options* shown in Table 111.6.

Table 111.6 Distance Function Options

Option	Description
BYTYPE	Requests categorical mark typewise calculation of distance functions
CROSS	Requests cross-type distance function analysis that is based on the categorical mark that is specified in the MARK= option
MAXDIST=	Specifies the ending distance for distance functions
MINDIST=	Specifies the starting distance for distance functions
NDIST=	Specifies the number of distances to use for different distance functions
NSIM=	Specifies the number of simulations to compute the CSR envelope
BLOCKS	Specifies the block size for calculation of confidence intervals for distance functions

BYTYPE(ALL|*value-list*)
requests distance function calculation by values of the mark variable. This option produces individual distance function calculations for each mark type. You can specify the following options:

ALL
requests distance function calculation for all available character mark variable values in the DATA= data set.

value-list
requests distance function calculation for certain formatted mark variable values, which you specify as quoted strings in the *value-list*.

CROSS=TYPES(*value-list1***<,***value-list2***>)**
requests cross-type distance function analysis between different mark values. For cross-type analysis, you must specify a mark variable in the point pattern definition by using the MARK= *pattern-option*. The CROSS= option applies only to any requested distance functions K, L, G, J, or PCF. You must specify the TYPES suboption as follows:

TYPES(*value-list1***<,***value-list2***>)**
requests cross-type analysis only among types that are specified in *value-list1* and an optional *value-list2*. If you specify only *value-list1*, then PROC SPP performs cross-type analysis within all the types that are specified in *value-list1*. If you also specify the additional *value-list2*, PROC SPP performs cross-type analysis across both lists. For *value-list1* and *value-list2*, specify quoted strings that correspond to values of the variable that is specified in the MARK=*pattern-option*.

MAXDIST=*value* | **MAX** | **CUT**
specifies the option to be used for computing the maximum distance for different distance functions. You can specify the following options:

value
specifies a *value* for the maximum distance for performing distance function calculations. The *value* must be positive and larger than the value of the MINDIST=*value* option. You can specify any positive value for the maximum distance. However, values that are too large might produce artifacts that do not reflect the true underlying process.

MAX
uses the maximum possible distance, based on the suggestion by Baddeley and Turner (2013). The maximum possible distance is calculated as follows:

- For the K and L functions, the maximum possible distance is calculated as

$$\min\{\min\{\text{Range}(x), \text{Range}(y)\}/4, \sqrt{1000/(\pi \times \lambda)}\}$$

  where λ is the intensity of the point pattern in the study area and the ranges of x and y are computed over the minimum bounding rectangular window of the study area.
- For the PCF functions, the maximum possible distance is calculated as in the case of K and L functions except that the ranges of x and y are computed over a block division of the study area and the λ corresponds to the intensity in a block division. The computed maximum distance for the PCF distance is the minimum of the maximum distance computed over all the block divisions in the study area.
- For the F and G functions, the maximum possible distance is calculated as

$$\min\{\text{Diameter}(W)/2, \sqrt{\log(100000)/(\pi \times \lambda)}\}$$

  where λ is the intensity of the point pattern in the study area and W is the minimum bounding rectangular window of the study area.
- For the J function, the maximum possible distance is calculated as $\text{Diameter}(W)/2$.

CUT
uses the maximum distance at certain cutoff values that are recommended by Baddeley (2014). The cutoff values are as follows:

- for the F and G functions, the distance at which the F or G value reaches 0.9
- for the J function, the distance at which the F or G value in the calculation of the J function reaches 0.9
- for the PCF function, the distance that corresponds to the MAX option that is applied to individual subdivisions of the study area for computing the confidence interval of the PCF statistic
- for the K and L functions, the distance that corresponds to theMAX option that is applied to the entire study area

By default, PROC SPP uses the value of MAXDIST is CUT.

MINDIST=*value*
specifies a positive number for the minimum distance (or starting) distance for all distance function calculations. The *value* of this option cannot be more than the value of MAXDIST= option.

NDIST=*value*
specifies the number of distance bins with which to compute all the specified distance functions. This is a global option that applies to all specified distance functions. When you specify a *value* for this option, the SPP procedure uses this *value* instead of others for distance function calculations.

NSIM=*value*
specifies a positive integer for the number of simulations to be used to compute envelopes for the CSR tests in all distance functions. When you specify this option, it applies to all specified distance functions.

BLOCKS(*NX***,** *NY***)**
specifies the block size that is required for calculating the confidence intervals of distance functions, where *NX* specifies the number of horizontal blocks and *NY* specifies the number of vertical blocks. The block size should be neither too small nor too large for this option to behave reasonably. For more information about estimating the confidence intervals for distance functions see the section "Confidence Intervals for Summary Statistics" on page 9299. The default block size is 5×5.

TREND Statement

TREND *name* **= FIELD***(field-definition)* **;**

The TREND statement enables you to define a spatial trend covariate, where *name* is a standard SAS variable name that names the trend and the FIELD suboption describes the field as follows:

FIELD (*X-variable***,** *Y-variable***,** *field-variable* **)**
specifies a spatial field variable as a trend by using any spatial field covariates that are available in the DATA= data set, where *X-variable* specifies the X coordinate and *Y-variable* specifies the Y coordinate of the spatial field. The third argument is the *field-variable*, which is a numeric variable in the DATA= data set. The *X-variable* and *Y-variable* should be the same as the ones in the PROCESS statement. If you specify a different *X-variable* and a different *Y-variable* from the ones specified in the PROCESS statement then, PROC SPP will produce an error.

Details: SPP Procedure

Testing for Complete Spatial Randomness

The homogeneous Poisson point process serves as a reference model for a completely spatially random (CSR) point pattern. A homogeneous Poisson point process that has intensity $\lambda > 0$ has the following properties:

- The number of points $N(X \cap W)$ that fall in any region W has a Poisson distribution whose mean is $\lambda \times |W|$, where $|W|$ denotes the area of W.
- If W_1 and W_2 are disjoint sets, then $N(X \cap W_1)$ and $N(X \cap W_2)$ are independent random variables.
- The $N(X \cap W)$ points within a study area W are independent and uniformly distributed.

Quadrat Count Test for CSR

The quadrat test is a test of complete spatial randomness (CSR) that uses the χ^2 statistic based on quadrat counts. In the quadrat test, the study area window W is divided into subregions called quadrats ($W_1, W_2, ...W_m$) of equal area. The test counts the number of points that fall in each quadrat $n_j = n(X \cap W_j)$ for $j = 1, ..., m$. Under the null hypothesis of CSR, the n_j are iid Poisson random variables. The following Pearson χ^2 test statistic assesses whether there is a departure from the homogeneous poisson process:

$$\chi^2 = \frac{\sum_j (n_j - n/m)}{n/m}$$

A significant p-value indicates that the underlying point pattern is not CSR.

Exploring Interpoint Interaction

A common question that arises while exploring point pattern data sets is whether points are distributed independent of each other or whether there exists some kind of interaction between points. There are two broad categories of summary statistics, which are based on distances between points:

- nearest-neighbor statistics, such as the F, G, and J functions
- statistics that are based on second-order characteristics, such as the K, L, and g functions (Illian et al. 2008).

The following subsections discuss these statistics in detail.

Nearest-Neighbor Distance Functions

The SPP procedure implements the following nearest-neighbor distance functions:

- empty-space F function
- nearest-neighbor G function
- J function

A typical test that uses any nearest-neighbor function compares the empirical distribution function with the corresponding function for a homogeneous Poisson process that has first-order intensity λ. Usually, the first-order intensity is obtained as the number of observations per unit of area, $\hat{\lambda} = n(x)/|W|$.

The empty-space F function is defined as the empirical distribution function of the observed empty-space distances, $d(g, x)$, which is measured from a set of reference grid points g to the nearest point in the point pattern. The empty-space distance can be defined as

$$d(g, x) = \min\{||g - x_i||, \text{for } x_i \in x\}$$

In practice, the computation of the empty-space F function also involves an edge correction. The edge-corrected empty-space F function is defined as

$$\hat{F}(r) = \sum_j e(g_j, r)\mathbf{1}\{d(g_j, x) \leq r\}$$

where $e(g_j, r)$ is an edge correction. PROC SPP implements the border edge correction (Illian et al. 2008, p. 185–186) as described in the section "Border Edge Correction for Distance Functions" on page 9298.

For a homogeneous Poisson process that has first-order intensity λ, the F function is

$$F_P(r) = 1 - \exp(-\lambda\pi r^2)$$

You compare the empirical and Poisson empty-space F function by using the EDF and the P-P plot in the F function summary panel plot. Values of $\hat{F}(r) > F_P(r)$ suggest a regularly spaced pattern, and values of $\hat{F}(r) < F_P(r)$ suggest a clustered pattern (Baddeley and Turner 2005).

The nearest-neighbor G function is the empirical distribution of the observed nearest-neighbor distance of the points within the point pattern. In practice, the G function also involves an edge correction and is defined as

$$\hat{G}(r) = \sum_i e(x_i, r)\mathbf{1}\{d_i \leq r\}$$

where $e(x_i, r)$ is the border edge correction (Illian et al. 2008, p. 185-186) as described in the section "Border Edge Correction for Distance Functions" on page 9298 and d_i is the distance to the nearest neighbor for the ith point.

For a homogeneous Poisson process that has first-order intensity λ, the G function can be defined as

$$G_P(r) = 1 - \exp(-\lambda\pi r^2)$$

The interpretation of $\hat{G}(r)$ is opposite to the interpretation of $\hat{F}(r)$. That is, values of $\hat{G}(r) > G_P(r)$ imply a clustered pattern, and values of $\hat{G}(r) < G_P(r)$ suggest a regular pattern (Baddeley and Turner 2005).

The third type of nearest-neighbor distance function is the J function, which is defined as a combination of both the F and G functions (Baddeley et al. 2000). The J function is defined for all distances r such that $F(r) < 1$. The J function can be defined as

$$J(r) = \frac{1 - G(r)}{1 - F(r)}$$

For a homogeneous Poisson process, $J_P(r) = 1$. When $J(r)$ takes values greater than 1, regularity is indicated; when $J(r)$ takes values less than 1, the underlying process is more clustered than expected. As can be seen from the expression of $J(r)$, the estimate is an uncorrected estimate of the J-function and hence its computation does not require an edge correction (Baddeley et al. 2000).

Statistics Based on Second-Order Characteristics

Statistics that are based on second-order characteristics include Ripley's K function, Besag's L function, and the pair correlation function (also called the g function). To understand why these functions are based on second-order characteristics, see Illian et al. (2008, p. 223-243). These functions usually involve computation of pairwise distances between points.

The K function of a stationary point process is defined such that $\lambda K(r)$ is the expected number of points within a distance of r from an arbitrary point of the process. The empirical K function of a set of points is the weighted and renormalized empirical distribution function of the set of pairwise distances between points. The empirical K function can be written as

$$\hat{K}(r) = \frac{1}{\hat{\lambda}^2 |W|} \sum_i \sum_{j \neq i} \mathbf{1}\{||x_i - x_j|| \leq r\} e(x_i, x_j; r)$$

where $e(x_i, x_j; r)$ is the border edge correction that is described in the section "Border Edge Correction for Distance Functions" on page 9298.

For a homogeneous Poisson process, $K_P(r)$ can be written as

$$K_P(r) = \pi r^2$$

Exploratory analysis usually involves computing both the empirical K function, $\hat{K}(r)$, and the K function for a Poisson process, $K_P(r)$. A comparison of $\hat{K(r)}$ and $K_P(r)$ might indicate clustering or regularity depending on whether $\hat{K}(r) > K_P(r)$ or $\hat{K}(r) < K_P(r)$.

Besag's L function is a transformation of the K function and is defined as

$$L(r) = \sqrt{\frac{K(r)}{\pi}}$$

For a homogeneous Poisson process, $L_P(r) = r$.

