APPENDICES

NON-ERGODIC PROBABILISTIC SEISMIC HAZARD METHODOLOGY USING PHYSICS-BASED GROUND MOTION PREDICTION THE CASE OF L'AQUILA, ITALY

JEDIDIAH JOEL AGUIRRE[1,2], BRUNO RUBINO[1], MAURIZIO VASSALLO[3], GIUSEPPE DI GIULIO[3] AND FRANCESCO VISINI[4]

[1]Università Degli Studi Dell'Aquila 67100 L'Aquila, Italy
[2]University of the Philippines 4031 Los Baños, Laguna, Philippines
[3]Istituto Nazionale di Geofisica e Vulcanologia – Sezione dell'Aquila 67100 L'Aquila, Italy
[4]Istituto Nazionale di Geofisica e Vulcanologia – Sezione di Pisa 56125 Pisa, Italy
(received: 4 January 2020; revised: 9 February 2020; accepted: 15 March 2020; published online: 1 April 2020)
DOI: https://doi.org/10.34808/tq2020/24.2/a_appendices

Appendices

Appendix A. Derivation of Seismic Hazard Rate

In the PSHA Methodology, we can treat y as a reference ground motion parameter, say the PGA, which depends mainly on distance R M, which are assumed to be independent of each other. Since magnitude occurrences are treated as rupture scenarios, the conditional exceedance probability in (98) is treated as the relative proportion of distances whose PGA exceeds that of the reference PGA, given the magnitude of the occurrence. Hence, given the uncertainty in the magnitude, the probability of exceedance for a single seismic source is given by

$$P(Y > y \cap M) = \int P[Y > y|M]\, f_M(m)\, dm, \tag{98}$$

where

$$P[Y > y|M] = \frac{N\left[R_{Y>y}\right]}{N_R}$$

For brevity, we rewrite (98) as

$$P(Y > y) = \int P[Y > y|M]\, f_M(m)\, dm. \tag{99}$$

Given N seismic sources (assuming) independent of each other and collectively exhaustive, we apply the Total Probability Theorem for the entire study area:

$$P(Y > y) = \sum_{i=1}^{N} \int P[Y > y|M]_i\, f_{M_i}(m)\, dm, \tag{100}$$

Evaluating the integral in (100) numerically, one obtains the probability mass function of the magnitude random variable

$$P(Y > y) = \sum_{i=1}^{N} \sum_{j=1}^{N_M} P[Y > y|M]_i\, P\left(M = m_j\right)_i, \tag{101}$$

where N_M is the number of magnitude occurrences in a seismic source. The product of the activity rate and the probability of occurrence at a given magnitude is the hazard rate of the rupture scenario. For brevity, considering all kinds of rupture scenarios in the study area regardless of the seismic source a rupture scenario due to the assumption in (100), (102) becomes the expression in (6)

$$\lambda_{IM}(im) = \sum_{n=1}^{N_{rup}} P[Y > y|rup_n]\, \lambda_{Rup}(rup_n). \tag{102}$$