The pair correlation function, *g(r)*, can also be expressed as a transformation of the K function:

$$g(r) = \frac{K'(r)}{2\pi r}$$

Illian et al. (2008), Stoyan (1987), and Fiksel (1988) suggest an alternative expression for $g(r)$:

$$g(r) = \rho(r)/\lambda^2$$

where $\rho(r)$ is the second-order product density function. Cressie and Collins (2001) provides an expression for $\rho(r)$ as

$$\rho(r) = \frac{\hat{\lambda^2} K'(r)}{2\pi r}$$

where $\hat{\lambda}^2 K'(r)$ can be written as a kernel estimate,

$$\hat{\lambda}^2 K'(r) = \frac{1}{a} \sum_{i=1}^{n} \sum_{j \neq i} k_h(||x_i - x_j|| - r)$$

where a is the area, $k_h(u) = k(u/h)/h$, and $k(.)$ is a kernel such as the uniform kernel or the Epanechnikov kernel (Silverman 1986). PROC SPP uses the version that is based on the uniform kernel; for more information about the uniform kernel, see the section "Nonparametric Intensity Estimation" on page 9302. Based on the formula for the second-order product density $\rho(r)$ in terms of the kernel estimate, Stoyan (1987) gives an edge-corrected kernel estimate for $\rho(r)$ as

$$\rho(r) = \frac{1}{2\pi r} \sum_i \sum_{j \neq i} \frac{k_h(||x_i - x_j|| - r)}{a(W_{x_i} \cap W_{x_j})}$$

Dividing $\rho(r)$ by $\hat{\lambda}^2$ gives the pair correlation function $g(r)$ as

$$g(r) = \frac{1}{2\pi r \hat{\lambda}^2} \sum_i \sum_{j \neq i} \frac{k_h(||x_i - x_j|| - r)}{a(W_{x_i} \cap W_{x_j})}$$

where W_{x_i} indicates the translation of the study area window W by the distance x_i from its origin. The above expression for $g(r)$ was given by Stoyan and Stoyan (1994) using the translation edge correction.

A border-edge-corrected version of $g(r)$ can be written as

$$g(r) = \frac{1}{2\pi r \hat{\lambda}} \frac{\sum_i \sum_{j \neq i} k_h(||x_i - x_j|| - r)}{\sum_i \mathbf{1}\{b_i \geq r\}}$$

where x_i and x_j are points within the boundary at a distance greater than or equal to r; where b_i is the distance of x_i to the boundary of W, ∂W; and where $k_h(u) = k(u/h)/h$ for a kernel $k(.)$, such as the uniform kernel or the Epanechnikov kernel. For more information about the uniform kernel, see the section "Nonparametric Intensity Estimation" on page 9302. For a homogeneous Poisson process, $g(r) = 1$. For any point pattern, values of $g(r)$ greater than 1 indicate clustering or attraction at distance r, whereas values of $g(r)$ less than 1 indicate regularity.

Distance Functions for Multitype Point Patterns

Distance functions (such as G, J, K, L, and g) can also be defined for point patterns that are "marked" with a categorical mark variable, called a type. Usually you consider mark variables that have more than one type to define distance functions. When distance functions are defined between two types, they are called cross-type distance functions. For any pair of types i and j, the cross-distance functions G_{ij}, J_{ij}, K_{ij}, L_{ij}, and g_{ij} can be defined analogously to the single-type distance functions. The interpretation of cross-type distance functions is slightly different from the interpretation of single type functions. Suppose that X is the point pattern, X_j refers to the subpattern of points of type j; X_i refers to the subpattern of points of type i, and λ_j represents the intensity of the subpattern X_j. Then the interpretation is to treat X_j as a homogeneous Poisson process and independent of X_i. If the computed empirical cross-type function is identical to the function that corresponds to a homogeneous Poisson process, then X_i and X_j can be treated as independent of each other.

The empirical cross-G-function, G_{ij}, is defined as the distribution of the distance from a point of type i in X_i to the nearest point of type j in X_j. Formally, G_{ij} can be written as

$$G_{ij}(r) = \sum_i e(x_i, r) \mathbf{1}\{d_{ij} \leq r\}$$

where $e(x_i, r)$ is an edge correction and d_{ij} is the distance from a point of type i to the nearest point of type j. If the two subpatterns X_i and X_j are independent of each other, then the theoretical cross-G-function is

$$G^*_{ij}(r) = 1 - \exp(\lambda_j \pi r^2)$$

The empirical cross-type J-function, J_{ij}, can be defined again in terms of the G_{ij} function and the empty-space F function for subpattern X_j as

$$J_{ij}(r) = J_{ij}(r) = \frac{1 - G_{ij}(r)}{1 - F_j(r)}$$

where $F_j(r)$ is the empty-space function for the subpattern X_j. If the two subpatterns X_i and X_j are independent of each other, then the theoretical cross-J-function is $J^*_{ij}(r) = 1$.

The empirical cross-type K function, K_{ij}, is $1/\lambda_j$ times the expected number of points of type j within a distance r of a typical point of type i. Formally, K_{ij} can be written as

$$K_{ij}(r) = K_{ij}(r) = \frac{1}{\lambda_j \lambda_i |W|} \sum_i \sum_j \mathbf{1}\{||x_i - x_j|| \le r\} e(x_i, x_j; r)$$

where $e(x_i, x_j; r)$ is an edge correction. If the two subpatterns X_i and X_j are independent of each other, then the theoretical cross-K-function is $K^*_{ij}(r) = \pi r^2$.

The empirical cross-type L function, L_{ij}, is a transformation of K_{ij}. Formally, L_{ij} can be written as

$$L_{ij}(r) = L_{ij}(r) = \sqrt{\frac{K_{ij}(r)}{2\pi r}}$$

If the two subpatterns X_i and X_j are independent of each other, then the theoretical cross-type L-function is $L^*_{ij}(r) = r$.

The empirical cross-type pair correlation function, g_{ij}, is a kernel estimate of the form

$$g_{ij}(r) = \frac{\rho(r)}{\hat{\lambda}^2} = \frac{1}{2\pi r \hat{\lambda}_i \hat{\lambda}_j} \sum_i \sum_j \frac{k_h(||x_i - x_j|| - r)}{|W \cap W_{i-j}|}$$

Based on the definition of Stoyan and Stoyan (1994), $g_{ij}(r)$ can be written as

$$g_{ij}(r) = \frac{\rho(r)}{\hat{\lambda}^2} = \frac{1}{2\pi r \hat{\lambda}_i \hat{\lambda}_j} \sum_i \sum_j \frac{k_h(||x_i - x_j|| - r)}{|W_{x_i} \cap W_{x_j}|}$$

A border-edge-corrected version of $g_{ij}(r)$ can be written as

$$g_{ij}(r) = \frac{1}{2\pi r \hat{\lambda_j}} \frac{\sum_i \sum_j k_h(||x_i - x_j|| - r)}{\sum_i \mathbf{1}\{b_i \ge r\}}$$

where b_i is the distance of x_i to the boundary of W, which is denoted as ∂W. If the two subpatterns X_i and X_j are independent of each other, then the theoretical cross-type pair correlation function is $g^*_{ij}(r) = 1$.

Border Edge Correction for Distance Functions

To compute the edge correction factors $e(x_i, r)$ that appear in the formulas of the distance functions, the SPP procedure implements border edge correction (Illian et al. 2008; Ripley 1988; Baddeley 2007). Border edge correction is necessary because the data are given for a bounded observation window W, but the pattern itself is assumed to extend beyond the observation window. However, because you can observe only what is within the window, a disc $b(x, r)$ of radius r around a point x that lies close to the boundary of W might extend outside W. Because the original process X is not observed outside W, the number of points of X in $b(x, r)$ is not observable (Baddeley 2007). Ignoring the fact that the observable quantity $n(X \cap W \cap b(x, r))$ is less

than or equal to $n(X \cap b(x,r))$ leads to a bias that is caused by edge effects. The border edge corrector is a simple strategy to eliminate the bias that is caused by edge effects. Under the border method, the window W is replaced by a reduced window,

$$W_{\ominus r} = W \ominus b(0,r) = \{x \in W : ||x - \partial W|| \geq r\}$$

where $||x - \partial W||$ denotes the minimum distance from X to a point on the boundary. The reduced window contains all the points in W that are at least r units away from the boundary ∂W.

Based on the preceding definition, the border edge corrected F, K, and G functions are

$$\hat{F}(r) = \frac{1}{\lambda |W_{\ominus r}|} \sum_{g_j \in W_{\ominus r}} \mathbf{1}\{d(g_j, x) \leq r\}$$

$$\hat{K}(r) = \frac{\sum_{i=1}^{n} \sum_{j \neq i} \mathbf{1}\{||x_i - x_j|| \leq r\}}{\hat{\beta} n(x \cap W_{\ominus r})}$$

$$\hat{G}(r) = \frac{\sum_{x_i \in W_{\ominus r}} \mathbf{1}\{||x_i - X/\, x_i|| \leq r\}}{n(X \cap W_{\ominus r})}$$

$$\hat{G}(r) = \frac{\sum_i \mathbf{1}\{d_i \leq r, b_i \geq r\}}{\sum_i \mathbf{1}\{b_i \geq r\}}$$

where $\hat{\beta} = n(x)/\lambda |W|$; $||x_i - X/\, x_i||$ is the observed nearest-neighbor distance, d_i, for the ith point x_i; and b_i is the distance from x_i to the boundary ∂W. For more information about these border-edge-corrected functions, see Baddeley (2007).

Confidence Intervals for Summary Statistics

The SPP procedure computes confidence intervals for the true value of a summary statistic such as the K, L, F, G, J, or PCF function. The window that contains the point pattern is divided into a number of blocks. By default, PROC SPP divides the window into 5×5 blocks. The summary statistic is calculated in each block, and the pointwise sample mean, sample variance, and sample standard deviation of these summary statistics are computed. If any edge corrections are required, they are also applied in the calculation of the individual summary statistics within each block. If the summary statistic is a function such as the K function, the estimate for a particular block B is computed by counting pairs of points in which the first point lies in B and the second point lies elsewhere (Baddeley and Turner 2013).

The variance of the summary statistics is estimated by

$$\text{var}(\tilde{K}(t)) \approx \frac{\sum_{i=1}^{m} \{k_i - \tilde{K}(t)\}}{\{m(m-1)\}}$$

where m is the number of blocks, k_i is the value of the summary statistic in individual blocks, and $\tilde{K}(t) = \sum_{i=1}^{m} k_i/m$ (Diggle 2003, pp. 52–53).

Ripley-Rasson Window Estimator

When the sampling window for a point pattern is unknown, it can be estimated from the data. A common error is to assume that W is the smallest rectangle that contains the data points, or is the convex hull that encloses the data points (Baddeley 2010). Either choice is an underestimate of the true region W and usually yields an overestimate of the point process intensity λ or summary statistics such as the K function. The Ripley-Rasson window estimator is an estimate of the spatial window from which the points were drawn (Ripley and Rasson 1977). For estimating a rectangular study region, the Ripley-Rasson estimate is the rescaled copy of the minimum bounding box of the collection of points, centered at the box's centroid and expanded using a scaling factor of $1/\sqrt{1-\frac{4}{n}}$, where n is the number of data points.

Covariate Dependence Tests

For analyses that include covariates, the SPP procedure implements nonparametric goodness-of-fit tests that are based on the empirical distribution function (EDF). PROC SPP provides weighted EDF tests that depend primarily on the covariate values.

The next subsection reviews the EDF tests that are at the heart of covariate dependency testing, and the subsequent subsection describes the covariate dependency tests in more detail.

EDF Goodness-of-Fit Tests

You use goodness-of-fit tests to examine the fit of a parametric distribution. In the SPP procedure, this task emerges when you test your data for dependence on a covariate. You can examine the goodness of fit by using tests that are based on the EDF. These tests offer advantages over traditional chi-square goodness-of-fit tests, as discussed in D'Agostino and Stephens (1986). The empirical distribution function is defined for a set of n independent observations, $X_1, \ldots, X_n$, that have a common distribution function $F(x)$ as follows. Denote the observations ordered from smallest to largest as $X_{(1)}, \ldots, X_{(n)}$. Then the empirical distribution function, $F_n(x)$, is

$$F_n(x) = \begin{cases} 0, & x < X_{(1)} \\ \frac{i}{n}, & X_{(i)} \le x < X_{(i+1)} \quad i = 1, \ldots, n-1 \\ 1, & X_{(n)} \le x \end{cases}$$

$F_n(x)$ is a step function that takes a step of height $\frac{1}{n}$ at each observation. This function estimates the distribution function $F(x)$. At any value x, $F_n(x)$ is the proportion of observations that are less than or equal to x, whereas $F(x)$ is the probability of an observation being less than or equal to x. EDF statistics measure the discrepancy between $F_n(x)$ and $F(x)$.

The computational formulas for the EDF statistics make use of the probability integral transformation $Z = F(X)$. If $F(X)$ is the true distribution function of X, then the random variable Z is uniformly distributed between 0 and 1. For example, assume that you believe $X \sim N(\mu, \sigma^2)$. In this case, the probability integral transform $Z = F(X)$ for the normal $N(\mu, \sigma^2)$ is given by the EDF of the standardized value $(X - \mu) / \sigma$. To test the fit of your sample EDF $F_n(x)$ to the assumed exact $F(X)$, you can equivalently test the fit of $F_n(z)$ to the EDF $F(Z)$ of Z. As $Z \sim U(0, 1)$, $F(Z)$ is the cumulative density function (CDF) of the standard uniform $U(0, 1)$, which is simply $F(Z) = z$. This also means that your empirical $F_n(x) = F_n(z)$. Consequently,

the probability integral transform translates the initial fit task into an easier comparison between $F_n(z)$ and $F(Z)$.

There are two main classes of EDF statistics: the supremum and the quadratic class. The supremum class is based on the largest vertical difference between $F(x)$ and $F_n(x)$. The quadratic class is based on the squared difference $(F_n(x) - F(x))^2$. Quadratic statistics have the following general form:

$$Q = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 \psi(x) dF(x)$$

The function $\psi(x)$ weights the squared difference $(F_n(x) - F(x))^2$.

As previously discussed, the SPP procedure considers the ordered observations $X_{(1)}, \ldots, X_{(n)}$ and computes the values $Z_{(i)} = F(X_{(i)})$ by applying the probability integral transform. PROC SPP examines the goodness of fit by computing the following two EDF statistics:

- Kolmogorov-Smirnov two-sided D from the supremum class
- Cramér-von Mises W^2 from the quadratic class

Within the different classes of EDF statistics, the quadratic class is known to have more powerful statistics than the supremum class. The details of the statistics used by PROC SPP are discussed in the following subsection.

After the EDF test statistics are computed, the SPP procedure computes the associated significance values. In the scope of the PROC SPP analysis, the true distribution function, $F(X)$, is a completely specified distribution. For computations in this scenario, PROC SPP applies slightly modified D and W^2 statistics, as described by D'Agostino and Stephens (1986).

Testing Covariate Dependency with EDF Tests

In a test for covariate dependency, the goal is to test the null hypothesis H_0 that the point process is independent of the covariate. PROC SPP tests H_0 by interpolating the covariate values at the event locations. The EDF is weighted by the intensity at the corresponding locations (Baddeley and Turner 2005). PROC SPP performs this weighted EDF test for covariates that are defined in a TREND statement.

Weighted EDF Tests

To test dependence on a trend covariate, PROC SPP initially computes the covariate EDF. The EDF is weighted by using intensity-based weights to account for the current intensity model. For example, under the CSR assumption the intensity λ is constant across the study area; hence, the weight for each of the M observations of a covariate is $\lambda_i / \sum_i M\lambda_i = \lambda / M\lambda = 1/M$. This weighted EDF is the predicted distribution that any other set of independent covariate observations should follow under the assumed intensity model.

Next, the covariate is interpolated at the n event locations $\boldsymbol{s}_i, i = 1, \ldots, n$, using ordinary kriging; kriging analysis assumes a linear semivariance correlation function and considers the four closest covariate observations for each event location. The outcome is a set of covariate values X_i. With the X_i in hand, PROC SPP assumes that the probability integral transform $Z = F(X)$ is the linear interpolation of the weighted EDF at the covariate values X_i, and it produces the transformed EDF Z in $[0, 1]$. If the intensity model assumption is correct, then Z follows a uniform distribution $U(0, 1)$. Finally, PROC SPP uses EDF tests to examine the fit of the EDF $F_n(x) = F_n(z)$ to a standard uniform EDF.

Nonparametric Intensity Estimation

The KERNEL option in the PROCESS statement enables you to perform nonparametric intensity estimation. You can use five different kernel types: Epanechnikov, Gaussian, uniform, triangular, and quartic (Silverman 1986), whose kernel functions are as follows, where $t = \sqrt{(s_x - x)^2 + (s_y - y)^2}/h$, s_x, s_y are the grid point coordinates, x and y are the point coordinates, and h is the bandwidth parameter:

- Epanechnikov

$$K(t) = \begin{cases} \frac{3}{4}(1 - \frac{t^2}{5})\frac{1}{\sqrt{5}} & |t| < \sqrt{5} \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian

$$K(t) = \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}}$$

- uniform

$$K(t) = \begin{cases} \frac{1}{2} & |t| < 1 \\ 0 & \text{otherwise} \end{cases}$$

- triangular

$$K(t) = \begin{cases} 1 - |t| & |t| < 1 \\ 0 & \text{otherwise} \end{cases}$$

- quartic

$$K(t) = \begin{cases} \frac{15}{16}(1 - t^2)^2 & |t| < 1 \\ 0 & \text{otherwise} \end{cases}$$

Given the preceding kernel definitions, the nonparametric intensity estimate can be computed as

$$\lambda(s) = \sum_{i=1}^{n} h^{-2} \times K\left(\frac{s - s_i}{h}\right)$$

where h is the fixed bandwidth. In practice, nonparametric intensity estimation also involves an edge correction. By default, PROC SPP divides the nonparametric estimate $\lambda(s)$ by an edge correction factor

$$\rho(s) = \int_A h^{-2} \times K\left(\frac{s - s_i}{h}\right)$$

where A is the study area. The choice of the bandwidth parameter that nonparametric intensity estimation requires is more important than the choice of the kernel type itself (Silverman 1986). The bandwidth can be spatially fixed or spatial varying. If the bandwidth is spatially varying, it is called adaptive kernel estimation. For adaptive kernel estimation, the SPP procedure uses the technique suggested in Silverman (1986, p. 101) and Diggle, Rowlingson, and Su (2005, p. 426), which is computed in two steps:

1. Use an initial bandwidth h to compute pilot estimates of the first-order intensity as

$$\lambda_0(s) = \sum_{i=1}^{n} h^{-2} \times K\left(\frac{s - s_i}{h}\right)$$

   where *K(.)* is a kernel.

2. Compute bandwidth factors as

$$h_i = h \times \left(\frac{\lambda_0(s)}{\widehat{g}}\right)^{-0.5}$$

   where $\widehat{g}$ is the geometric mean of the pilot estimates $\lambda_0(s)$.

Based on the computed bandwidth estimates, h_i, the nonparametric intensity estimates are computed as

$$\lambda(s) = \sum_{i=1}^{n} h_i^{-2} \times K\left(\frac{s - s_i}{h_i}\right)$$

In PROC SPP, adaptive kernel estimation does not incorporate edge correction.

Inhomogeneous Poisson Process Model Fitting

An inhomogeneous Poisson process that has intensity function $\lambda(s)$ is a point process in which the number of points that fall in a spatial region W, $N(X \cap W)$ has the following expectation:

$$\mathbb{E}[N(X \cap W)] = \int_W \lambda(s)ds$$

Also, the $N(X \cap W)$ points are independent and identically distributed for disjoint subsets W with a probability density of

$$f(s) = \frac{\lambda(s)}{\int_W \lambda(s)ds}$$

.

Likelihood Methods for Model Fitting

The intensity function $\lambda_\theta(s)$ is assumed to be log linear in the parameters θ. So

$$\log \lambda_\theta(s) = \theta \cdot Z(s)$$

where $Z(s)$ is a real-valued or vector-valued function of location s. $Z(s)$ can include a polynomial function of coordinate variables or a spatial covariate. The log likelihood for the parameters θ is given by

$$\log L(\theta; x) = \sum_{i=1}^{n} \log \lambda_\theta(x_i) - \int_W \lambda_\theta(s)ds$$

The integral in the expression for the log likelihood can be approximated using quadrature as

$$\int_W \lambda_\theta(s;x)ds \approx \sum_{j=1}^{m} \lambda_\theta(s_j;x)w_j$$

for some quadrature weights w_j. Hence, the log likelihood can be rewritten as follows:

$$\log L(\theta;x) \approx \sum_{i=1}^{n(x)} \log \lambda_\theta(x_i;x) - \sum_{j=1}^{m} \lambda_\theta(s_j;x)w_j$$

Based on the observation by Berman and Turner (1992) and Baddeley and Turner (2000), the log likelihood can be approximated as

$$\log L(\theta;x) \approx \sum_{j=1}^{m} (y_j \log \lambda_j - \lambda_j)w_j$$

where $\lambda_j = \lambda_\theta(s_j)$. If the list of points $\{s_j, j = 1,\ldots,m\}$ also includes the collection of data points $\{x_i, i = 1,\ldots,n\}$, then $y_j = z_j/w_j$ and

$$z_j = \begin{cases} 1 & \text{if } s_j \text{ is a data point, } s_j \in x \\ 0 & \text{if } s_j \text{ is a dummy point, } s_j \notin x \end{cases}$$

The log pseudolikelihood can be maximized using standard optimization algorithms.

Fit Statistics

The SPP procedure displays three fit statistics for model selection. For a model that has p parameters, uses n event observations, and produces a maximum log likelihood Log L, these criteria are calculated as in Table 111.7.

Table 111.7 Fit Statistics

Option	Description
–2 log likelihood	$2LL = -2 \text{ Log L}$
Akaike's information criterion (AIC)	$AIC = -2 \text{ Log L} + 2p$
Schwarz criterion or Bayesian information criterion (BIC)	$BIC = -2 \text{ Log L} + p \log(n)$

The AIC and BIC statistics give two different ways of adjusting the –2 Log L statistic for the number of terms in the model and the number of observations used. These statistics can be used when different models for the same data are compared. Lower values of the statistics indicate a more desirable model.

Fitted Model Validation That Uses Goodness-of-Fit Tests

If you want to check how likely the data are to be generated by the fitted model, you can perform a goodness-of-fit test that is based on a chi-square statistic. The model goodness-of-fit test that is displayed by the GOF option in the MODEL statement uses quadrats to compute the observed and expected counts and subsequently to perform the chi-square test. The model goodness-of-fit test is a simulation-based test that uses the fitted model to generate different realizations of the point process. For each simulated realization, the SPP procedure calculates the expected count under the model and computes the mean of this expected count over all the realizations. The mean of this expected count over all realizations is used to compute a Pearson residual as

$$\text{Pearson residual} = \frac{O_c - E_c}{\sqrt{E_c}}$$

where O_c is the observed count in each quadrat, based on the data, and E_c is the expected count under the model. Based on these observed and expected counts, a chi-square statistic is computed and a Pearson chi-square test is performed. A small p-value indicates that the data are not likely to be generated by the model.