Appendix B. Distance Probability Mass Functions

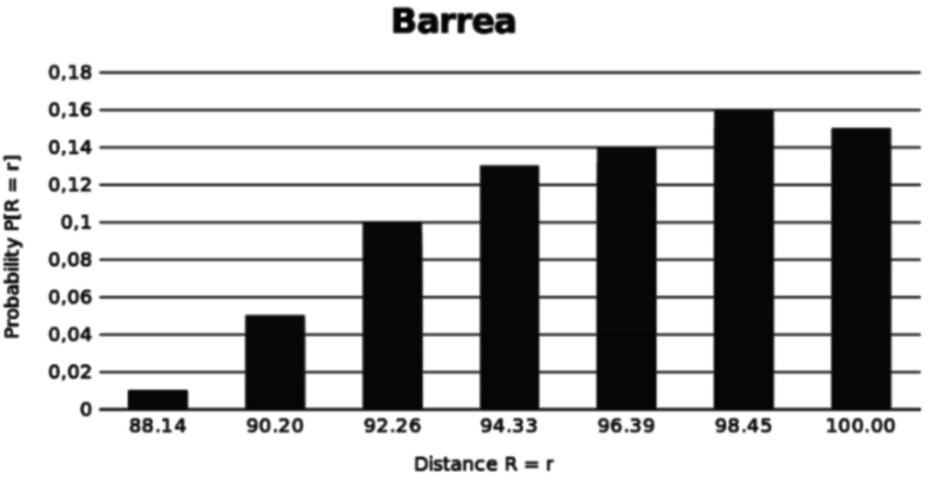

Figure 26. Distance Probability Mass Function for Barrea Fault

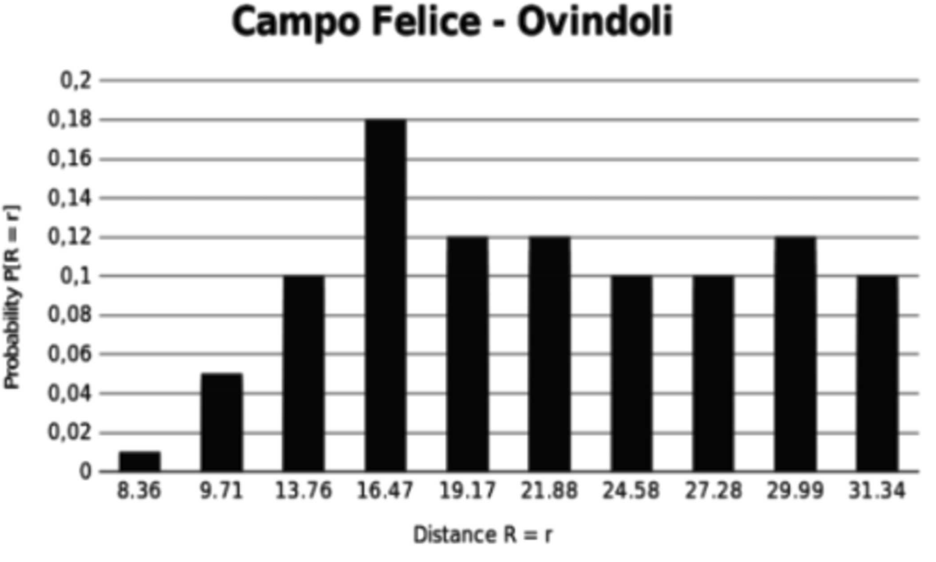

Figure 27. Distance Probability Mass Function for Campo Felice-Ovindoli Fault

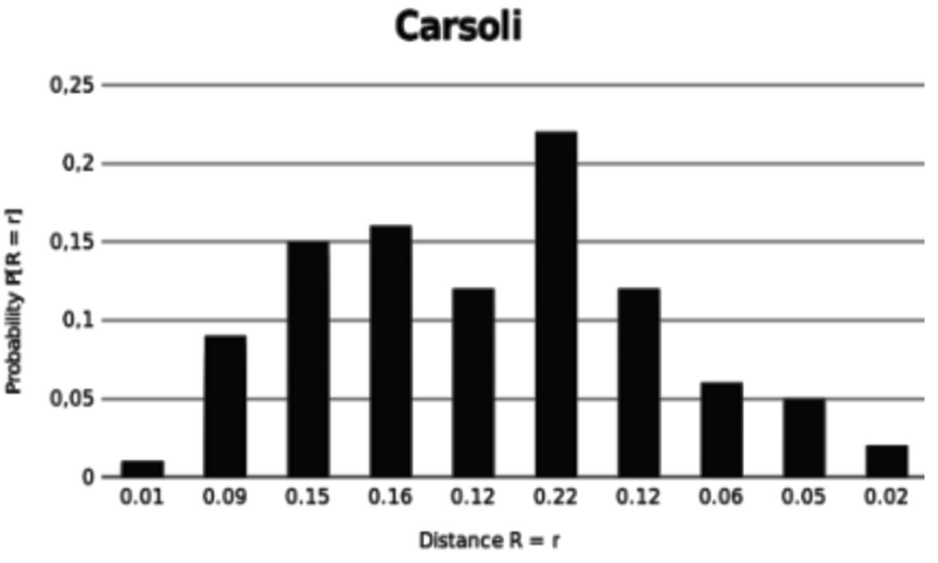

Figure 28. Distance Probability Mass Function for Carsoli Fault

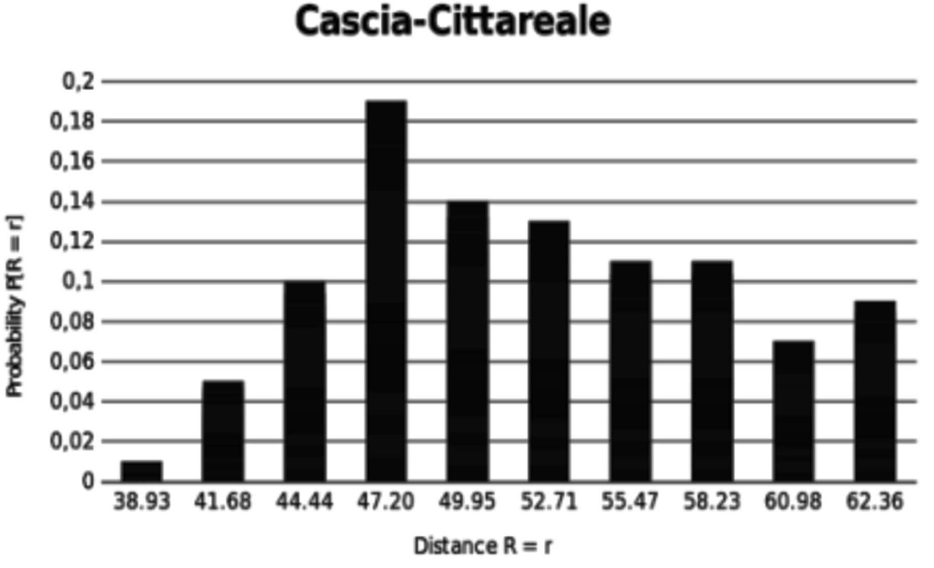

Figure 29. Distance Probability Mass Function for Cascia-Cittareale Fault