Fitted Model Validation That Uses Residuals

Residual diagnostics are tools for checking and examining the fitted model. Residual plots and influence diagnostics help you identify influential observations, assess model assumptions, and recognize departures from the model. Baddeley et al. (2005) define four types of residuals: raw residuals, inverse residuals, Pearson residuals, and score residuals. PROC SPP implements only raw residuals. Given a point pattern x and using a parameter estimate $\hat{\theta} = \hat{\theta}(x)$, the raw residuals can be defined as

$$R_{\hat{\theta}}(W) = n(x \cap W) - \int_W \hat{\lambda}(s, x)ds$$

In order to be able to compute the raw residual, Baddeley et al. (2005) suggest a discretization of this residual measure. According to Baddeley and Turner (2013), discretization of the raw residuals yields

$$r_j = z_j - w_j \lambda_j$$

at the quadrature points u_j, where z_j is an indicator equal to 1 if u_j is a data point or 0 if u_j is a dummy point, w_j is the quadrature weight that is attached to u_j, and $\lambda_j = \hat{\lambda}(u_j, x)$ is the conditional intensity of the fitted model at u_j.

Smoothed Residuals

The smoothed raw residuals are defined as

$$s(u) = \hat{\lambda}(u) - \tilde{\lambda}(u)$$

where $\hat{\lambda}(u)$ is a nonparametric kernel estimate of the intensity,

$$\hat{\lambda}(u) = e(u) \sum_{i=1}^{n(x)} k(u - x_i)$$

where $e(u)$ is an edge correction and $\tilde{\lambda}(u)$ is a smoothed version of the parametric estimate of the intensity according to the fitted model:

$$\tilde{\lambda}(u) = e(u) \int_W k(u-s)\lambda_{\hat{\theta}}(s)ds$$

If the fitted model is correct, the kernel estimate and the kernel smoothed estimate of the fitted intensity should be approximately equal. Positive values of $s(u)$ suggest that the model underestimates the intensity (Baddeley and Turner 2005).

Lurking Variable Plots

Lurking variable plots help detect dependence on an unobserved covariate. Any systematic pattern in these plots indicate a departure from the model (Baddeley and Turner 2005). For point process models, you can plot the residuals against a spatial covariate or one of the coordinates to investigate the presence of a spatial trend and to assess whether the true trend differs from the trend that is specified by the fitted model. For a spatial covariate $Z(u)$ that is defined at each location $u \in W$, the residual on each sublevel set,

$$W(z) = \{u \in W : Z(u) \leq z\}$$

yields a cumulative residual function for the raw residuals as follows:

$$A(z) = n(\{x \cap W(z)\}) - \int_{W(z)} \hat{\lambda}(u,x)du$$

In addition to plotting the cumulative residual function, the lurking variable plot also shows 2σ limits based on the variance of the innovations under an inhomogeneous Poisson process (Baddeley et al. 2005). The variance of the innovations under an inhomogeneous Poisson process is

$$\text{var}\{A(z)\} = \text{var}\{I\{W(z)\}\} = \int_{W(z)} \lambda(u)du$$

The 2σ limits can be interpreted as pointwise significance limits. A systematic violation of the limits suggests that the proposed model does not account for the dependence on the covariate under consideration (Baddeley et al. 2005).

Negative Binomial Modeling

Spatial data in many applications can be overdispersed—that is, the variance of the counts of spatial events might be inflated relative to what is expected under the assumption of a Poisson model. The negative binomial modeling in PROC SPP serves as a diagnostic to assess overdispersion in your data.

The log likelihood for the negative binomial model is

$$\log L(\theta;x) = \sum_{i=1}^{n} \left\{ y_i \cdot \log\left\{\frac{\phi \cdot \lambda_i}{w_i}\right\} - (y_i + \frac{w_i}{\phi}) \cdot \log\left\{1 + \frac{\phi \cdot \lambda_i}{w_i}\right\} + \log\left\{\frac{\Gamma(y_i + w_i/\phi)}{\Gamma(w_i/\phi)\Gamma(y_i+1)}\right\}\right\}$$

where y_i is the response at a location, λ_i is the intensity associated with the location, ϕ is the negative binomial scale parameter, and w_i is the quadrature weight associated with the location. Overdispersion is indicated by the value of the scale parameter ϕ: for $\phi = 0$, the negative binomial distribution is identical to the Poisson distribution; therefore, large values of ϕ indicate overdispersion.

Because the negative binomial model in PROC SPP is intended only for diagnostic purposes, the only results are the estimated parameters themselves, including ϕ. No fitted intensity is produced, and likewise nothing that depends on the fitted intensity, such as the goodness-of-fit tests and the residual diagnostics, is produced.

Output Data Sets

The SPP procedure produces output data sets that are specified in the OUT= suboption of the KERNEL option in the PROCESS statement, and in the OUTINTESITY= and OUTSIM= options in the MODEL statement. These data sets are described in the following sections.

OUT= Suboption in the KERNEL Option in the PROCESS Statement

This suboption specifies the name of an output data set to store the kernel-based nonparametric estimates. This data set contains the following variables:

- KERNEL, the kernel type that is used for the corresponding intensity estimate
- BANDW, the kernel bandwidth that is used for the corresponding intensity estimate
- VARNAME, the label of the event variable
- GXC, the x coordinate of the grid point at which the intensity estimate is made
- GYC, the y coordinate of the grid point at which the intensity estimate is made
- ESTIMATE, the intensity estimate

OUTSIM= Option in the PROCESS Statement

This option specifies the name of an output data set to store the simulations of distance functions that are included as options in the PROCESS statement. This data set contains the following variables:

- GXC, X coordinate of current simulated event location
- GYC, Y coordinate of current simulated event location
- ITER, current iteration
- IXY, simulated event observation number in the current iteration
- MARK, label of current mark
- MTYPE, categorical mark level
- MVALUE, numeric mark simulated value

- NITER, number of iterations in simulation
- NXY, number of simulated events in current iteration
- VARNAME, label of the event the results refer to

OUTINTENSITY= Option in the MODEL Statement

This option specifies the name of an output data set to store the output intensity estimate. This data set contains the following variables:

- ESTIMATE, fitted intensity at current location
- GXC, X coordinate of current output grid location
- GYC, Y coordinate of current output grid location
- STDERR, standard error for the intensity estimate
- VARNAME, event label from the DATA= data sets that are associated with events and identified during the procedure call
- PROB, probability of occurrence based on a logit transformation of the fitted intensity estimate at the grid. This represents the probability that the number of events at the location is not zero.

OUTSIM= Option in the MODEL Statement

This option specifies the name of an output data set to store a simulated point pattern from a fitted intensity model. This data set contains the following variables:

- ESTIMATE, fitted intensity estimate at the simulated event location
- GXC, X coordinate of current simulated event location
- GYC, Y coordinate of current simulated event location
- ITER, current simulation iteration
- NITER, number of iterations in current simulation
- NXY, number of simulated events in current iteration
- VARNAME, event label from the DATA= data set that is associated with events and identified during the procedure call

Displayed Output

The SPP procedure produces the following output objects.

- By default, PROC SPP outputs a "Number of Observations" table, which displays the number of observations that are read from the input data set and the number of valid observations that are used. The actual number of observations that are used in the analysis can be equal to or smaller than the number of valid observations, depending on the specification of the study window and the existence and handling of duplicate observations. When you include a covariate variable in your analysis, this table contains more detailed information about the number of observations that are used in the study window for each variable.
- If you use the PROCESS statement to specify a point pattern, PROC SPP outputs a table is displayed by default that contains exploratory information about the point pattern, any mark variable that is present, and information about the point pattern domain window and grid.
- If you use the PROCESS statement to specify a point pattern, PROC SPP outputs a default plot of the event observations in the point pattern.
- If you specify a mark variable for the point pattern, PROC SPP outputs a table that contains information about the mark variable.
- If you do not specify any options for the PROCESS statement, PROC SPP performs the quadrat test by default and outputs a table that shows the Pearson chi-square test for CSR by default. If you specify the QUADRAT option with the DETAILS suboption, PROC SPP outputs a detailed quadrat counts table, a quadrat information table that contains Pearson residuals, and a table for the Pearson chi-square test for CSR.
- If you specify a KERNEL option in the PROCESS statement, PROC SPP outputs a kernel intensity information table. In addition, if you request the ADAPTIVE suboption in the KERNEL option, an adaptive kernel information table is displayed. In addition, if you specify the KERNEL option and ODS Graphics is enabled, a map of the kernel intensity estimate is also produced.
- If you specify one or more of the K, L, G, and F options in the PROCESS statement, PROC SPP outputs an information table for each of the specified distance functions. The information table contains basic information, such as the minimum analysis distance, maximum analysis distance, maximum difference between the empirical distribution function of the summary statistic and the CSR function, and the distance at which the maximum difference is observed. In addition, PROC SPP outputs a panel plot for each distance function that is included as an option in the PROCESS statement. Each panel plot contains four constituent plots: the empirical distribution function (EDF) plot, the EDF−CSR difference plot, a probability-probability plot that compares the EDF and CSR, and a confidence interval plot for the summary statistic.
- If you specify the J and PCF options in the PROCESS statement, PROC SPP outputs a combined plot that shows the EDF, the simulation intervals, and the confidence interval for the summary statistic.
- If you specify a COVTEST statement that has appropriate trend covariates on the right side, PROC SPP outputs a table for the Kolmogorov-Smirnov EDF test statistic and creates a plot of the empirical and transformed EDF by default for each covariate that you include in the COVTEST statement.

The plot illustrates the Kolmogorov-Smirnov test analysis for testing for point pattern dependency on covariates. If you specify the Cramér–von Mises EDF test statistic in the TEST= option in the COVTEST statement, PROC SPP outputs the table for the Cramér–von Mises EDF test statistic.

- If you specify a MODEL statement to fit a model for the first-order intensity of the point pattern that is defined in a preceding PROCESS statement, PROC SPP produces the following results by default:
  - a "Model Information" table that lists the intercept, covariates, and polynomial terms that are included in the model, along with the initial values for the coefficients
  - an optimization information table that shows the optimization technique, the number of parameters in the optimization, and the number of fixed parameters and starting values
  - a table for the convergence status that shows the convergence criterion
  - a "Parameter Estimates" table that shows the estimate for each parameter, the standard error, the number of degrees of freedom, a t value, and a p-value
  - a "Fit Statistics" table that shows different fit statistics, such as the log likelihood, Akaike's information criterion, and the Bayesian information criterion
  - a map that shows the fitted intensity estimate based on the model
- If you specify the MODEL statement and include the ITHIST option, PROC SPP outputs an iteration history table that shows the value of the objective function and the maximum value of the gradient over different iterations of the optimization algorithm.
- If you specify the MODEL statement and include the CORRB option, PROC SPP outputs the approximate correlation matrix.
- If you specify the MODEL statement and include the COVB option, PROC SPP outputs the approximate covariance matrix.
- If you specify the MODEL statement and include the GOF option, PROC SPP outputs a table that shows the Pearson chi-square test for goodness of fit. This table shows a p-value that indicates how likely it is for the data to be generated by the fitted model. In addition, if you also specify a QUADRAT option with the DETAILS suboption for the response process in a preceding PROCESS statement, PROC SPP also displays a quadrat information table that shows Pearson residuals that are based on the expected counts under the fitted model and observed counts from the point pattern data set that is defined for the response process.

The complete listing of the PROC SPP output follows in the sections "ODS Table Names" on page 9310 and "ODS Graph Names" on page 9312.

ODS Table Names

Each table that PROC SPP creates has a name that is associated with it, and you must use this name to refer to the table when you use ODS Graphics. Table 111.8 lists these names and shows the statement and options that you must specify to produce the table.

Table 111.8 ODS Tables Produced by PROC SPP

ODS Table Name	Description	Statement	Option
CenScale	Model parameter standardization information	MODEL	CENSCALE
CenScaleCorrB	Approximate correlation matrix of model-standardized parameter estimates	MODEL	CORRB, SOLUTION
CenScaleCovB	Approximate covariance matrix of model-standardized parameter estimates	MODEL	COVB, SOLUTION
CenScaleParms	Parameter estimates for standardized output	MODEL	SOLUTION
ConvergenceStatus	Status of optimization at conclusion	MODEL	Default output
CorrB	Approximate correlation matrix of model parameter estimates	MODEL	CORRB
CovariateInfo	Numerical covariate information	COVTEST	Default output
CovariateLevelInfo	Levels of categorical covariate	COVTEST	Default output
CovB	Approximate covariance matrix of model parameter estimates	MODEL	COVB
EdfCsrTest	EDF test for complete spatial randomness	COVTEST	Default output
ExploratoryInfo	General point pattern information	PROC	Default output
FitStatistics	Goodness-of-fit information	MODEL	Default output
IterHist	Iteration history	MODEL	Default output
KernIntensityInfo	Intensity function information from kernel density estimation	PROCESS	KERNEL
MarkInfo	Numerical mark information	PROC SPP	Default output
MarkLevelInfo	Levels of categorical mark	PROC SPP	Default output
ModelInfo	Model information	MODEL	Default output
ModelPearsonsChiSq	Chi-square goodness-of-fit test	MODEL	GOF
ModelQuadratInfo	Detailed fitted model quadrat information	MODEL	GOF
ModelSimulationInfo	Information table for simulating a point pattern from a fitted model	MODEL	OUTSIM
NObs	Number of observations read and used	PROC SPP	Default output
OptInfo	Optimization information	MODEL	Default output
ParameterEstimates	Model-fitting solution and statistics	MODEL	Default output
ParmSearch	Parameter search values	PARMS	Default output
PearsonsChiSq	Chi-square test for CSR	PROCESS	QUADRAT
QuadratCount	Counts of quadrats	PROCESS	QUADRAT

Table 111.8 *continued*

ODS Table Name	Description	Required Statement	Option
QuadratInfo	Detailed quadrat information	PROCESS	QUADRAT/DETAILS
FFuncInfo	F function information	PROCESS	F
GFuncInfo	G function information	PROCESS	G
KFuncInfo	K function information	PROCESS	K
LFuncInfo	L function information	PROCESS	L
SummaryInfo	Summary statistics information	PROCESS	F, G, K, L, PCF, J

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, "Statistical Graphics Using ODS."

Before you create graphs, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 623 in Chapter 21, "Statistical Graphics Using ODS." For additional control of the graphics that are displayed, see the PLOTS= option in the section "PROC SPP Statement" on page 9275.

ODS Graph Names

PROC SPP assigns a name to each graph that it creates by using ODS Graphics. You can use these names to refer to the graphs when you use ODS Graphics. Table 111.9 lists the names and shows the statement and option that you must specify to produce the graph.

Table 111.9 Graphs Produced by PROC SPP

ODS Graph Name	Plot Description	Statement	Option
CovariateEDFPlot	Plot of Kolmogorov-Smirnov test analysis for CSR	COVTEST	PLOTS=CSRKSTEST
EmptySpacePlot	Surface plot of nearest-neighbor distances from any window location	PROCESS	
		PROC SPP	PLOTS=EMPTYSPACE
FCIPlot	Confidence interval plot of empty-space function F	PROCESS	F
		PROC SPP	PLOTS=F(UNPACK)
FDiffPlot	Difference plot of empty-space function F and CSR	PROCESS	F
		PROC SPP	PLOTS=F(UNPACK)
FEdfPlot	EDF plot of empty-space function F	PROCESS	F
		PROC SPP	PLOTS=F(UNPACK)

Table 111.9 *continued*

ODS Graph Name	Plot Description	Statement	Option
FNppPlot	PP plot of empty-space function F and CSR	PROCESS	F
		PROC SPP	PLOTS=F(UNPACK)
FPanelPlot	Panel plot of empty-space function F	PROCESS	F
		PROC SPP	PLOTS=F(ALL)
GCIPlot	CI plot of nearest-neighbor function G	PROCESS	G
		PROC SPP	PLOTS=G(UNPACK)
GDiffPlot	Difference plot of nearest-neighbor function G and CSR	PROCESS	G
		PROC SPP	PLOTS=G(UNPACK)
GEdfPlot	EDF plot of nearest-neighbor function G	PROCESS	G
		PROC SPP	PLOTS=G(UNPACK)
GNppPlot	PP plot of nearest-neighbor function G and CSR	PROCESS	G
		PROC SPP	PLOTS=G(UNPACK)
GPanelPlot	Plot of nearest-neighbor function G	PROCESS	G
		PROC SPP	PLOTS=G(ALL)
IntensityPlot	Surface plot of estimated intensity	Default	Default
JCIPlot	CI plot of function J	PROCESS	J
		PROC SPP	PLOTS=J(UNPACK)
JCombinedPlot	Plot of function J	PROCESS	J
		PROC SPP	PLOTS=J(ALL)
JEdfPlot	EDF plot of function J	PROCESS	J
		PROC SPP	PLOTS=J(UNPACK)
KCIPlot	CI plot of Ripley's function K	PROCESS	K
		PROC SPP	PLOTS=K(UNPACK)
KDiffPlot	Difference plot of Ripley's function K and CSR	PROCESS	K
		PROC SPP	PLOTS=K(UNPACK)
KEdfPlot	EDF plot of Ripley's function K	PROCESS	K
		PROC SPP	PLOTS=K(UNPACK)
KNppPlot	PP plot of Ripley's function K and CSR	PROCESS	K
		PROC SPP	PLOTS=K(UNPACK)
KPanelPlot	Plot of Ripley's function K	PROCESS	K
		PROC SPP	PLOTS=K(ALL)

Table 111.9 *continued*

ODS Graph Name	Plot Description	Statement	Option
LCIPlot	CI plot of Besag's L function	PROCESS	L
		PROC SPP	PLOTS=L(UNPACK)
LDiffPlot	Difference plot of Besag's L function and CSR	PROCESS	L
		PROC SPP	PLOTS=L(UNPACK)
LEdfPlot	EDF plot of Besag's L function	PROCESS	L
		PROC SPP	PLOTS=L(UNPACK)
LNppPlot	PP plot of Besag's L function	PROCESS	L
		PROC SPP	PLOTS=L(UNPACK)
LPanelPlot	Plot of Besag's L function	PROCESS	L
		PROC SPP	PLOTS=L(ALL)
LurkingPanel	Cumulative residual and lurking variable panel plot	MODEL	
		PROC SPP	PLOTS=LURKING(ALL)
LurkingVariable	Cumulative residual and lurking variable plot	MODEL	
		PROC SPP	PLOTS=LURKING(UNPACK)
ObservationsPlot	Scatter plot of observed events, marked events, or covariate variables	PROC SPP	PLOTS=OBSERV
PCFCIPlot	CI plot of the pair correlation function, g	PROCESS	PCF
		PROC SPP	PLOTS=PCF(UNPACK)
PCFCombinedPlot	Plot of the pair correlation function, g	PROCESS	PCF
		PROC SPP	PLOTS=PCF(ALL)
PCFEdfPlot	EDF plot of the pair correlation function, g	PROCESS	PCF
		PROC SPP	PLOTS=PCF(UNPACK)
ProbabilityPlot	Surface plot of occurrence probability under a poisson model	Default	Default
RawResidual	Raw residual plot	MODEL	RESIDUAL
		PROC SPP	PLOTS=RESIDUAL(UNPACK)
ResidualPanel	Raw residual panel plot	MODEL	RESIDUAL
		PROC SPP	PLOTS=RESIDUAL
ResidualScatter	Cumulative residual plot	MODEL	RESIDUAL
		PROC SPP	PLOTS=RESIDUAL(UNPACK)
SmoothedResidual	Cumulative residual plot	MODEL	RESIDUAL
		PROC SPP	PLOTS=RESIDUAL(UNPACK)
TrendCovariatePlot	Trend covariate plot	TREND	PLOTS=CSRKSTEST

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the statements indicated in Table 111.9. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Examples: SPP Procedure

Example 111.1: Exploration of a Multitype Point Pattern

This example demonstrates how you can use PROC SPP to explore a multitype point pattern. Consider the following data set, which consists of locations of retinal amacrine cells in a rabbit's eye. The data set contains three variables: X and Y are the coordinates of the cell locations, and Type is the type of each cell (which is based on whether it turns *on* or *off* when exposed to light). The data were originally analyzed by Diggle, Eglen, and Troy (2006) and Hughes (1985).

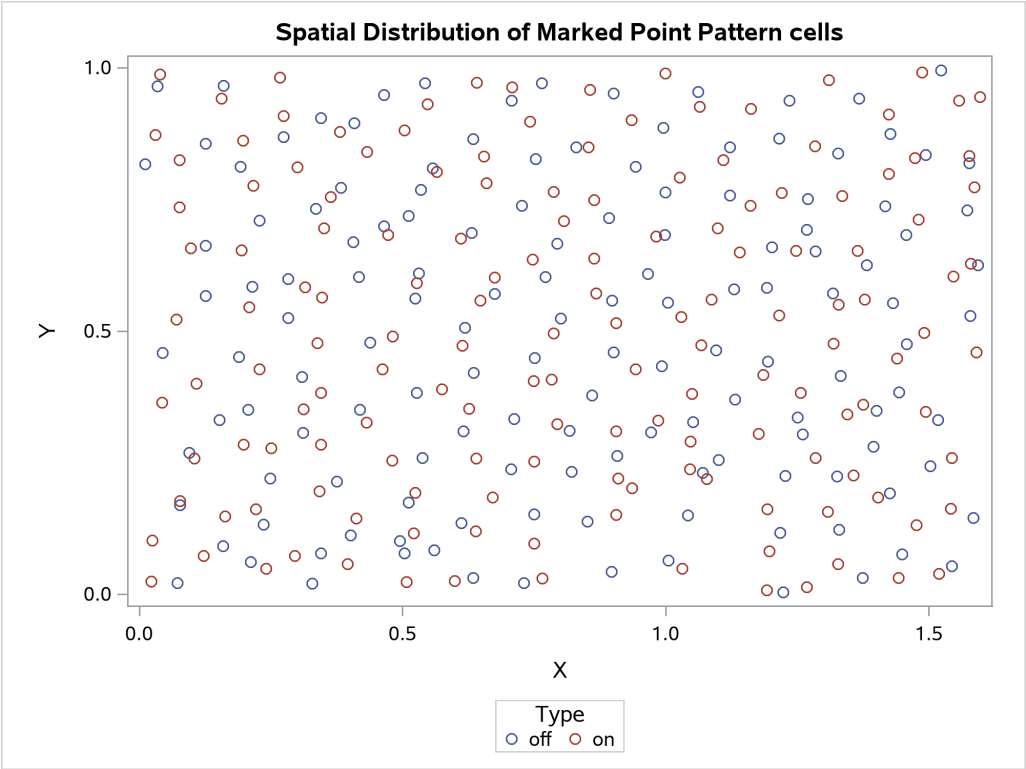
```
data amacrine;
   input X Y Type $ @@;
   label Type='Cell Type';
   datalines;
0.0224   0.0243   on
0.0243   0.1028   on
0.1626   0.1477   on
0.1215   0.0729   on
0.2411   0.0486   on
0.0766   0.1776   on
0.1047   0.2579   on

   ... more lines ...

1.5729   0.729    off
1.457    0.6822   off
1.4168   0.7374   off
;
```

Accounting for mark types enables you to study possible interaction across types. You can study such interactions by using the cross-type variants of distance functions. The following statements compute the cross-K-function between the types of amacrine cells:

```
proc spp data=amacrine edgecorr=on seed=1
         plots(equate)= (observations(attr=mark) K);
   process cells= (X,Y / area=(0,0,1.6,1) mark=Type)
                  / K cross=types('on' 'off');
run;
```

Output 111.1.1 shows two types of cells that are characterized by their state as *on* (in red) and *off* (in blue).

Output 111.1.1 Amacrine Cell Types



Output 111.1.2 lists exploratory information for the amacrine point pattern, and identifies it as a marked point pattern.