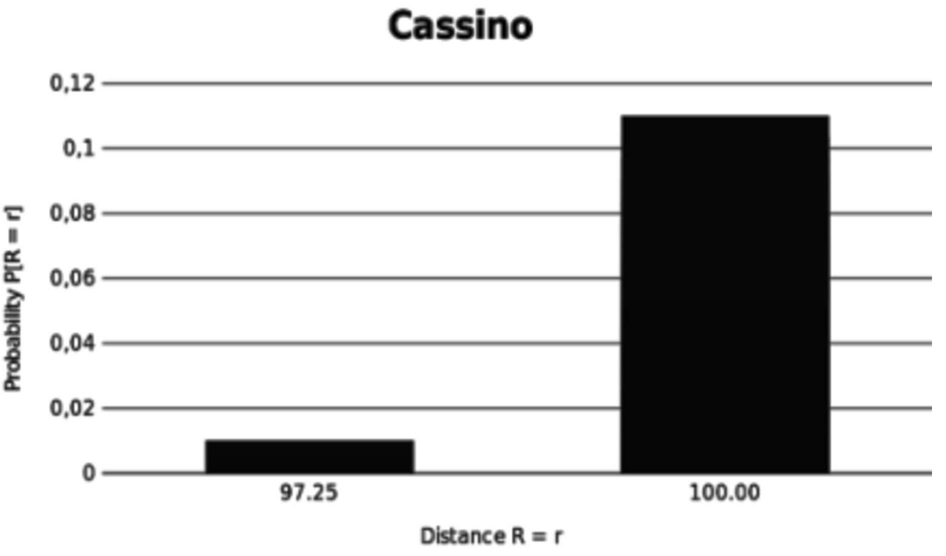

Figure 30. Distance Probability Mass Function for Cassino Fault

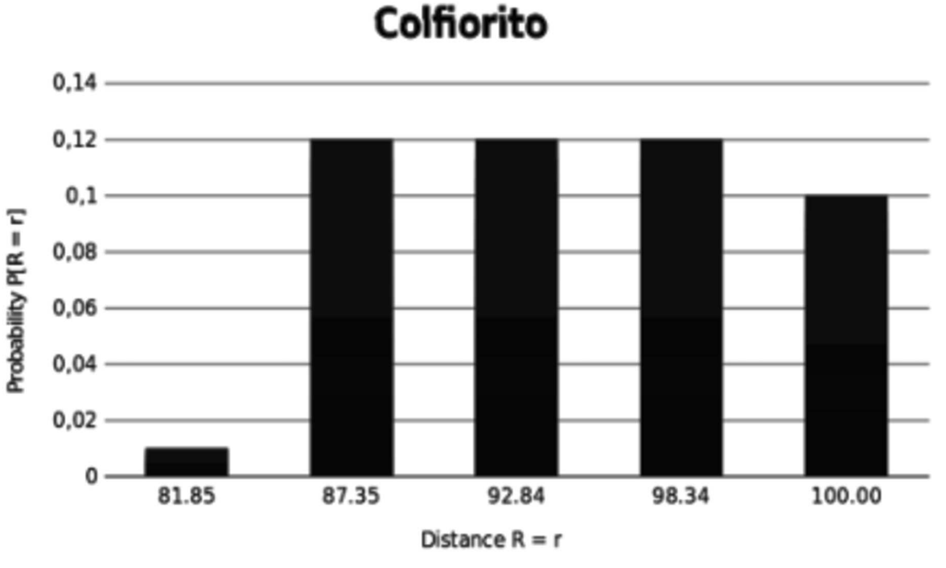

Figure 31. Distance Probability Mass Function for Colfiorito Fault

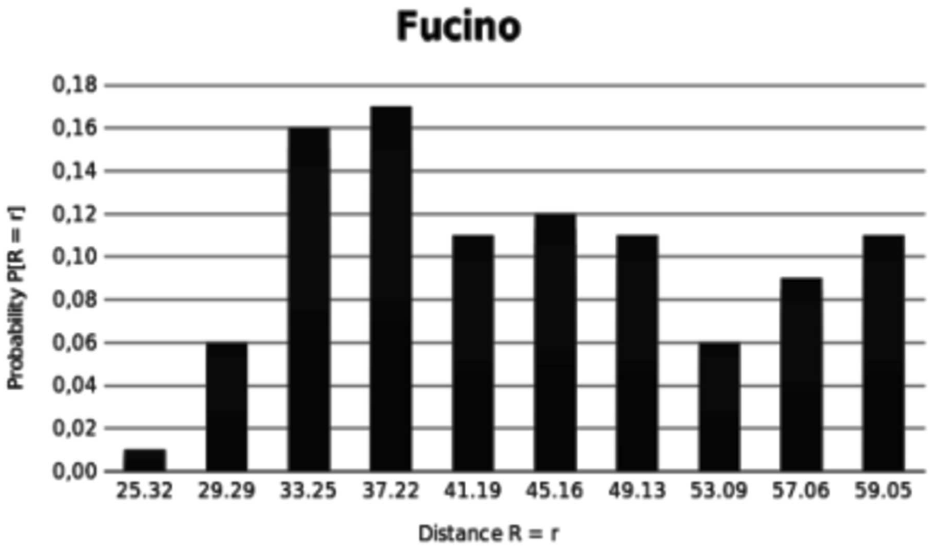

Figure 32. Distance Probability Mass Function for Fucino Fault

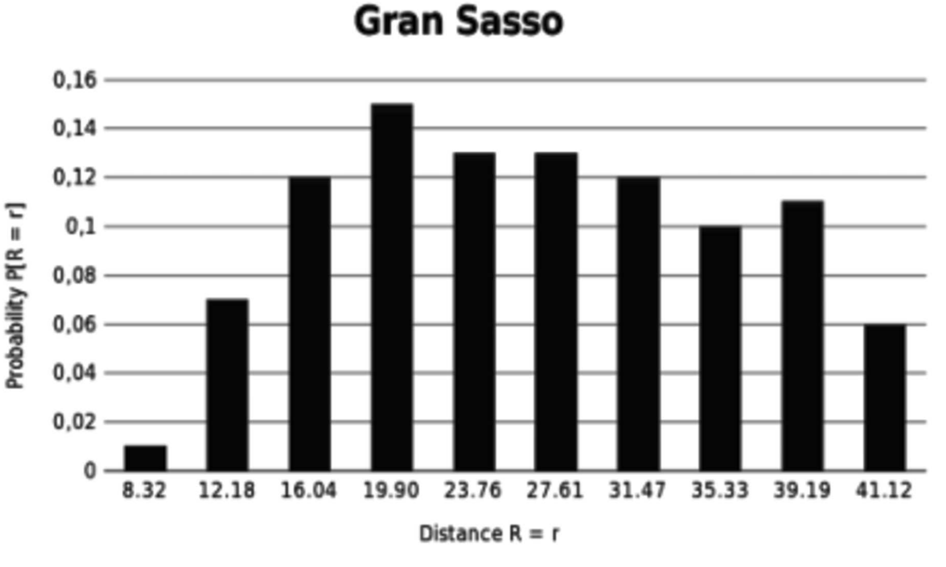

Figure 33. Distance Probability Mass Function for Gran Sasso Fault

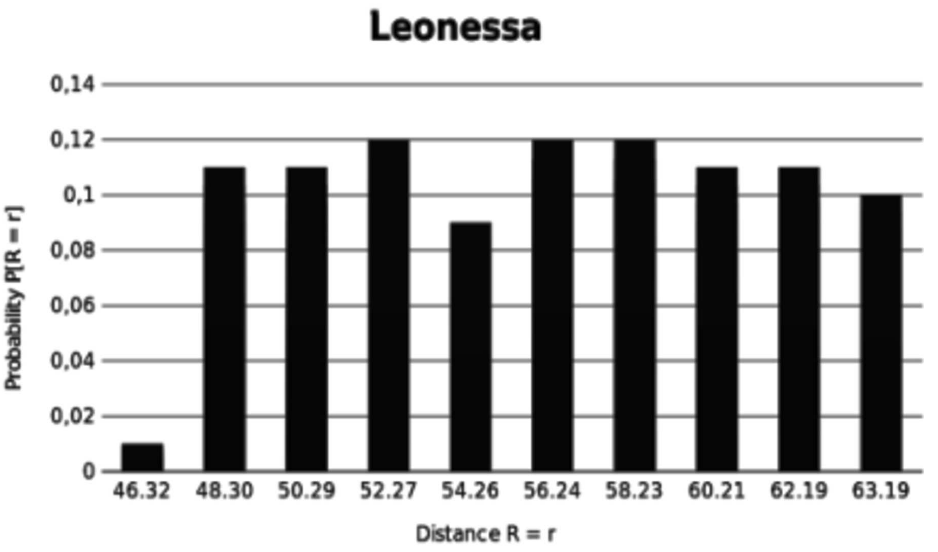