Output 111.1.2 Exploratory Information for the Amacrine Marked Point Pattern

Summary of Point Pattern	
Data Type	Marked Point Pattern
Pattern Name	cells
Region Type	User Defined Window
Region X Range	[0,1.6] Units
Region Y Range	[0,1] Units
Region X Size	1.6 Units
Region Y Size	1 Unit
Region Area	1.6 Square Units
Observations in Window	294
Average Intensity	183.75
Grid Nodes in X	50
Grid Nodes in Y	50
Grid Nodes in Window	2500
Mark	Type
Mark Type	Multitype

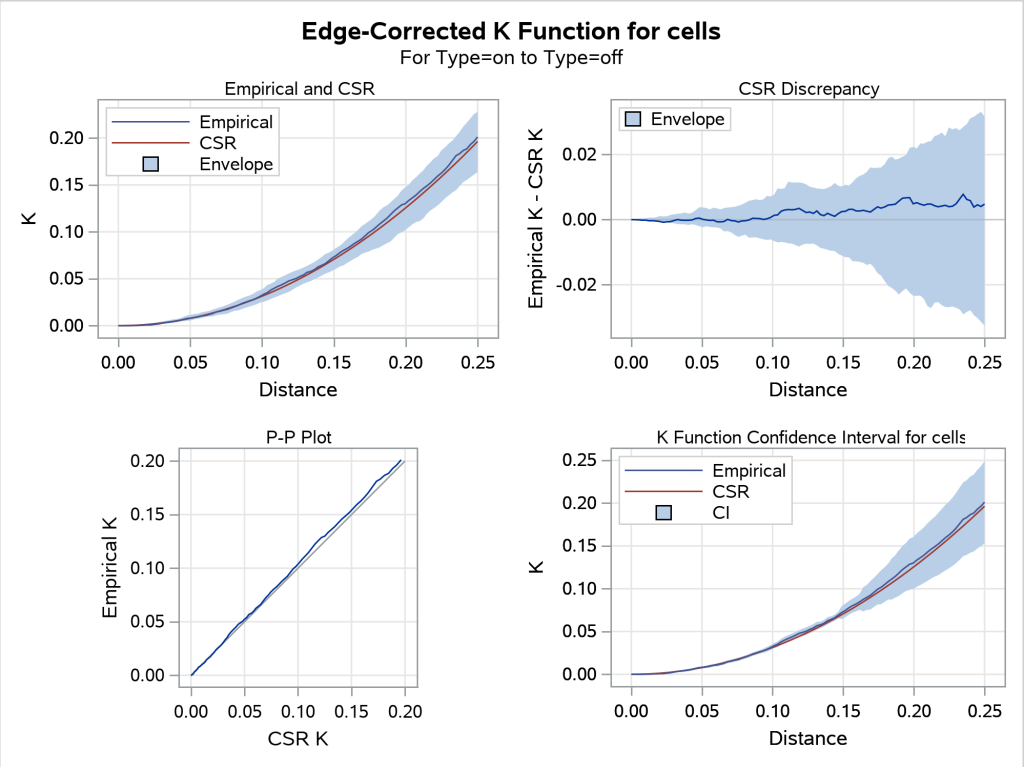
Output 111.1.3 shows the information for each mark type, including the frequency, percentage in the data set, and the first-order intensity, which measures the number of events of that particular mark type per unit of area that is contained in the marked point pattern.

Output 111.1.3 Mark Information for the Amacrine Marked Point Pattern

Level Information for Type			
Value	**Frequency**	**Percentage**	**Intensity**
off	142	48.30%	88.7500
on	152	51.70%	95.0000

Output 111.1.4 shows the cross-K-function test to detect a clustering of points for which Type=*off* around points for which Type=*on* and vice versa. It is very clear from the plots in the top left corner and top right corner that there is no significant difference between the computed cross-K function and the theoretical cross-K function. This clearly indicates that there is no significant clustering of *on* amacrine cells around *off* amacrine cells.

Output 111.1.4 Cross-K Function Panel



Example 111.2: Testing Covariate Dependence of a Point Pattern

In most spatial analysis applications, you are likely to have one or more covariates in addition to the point pattern data set. Hence, you can test for a possible dependency between the observed point pattern and the covariates by using covariate dependency tests that compute empirical distribution function (EDF) statistics. These tests are nonparametric, and the selected EDF statistic indicates whether the covariate values interpolated at the point locations are independent of the transformed covariate and a known model of point pattern intensity such as CSR. Covariate dependency testing serve multiple objectives:

- They tell you whether the empirical distribution of the covariates at the point locations and the empirical distribution function weighted and transformed according to the underlying intensity model (or the predicted distribution) are similar.
- In cases of dissimilarity between the empirical distribution and the predicted distribution, the interpretation is that the covariate gives evidence against the intensity model (CSR in this case).

To request a covariate dependency test that is based on an EDF statistic, you use the COVTEST statement, in which you specify the point process and the covariates that need to be tested. The following statements perform a covariate dependency test that is based on an EDF statistic:

```
proc spp data=sashelp.bei;
   process trees = (x,y /area=(0,0,1000,500) event=Trees);
   trend grad = field(x,y,Gradient);
   trend elev = field(x,y,Elevation);
   covtest trees = grad elev;
run;
```

When you do not request any specific EDF test statistic, by default the SPP procedure produces an EDF test that uses the Kolmogorov-Smirnov statistic for each covariate that is specified in the COVTEST statement, as shown in Output 111.2.1.

Output 111.2.1 Weighted EDF Test Statistic

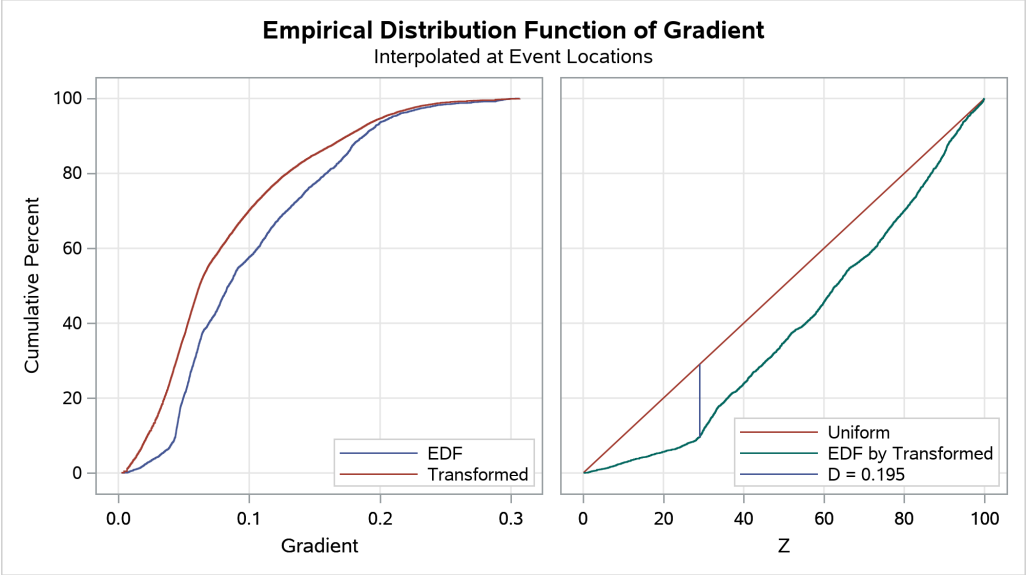
The SPP Procedure

Kolmogorov-Smirnov Weighted EDF Test for Covariate Values

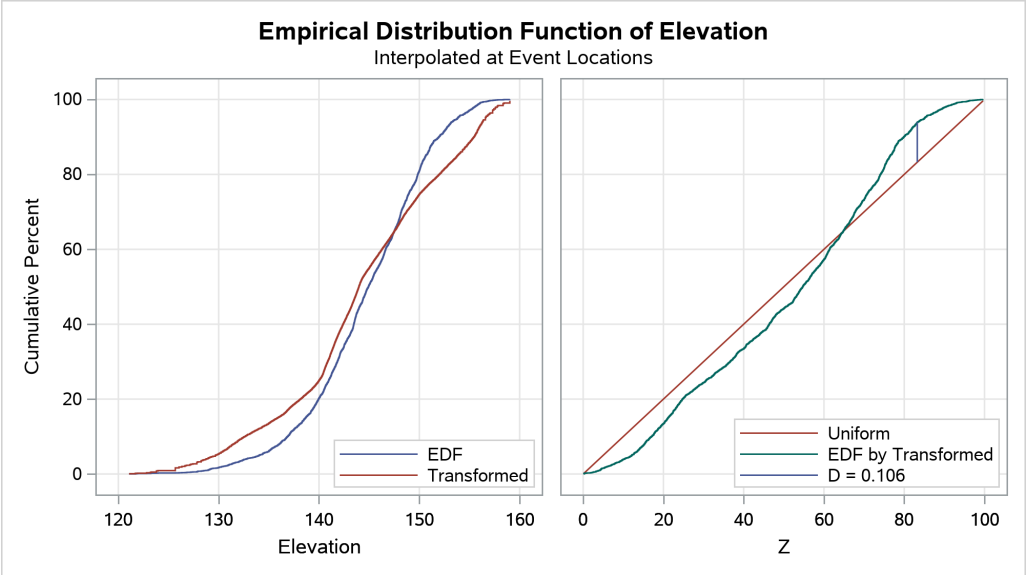
Source	D Statistic	p Value
Gradient	0.194805	<0.0010
Elevation	0.106492	<0.0010

Output 111.2.1 reports the value of the Kolmogorov-Smirnov D test statistic and a p-value. The p-value indicates that the null hypothesis of independence on the covariate is rejected. In addition to the test statistic and the p-value, PROC SPP produces a plot of the empirical and transformed distribution function for each covariate. Output 111.2.2 and Output 111.2.3 show the plots of the Kolmogorov-Smirnov statistic for the covariates Gradient and Elevation, respectively. In each figure, the plot on the left includes the empirical density function (EDF) of the variable (solid blue line) and the weighted EDF (transformed EDF or transformed line). The plot on the right is a PP plot that plots the Empirical probability (EDF) versus the Transformed probability(Transformed) against the reference standard uniform (red line). The same plot also shows the largest vertical difference between the normal and uniform lines, which is the Kolmogorov-Smirnov statistic D. From the right plots in Output 111.2.2 and Output 111.2.3, it is quite apparent that the EDF by Transformed line overlaps only at the ends for the Gradient covariate and crosses the uniform line once for the Elevation covariate. Thus, you can infer that the Gradient covariate (in addition to having a higher D statistic value) deviates considerably from the uniform line.