Figure 34. Distance Probability Mass Function for Leonessa Fault

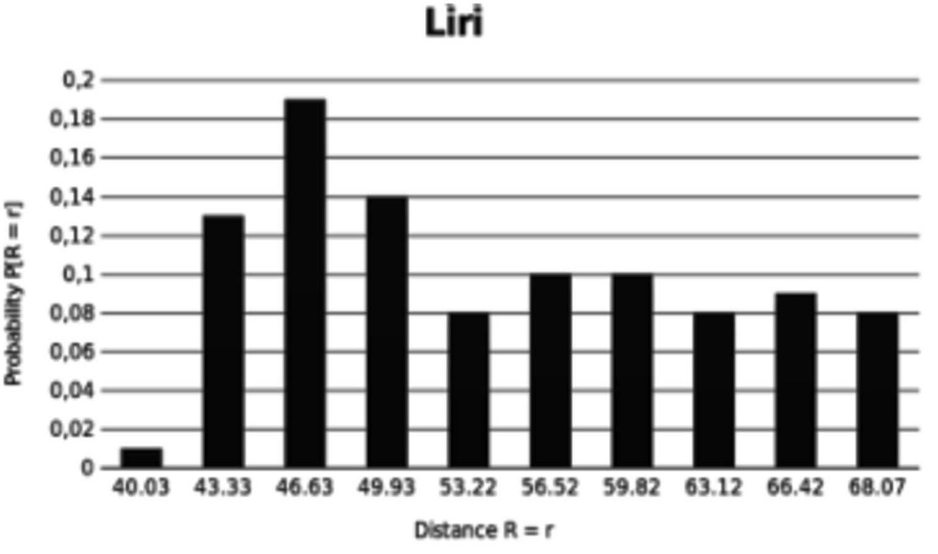

Figure 35. Distance Probability Mass Function for Liri Fault

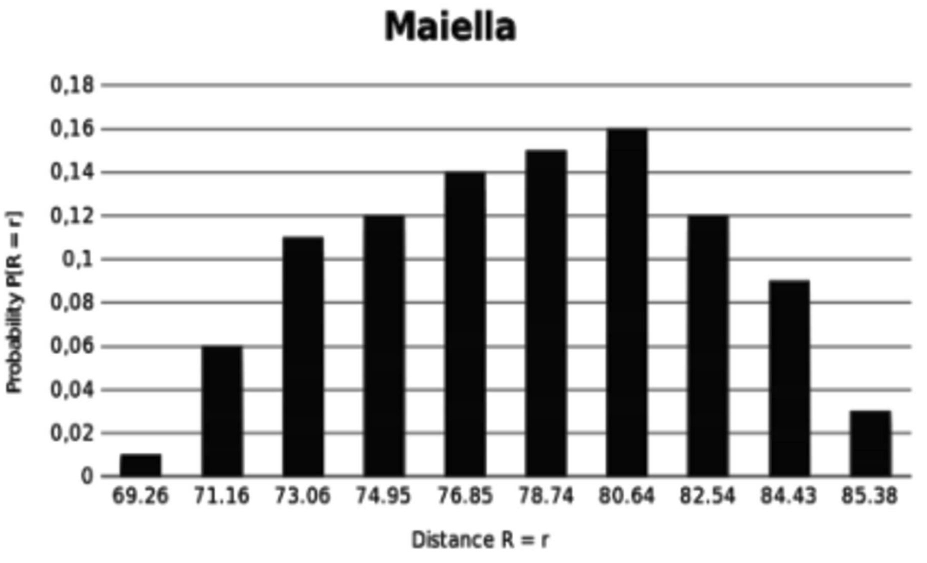

Figure 36. Distance Probability Mass Function for Maiella Fault

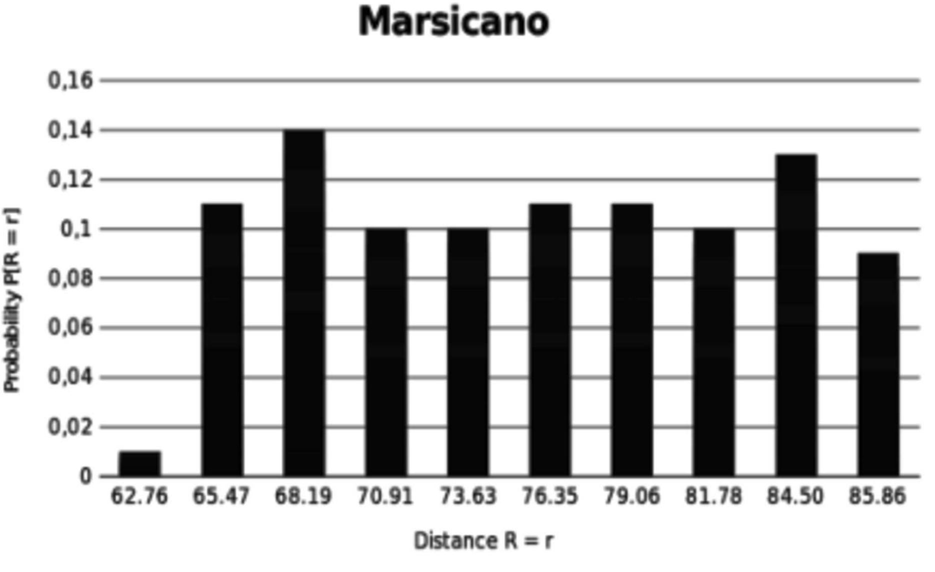

Figure 37. Distance Probability Mass Function for Marsicano Fault

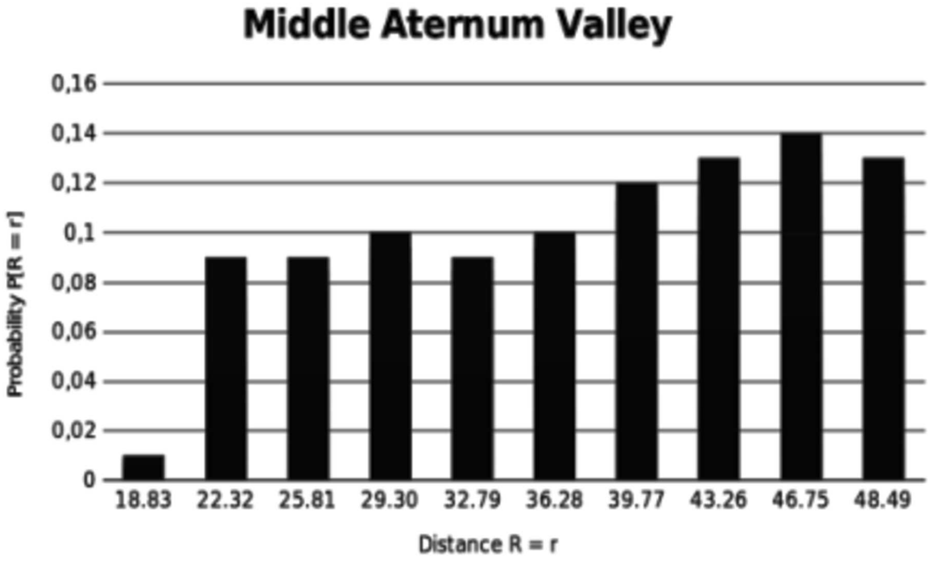

Figure 38. Distance Probability Mass Function for Middle Aternum Valley Fault

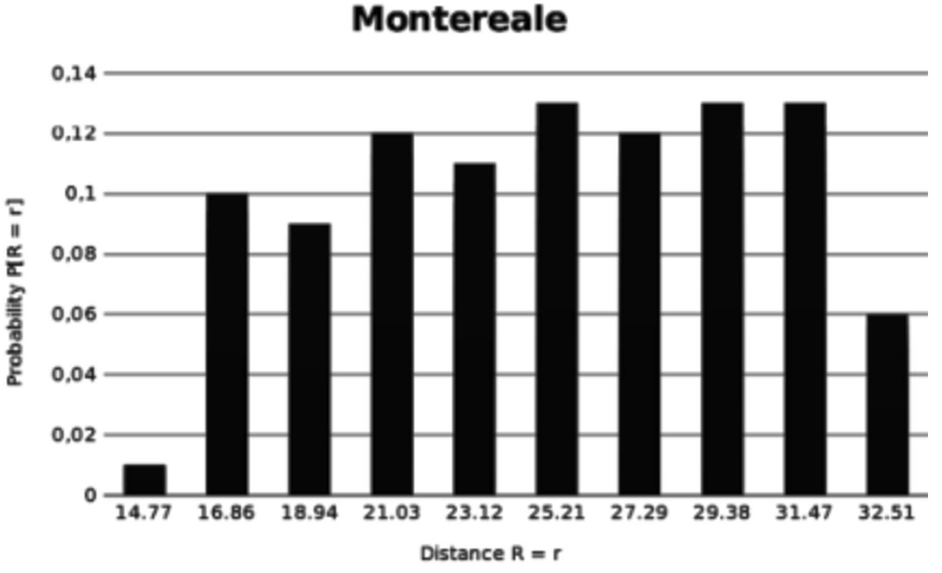

Figure 39. Distance Probability Mass Function for Montereale Fault

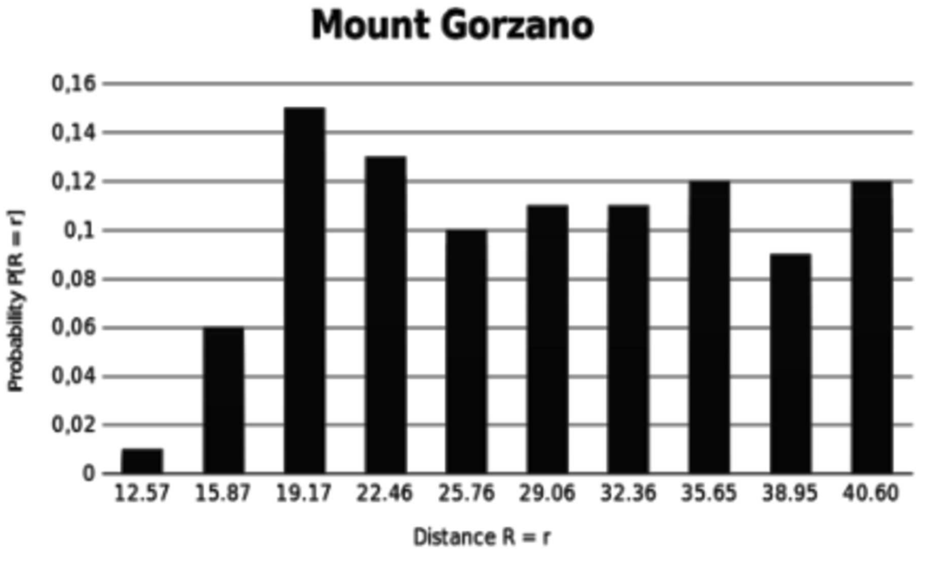

Figure 40. Distance Probability Mass Function for Mount Gorzano Fault

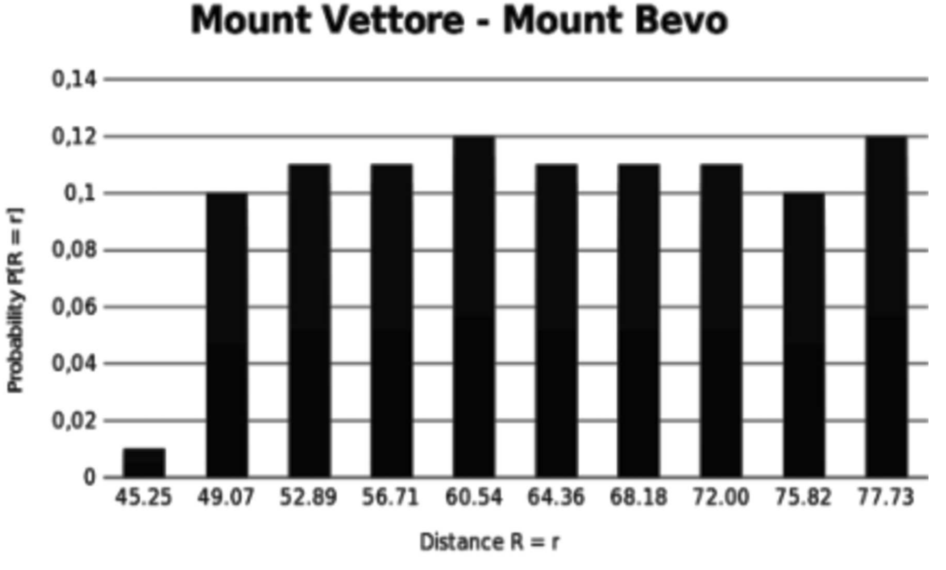

Figure 41. Distance Probability Mass Function for Mount Vettore — Mount Bevo

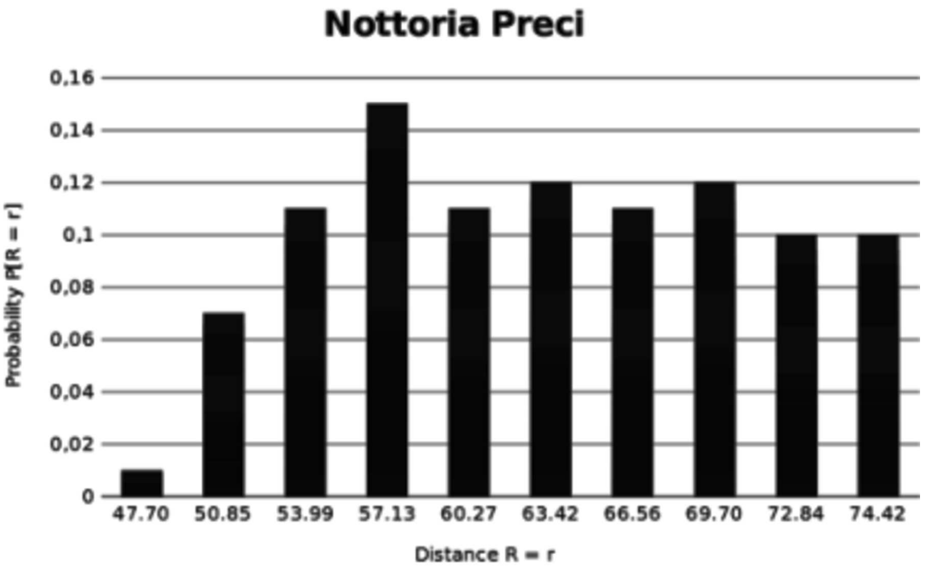

Figure 42. Distance Probability Mass Function for Nottoria Preci Fault