Output 111.2.2 Kolmogorov-Smirnov EDF Test Plot for Gradient



Output 111.2.3 Kolmogorov-Smirnov EDF Test Plot for Elevation



Example 111.3: Intensity Model Validation Diagnostics

Model validation diagnostics help you evaluate whether the fitted model that involves the covariates is appropriate for the specified point pattern. The SPP procedure provides two types of diagnostics:

- goodness-of-fit test
- residual diagnostics

This example demonstrates the usage of these diagnostics for model validation. It uses a simulated point pattern data set and also simulates two covariates over a 50×50 grid. The following statements define functions for simulating a spatial point pattern given an intensity function by the method of random thinning of Lewis and Shedler (1979), as discussed in Schabenberger and Gotway (2005) and Wicklin (2013). For more information about the method and the code, see Wicklin (2013). The functions are saved in a SAS/IML storage catalog to make them available for reuse.

```
ods graphics on;

proc iml;
   start Uniform2d(n, a, b);
      u = j(n, 2);
      call randgen(u, "Uniform");
      return( u # (a||b) );
   finish;

   start HomogPoissonProcess(lambda, a, b);
      n = 1;
      call randgen(n,"Poisson", lambda*2500);
      return( Uniform2d(n, a, b) );
   finish;

   start InhomogPoissonProcess(a, b) global(lambda0);
      u = HomogPoissonProcess(lambda0, a, b);
      lambda = Intensity(u[,1], u[,2]);
      r = shape(.,sum(lambda<=lambda0),1);
      call randgen(r,"Bernoulli", lambda[loc(lambda<=lambda0)]/lambda0);
      return( u[loc(r),] );
   finish;

   reset storage=sasuser.SPPThin;
   store module=Uniform2d
         module=HomogPoissonProcess
         module=InhomogPoissonProcess;
quit;
```

The following statements define a certain intensity function that is based on the elevation and slope of the land around particular hills, with hills characterized by a four-column matrix Hills, where the first two columns give the X and Y coordinates of each hill center and the last two columns give their height and radii. In the model, both elevation and slope are assumed to be positive, with a negative effect on intensity, so lambda0 (the maximum value of intensity) is the value at Elevation=Slope=0.

```
proc iml;
   %let xH = Hills[iHill,1];
   %let yH = Hills[iHill,2];
   %let hH = Hills[iHill,3];
   %let rH = Hills[iHill,4];

   start Elevation(x,y) global(Hills);
      Elevation = 0;
      do iHill = 1 to nrow(Hills);
         Height = &hH*exp(-((x - &xH)##2 + (y - &yH)##2)/&rH);
         Elevation = Elevation + Height;
```

```
      end;
      return(Elevation);
   finish;

   start Slope(x,y) global(Hills);
      xslope = 0;
      yslope = 0;
      do iHill = 1 to nrow(Hills);
         Height = &hH*exp(-((x - &xH)##2 + (y - &yH)##2)/&rH);
         dxHeight = -2*Height#(x - &xH)/&rH;
         dyHeight = -2*Height#(y - &yH)/&rH;
         xslope   = xslope + dxHeight;
         yslope   = yslope + dyHeight;
      end;
      Slope = sqrt(xslope##2 + yslope##2);
      return(Slope);
   finish;

   start Intensity(x,y) global(lambda0);
      lin = 0.5 - 2*Elevation(x,y) - 10*Slope(x,y);
      return(exp(lin));
   finish;

   lambda0 = exp(0.5);

   reset storage=sasuser.SPPFlowers;
   store lambda0 module=Elevation module=Slope module=Intensity;
quit;
```

Finally, the following statements use the simulation method of Wicklin (2013) and the previously defined intensity to simulate a spatial point pattern on 10 hills in an area of 50×50 units. The covariates, Elevation and Slope, are also computed over a grid of points in the region of interest.

```
proc iml;
   reset storage=sasuser.SPPThin;
   load module=Uniform2d
        module=HomogPoissonProcess
        module=InhomogPoissonProcess;

   reset storage=sasuser.SPPFlowers;
   load  module=Elevation module=Slope module=Intensity lambda0;

   a = 50;
   b = 50;

   Hills = { 9.2 48.5 0.2 13.0,
           46.1 48.5 0.3 26.6,
            2.5  3.3 0.7 26.2,
           42.7  3.4 0.9 14.9,
           13.6 34.5 1.0 11.3,
           34.4 20.6 0.3 14.4,
           23.8 42.2 0.4 29.5,
           29.1 18.9 0.5 25.3,
           46.6 46.5 0.3 14.9,
           19.6 23.6 0.5  8.4};
```

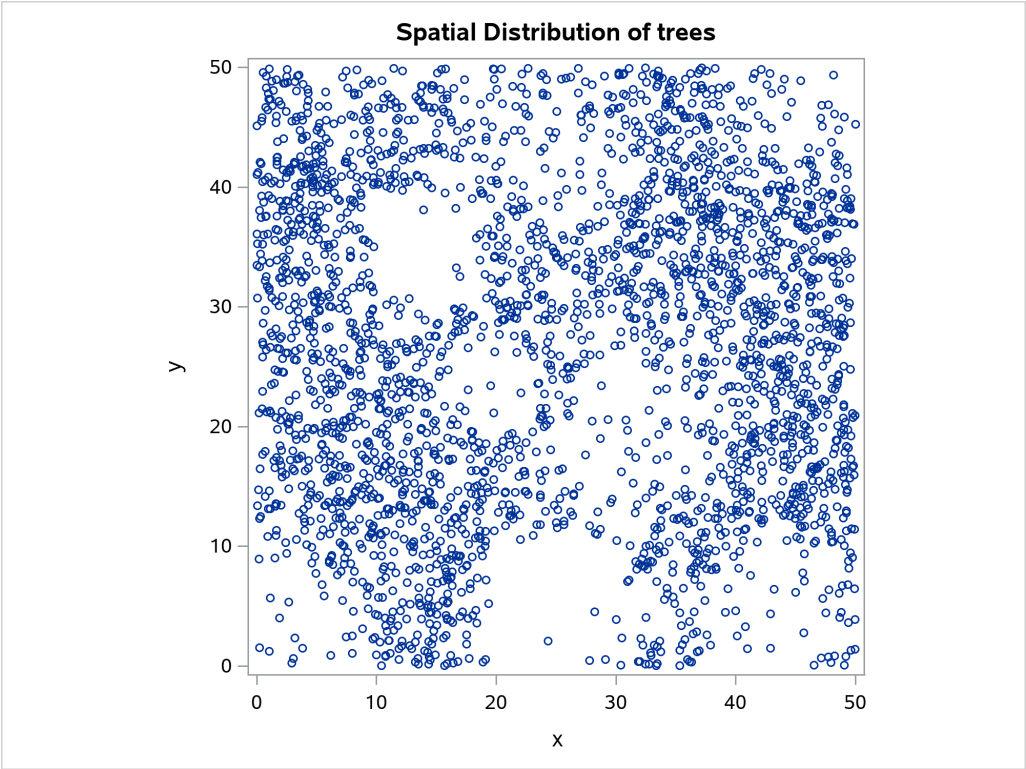
```
   call randseed(12345);

   free Cov;
   Cov = j((a+1) * (b+1), 5, 0);
   do x = 0 to a; do y = 0 to b;
      Cov[x#(a+1)+y+1,] =  (x || y || Elevation(x,y) || Slope(x,y)
                           || Intensity(x,y));
   end; end;

   create Covariates var {"x" "y" "Elevation" "Slope" "Intensity"};
   append from Cov;
   close Covariates;

   Hills = Hills // {25 5 2 15};
   z = InhomogPoissonProcess(a, b);

   create Events var {"x" "y"};
   append from z;
   close;

quit;

data simAll;
   set Events(in=e) Covariates;
   Flowers = e;
run;
```

The point pattern data set and the covariate data set are combined in the simAll data set and the event observations can be identified by using a variable Flowers.

```
proc spp data=simAll plots(equate)=(trends observations);
   process trees = (x, y /area=(0,0,50,50) Event=Flowers);
   trend grad = field(x,y, Elevation);
   trend elev = field(x,y, Slope);
run;
```

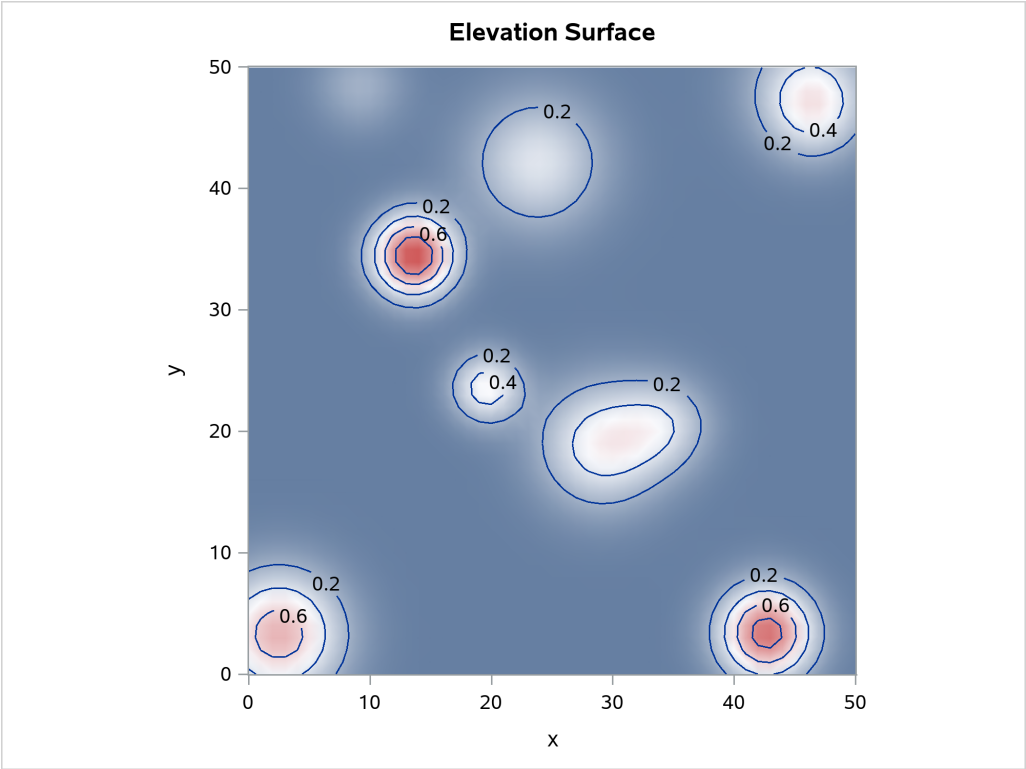
Output 111.3.1 shows the point pattern. The point pattern has been simulated to include a Gaussian bump at the center of the study region.

Output 111.3.1 Spatial Point Pattern of Simulated Flowers

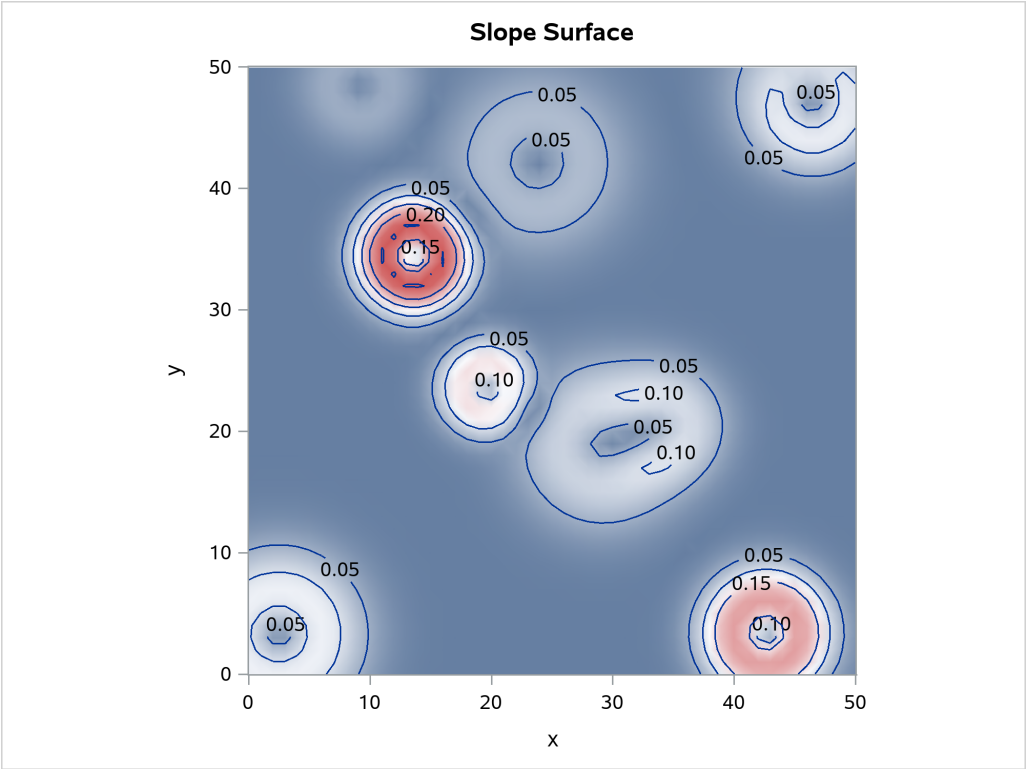


Output 111.3.2 shows the spatial covariate Elevation, and Output 111.3.3 shows the spatial covariate Slope. The covariates have been simulated to include several Gaussian hills, and they are continuous within the 50×50 study region (that is, every point in the region has a value for these covariates).