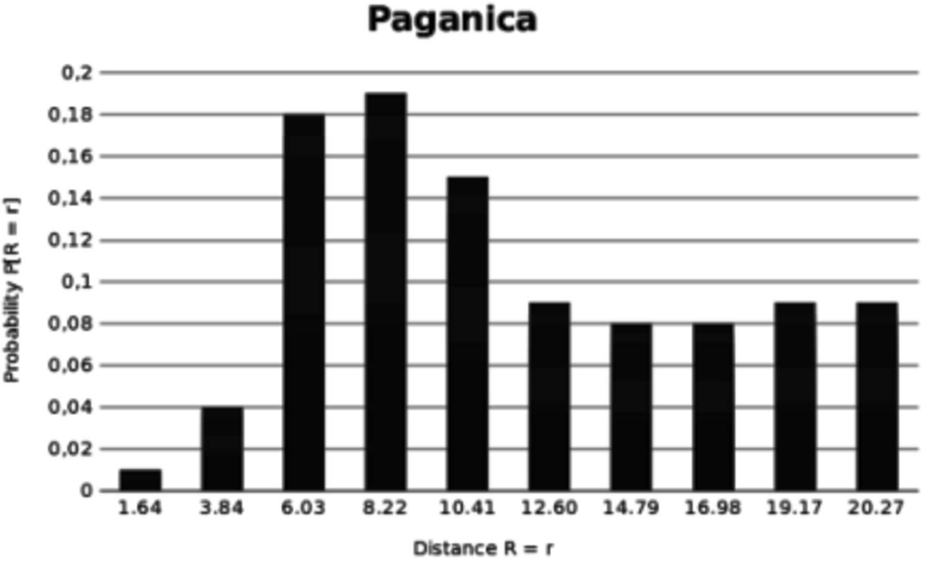

Figure 43. Distance Probability Mass Function for Paganica Fault

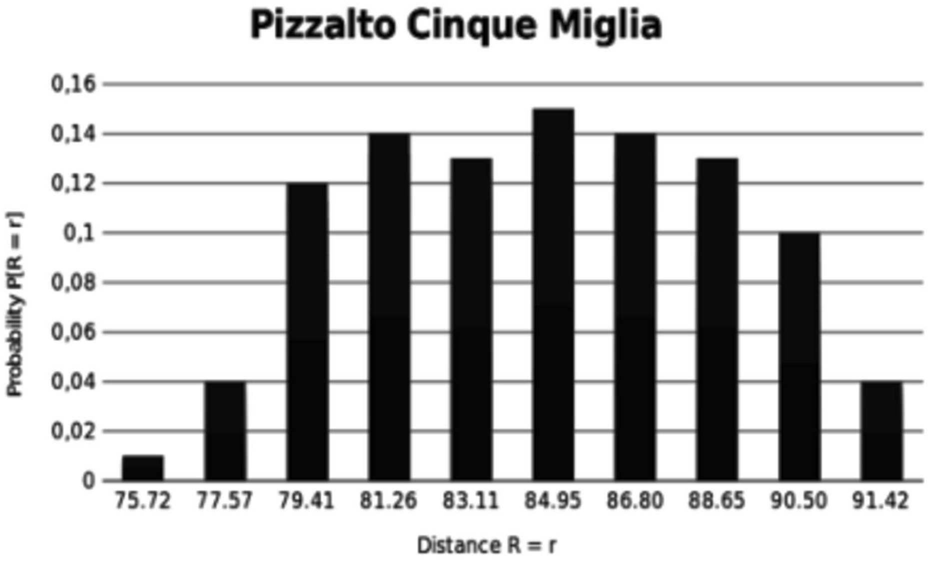

Figure 44. Distance Probability Mass Function for Pizzalto Cinque Miglia Fault

Figure 45. Distance Probability Mass Function for Pizzoli-Pettino Fault

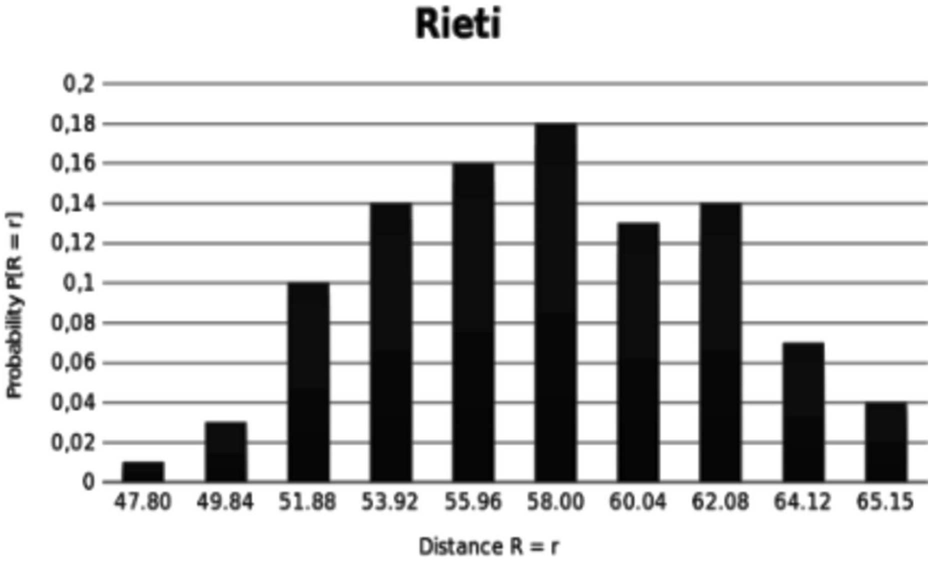

Figure 46. Distance Probability Mass Function for Rieti Fault

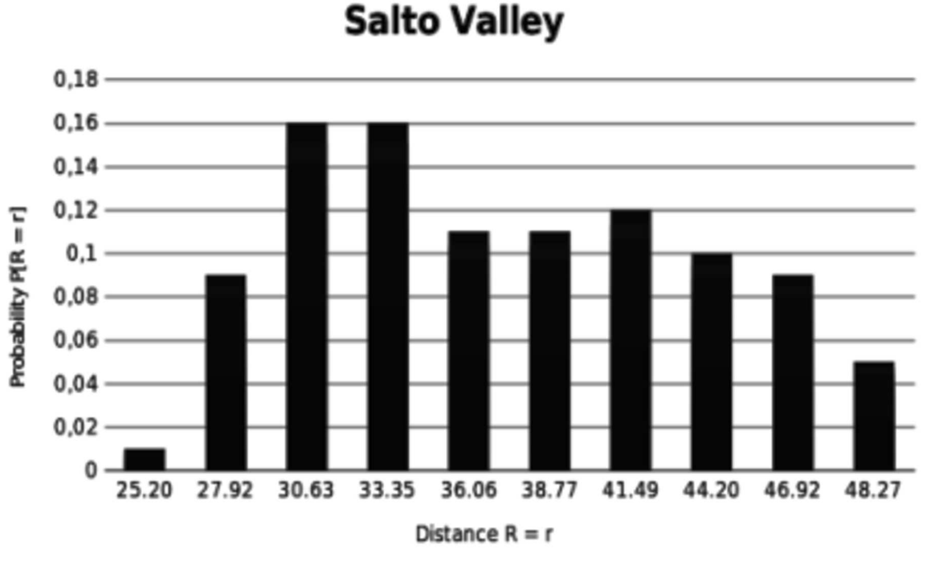

Figure 47. Distance Probability Mass Function for Salto Valley Fault

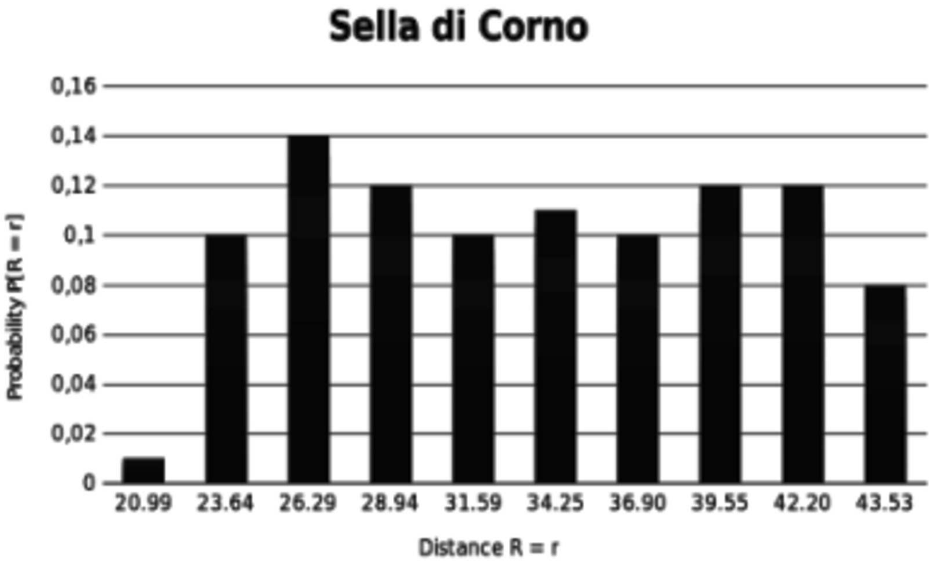

Figure 48. Distance Probability Mass Function for Sella di Corno Fault

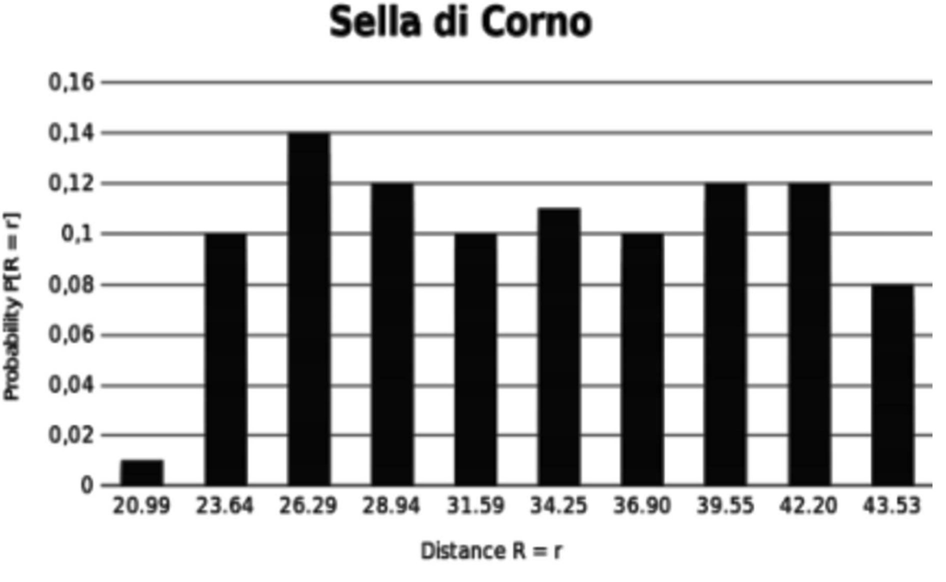

Figure 49. Distance Probability Mass Function for Sora Fault

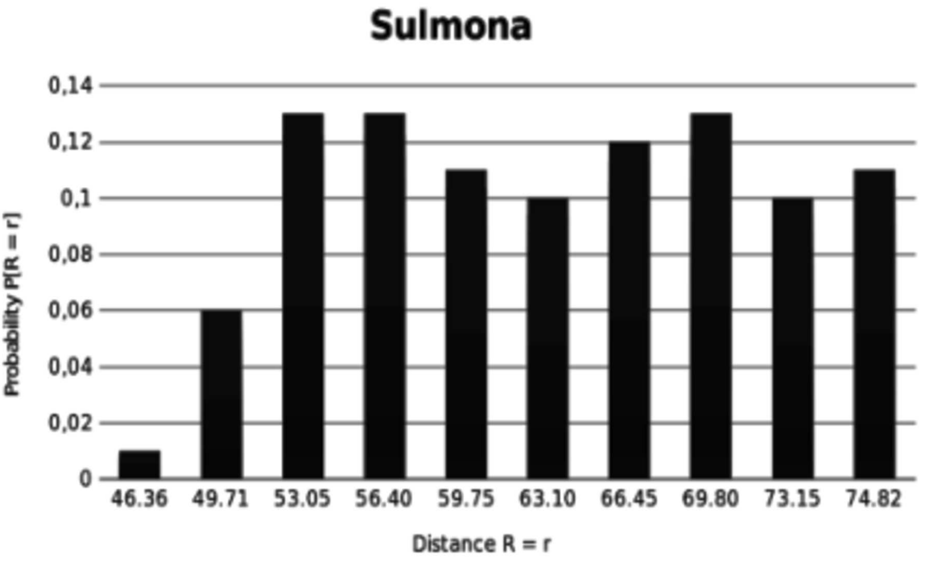

Figure 50. Distance Probability Mass Function for Sulmona Fault

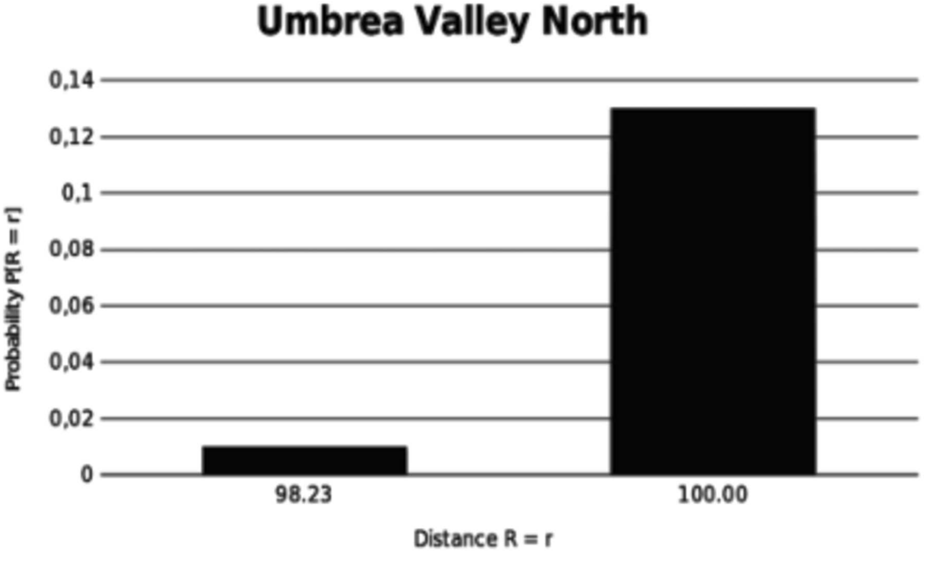

Figure 51. Distance Probability Mass Function for Umbrea Valley Fault North Segment