Output 111.3.2 Spatial Covariate Elevation



Output 111.3.3 Spatial Covariate Slope

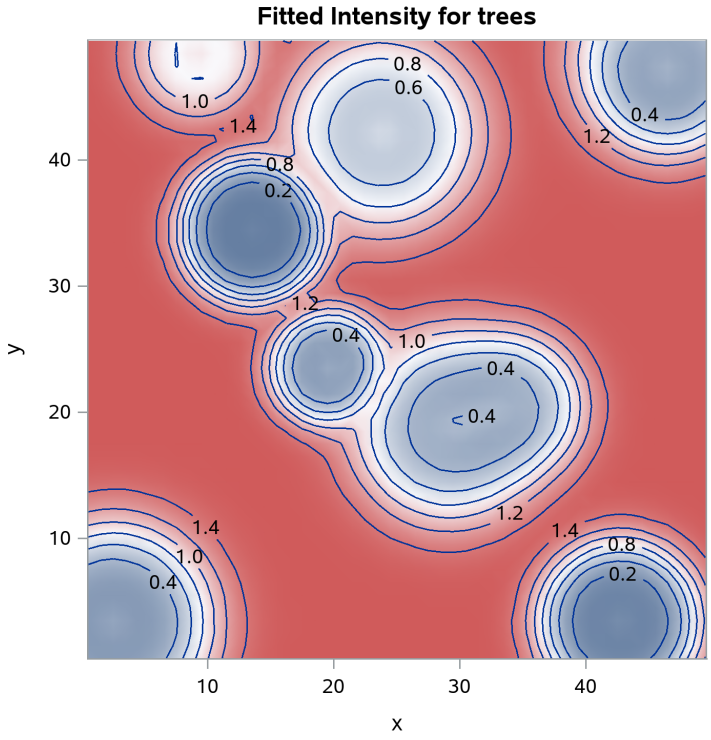


The following code fits an intensity model for the simulated point pattern that involves the simulated covariates Elevation and Slope. It also requests model validation diagnostics, including residuals and the goodness-of-fit test.

```
proc spp data=simAll seed=1  plots(equate)=(residual);
   process trees = (x,y /area=(0,0,50,50) Event=Flowers) /quadrat(4,2 /details);
   trend elev = field(x,y, Elevation);
   trend slope = field(x,y, Slope);
   model trees = elev slope/residual(b=5) gof;
run;
```

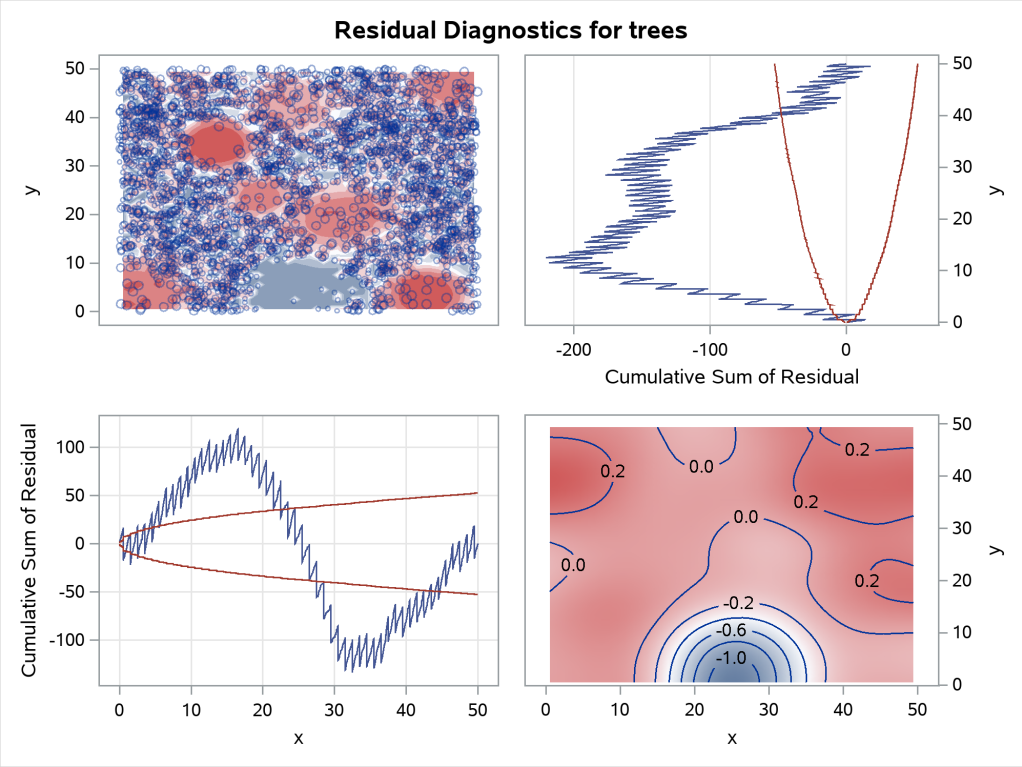
Output 111.3.4 shows the fitted intensity estimate that is based on the model that involves just the covariates Elevation and Slope.

Output 111.3.4 Fitted Intensity Estimate for the Simulated Point Pattern



Output 111.3.5 shows the residual diagnostics for the model. It is clear from the smoothed residual plot at the bottom right corner of Output 111.3.5 that the model that involves just the covariates Elevation and Slope fails to account for the Gaussian bump in the middle of the study region. This is revealed by the trend at the center of the smoothed residual plot at the bottom right corner of Output 111.3.5.

Output 111.3.5 Residual Diagnostics for the Fitted Model



Consequently, the goodness-of-fit test rejects the hypothesis that the point pattern was generated by the fitted model. This is evident in the low *p*-value that is obtained for the Pearson chi-square test for goodness of fit, which is shown in Output 111.3.6.

Output 111.3.6 Pearson Chi-Square Test for Goodness of Fit

Pearson Chi-Square Test for Goodness-of-Fit			
DF	Dispersion Index	Chi-Square	Pr > ChiSq
7	12.742	89.19	<.0001

Output 111.3.7 shows the corresponding Pearson residuals for the goodness-of-fit test.

Output 111.3.7 Quadrat Information and Pearson Residuals for Goodness-of-Fit Test

Quadrat Information for Goodness-of-Fit Test					
ID	Quadrat	Expected Frequency	Count	Percentage	Pearson Residual
1	(1,1)	352	375	13.58%	1.20
2	(2,1)	398	310	11.22%	-4.41
3	(3,1)	298	184	6.66%	-6.60
4	(4,1)	323	349	12.64%	1.45
5	(1,2)	383	450	16.29%	3.43
6	(2,2)	261	264	9.56%	0.19
7	(3,2)	383	397	14.37%	0.70
8	(4,2)	371	433	15.68%	3.22

When the model involves only the covariates Elevation and Slope, the residual diagnostics and the goodness-of-fit test both reveal discrepancies in the model that do not fully account for the simulated point pattern. In particular, the model misses the Gaussian bump in the middle of the study region.

References

Baddeley, A. (2007). "Spatial Point Processes and Their Applications." In *Stochastic Geometry*, edited by W. Weil, 1–75. Vol. 1892 of Lecture Notes in Mathematics. Berlin: Springer.

Baddeley, A. (2010). "Modeling Strategies." In *Handbook of Spatial Statistics*, edited by A. E. Gelfand, P. J. Diggle, M. Fuentes, and P. Guttorp, 339–369. Boca Raton, FL: Chapman & Hall/CRC.

Baddeley, A. (2014). Personal communication.

Baddeley, A., and Turner, R. (2000). "Practical Maximum Pseudolikelihood for Spatial Point Patterns." *Australian and New Zealand Journal of Statistics* 42:283–322.

Baddeley, A., and Turner, R. (2005). "Spatstat: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software* 12:1–42.

Baddeley, A., and Turner, R. (2013). "Spatstat: An R Package for Spatial Statistics (ver. 1.31-3)." http://www.spatstat.org/.

Baddeley, A., Turner, R., Moller, J., and Hazelton, M. (2005). "Residual Analysis for Spatial Point Processes." *Journal of the Royal Statistical Society, Series B* 67:616–666.

Baddeley, A. J., Kerscher, M., Schladitz, K., and Scott, B. T. (2000). "Estimating the *J* Function without Edge Correction." *Statistica Neerlandica* 54:315–328.

Banerjee, S. (2005). "On Geodetic Distance Computations in Spatial Modeling." *Biometrics* 61:617–625.

Berman, M., and Turner, R. (1992). "Approximating Point Process Likelihoods with GLIM." *Journal of the Royal Statistical Society, Series C* 41:31–38.

Condit, R. (1998). *Tropical Forest Census Plots: Methods and Results from Barro Colorado Island, Panama, and a Comparison with Other Plots*. Berlin: Springer-Verlag.

Condit, R., Hubbell, S. P., and Foster, R. B. (1996). "Changes in Tree Species Abundance in a Neotropical Forest: Impact of Climate Change." *Journal of Tropical Ecology* 12:231–256.

Cressie, N., and Collins, L. B. (2001). "Analysis of Spatial Point Patterns Using Bundles of Product Density LISA Functions." *Journal of Agricultural, Biological, and Environmental Statistics* 6:118–135.

D'Agostino, R. B., and Stephens, M., eds. (1986). *Goodness-of-Fit Techniques*. New York: Marcel Dekker.

Diggle, P. J. (2003). *Statistical Analysis of Spatial Point Patterns*. New York: Oxford University Press.

Diggle, P. J., Eglen, S. R., and Troy, J. B. (2006). "Modelling the Bivariate Spatial Distribution of Amacrine Cells." In *Case Studies in Spatial Point Process Modeling*, edited by A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan, 215–233. New York: Springer.

Diggle, P. J., Rowlingson, B., and Su, T.-L. (2005). "Point Process Methodology for On-Line Spatio-temporal Disease Surveillance." *Environmetrics* 16:423–434.

Fiksel, T. (1988). "Edge-Corrected Density Estimators for Point Processes." *Statistics* 19:67–75.

Hubbell, S. P., and Foster, R. B. (1983). "Diversity of Canopy Trees in a Neotropical Forest and Implications for the Conservation of Tropical Trees." In *Tropical Rain Forest: Ecology and Management*, edited by S. J. Sutton, T. C. Whitmore, and A. C. Chadwick, 25–41. Oxford: Blackwell.

Hughes, A. (1985). "New Perspectives in Retinal Organisation." *Progress in Retinal Research* 4:243–314.

Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. Hoboken, NJ: John Wiley & Sons.

Lewis, P. A. W., and Shedler, G. S. (1979). "Simulation of Nonhomogeneous Poisson Processes by Thinning." *Naval Research Logistics Quarterly* 26:403–413.

Ripley, B. D. (1988). *Statistical Inference for Spatial Processes*. Cambridge: Cambridge University Press.

Ripley, B. D., and Rasson, J.-P. (1977). "Finding the Edge of a Poisson Forest." *Journal of Applied Probability* 14:483–491.

Schabenberger, O., and Gotway, C. A. (2005). *Statistical Methods for Spatial Data Analysis*. Boca Raton, FL: Chapman & Hall/CRC.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. New York: Chapman & Hall.

Stoyan, D. (1987). "Statistical Analysis of Spatial Point Processes: A Soft-Core Model and Cross-Correlations of Marks." *Biometrical Journal* 29:971–980.

Stoyan, D., and Stoyan, H. (1994). *Fractals, Random Shapes, and Point Fields: Methods of Geometrical Statistics*. Chichester, UK: John Wiley & Sons.

Wicklin, R. (2013). *Simulating Data with SAS*. Cary, NC: SAS Institute Inc.

Subject Index

Syntax Index