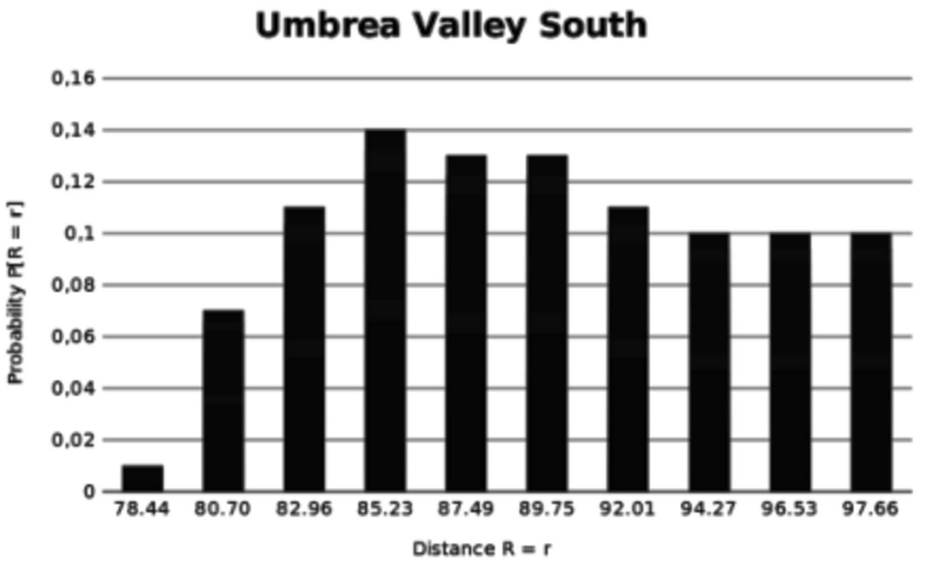

Figure 52. Distance Probability Mass Function for Umbrea Valley Fault South Segment

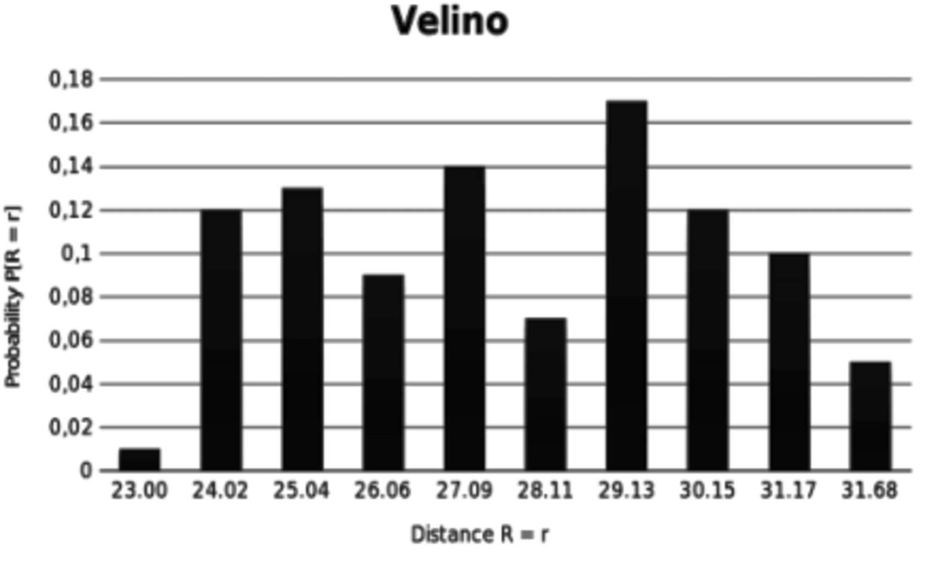

Figure 53. Distance Probability Mass Function for Velino Fault

Appendix C. MATLAB Script for Solving the One-Dimensional Elastodynamic Equation and Hooke's Law

```
% Author: Jedidiah Joel Aguirre 2019
%
% Finite-Difference seismic wave simulation
% Discretization of the first-order elastic wave equation
%
% Temporal second-order accuracy O(DT^2)
% Spatial second-order accuracy  O(DX^2)

%% Initialization
disp(' ');
disp(['Starting ', mfilename ]);
close all; clearvars;
    addpath functions

%% Parameters

%Boundary Condition Parameters
tau_one=0; %Dynamic Stress in the fault (in MPa)
tau_zero=tau_one+4.4; %Maximum Static Stress (in MPa)
t1 = .295; % characteristic time from friction law
%Crust Parameters
beta = 3200; %S-wave speed in m/sec
rho = 2600; %density of rocks in kg/m^3
mu = rho*beta^2/1e6; %lame constant

%Fault Rupture Parameters
L = 11; %fault length in km
vr = 0.9*beta; %rupture velocity in m/sec
tr = round(0.5*L*1000/vr,2); %rupture time

%Discretization Parameters and Vector Assignments
L_p = 100+.2; %prop length in km
T = round(1000*L_p/beta,2); %time of arrival in the site
tf = T+10;  % duration of seismogram
dx = 200;   % meters in spacing
dt = 0.005; %time interval for typical seismograms
x = 0:dx:L_p*1000; % x vector
J = numel(x); % no. of grids
t = 0:dt:tf+dt; % time vector
nt = numel(t);% no. of time steps

% Preallocate space
p = zeros(nt,J);    % pressure matrix
v = zeros(nt,J); % velocity matrix

%Initial conditions
for i=2:J
    p(1,i)=0;  %zero pressure
    v(1,i)=0;  %zero velocity
end

%Boundary conditions
for i = 1:nt
    v(i,1) = 0; %zero velocity
    [p(i,1)] = tau(dt*i,tr,t1,tau_one,tau_zero); %friction law at the boundary
end
```

```
%update velocity and pressure
for n=2:nt;
    for kx = 1:J-1
         % Calculating spatial derivative
            p_x=(p(n-1,kx+1)-p(n-1,kx))/dx;

            % Update velocity
            v(n-1,1)=0;
            v(n,kx)=v(n-1,kx)+(dt/rho)*p_x*1e6;
    end

    % Update pressure
    for kx=2:J;

            % Calculating spatial derivative
            vx_x=(v(n,kx)-v(n,kx-1))/dx;

            % Update pressure
            p(n,kx)=p(n-1,kx)+mu*dt*(vx_x);

    end
end


a = diff(v(:,J-1))./diff(t')/9.81;
acc_max = max(max(a),abs(min(a)));
disp(['PGA = ',num2str(acc_max),'g'])
```

References