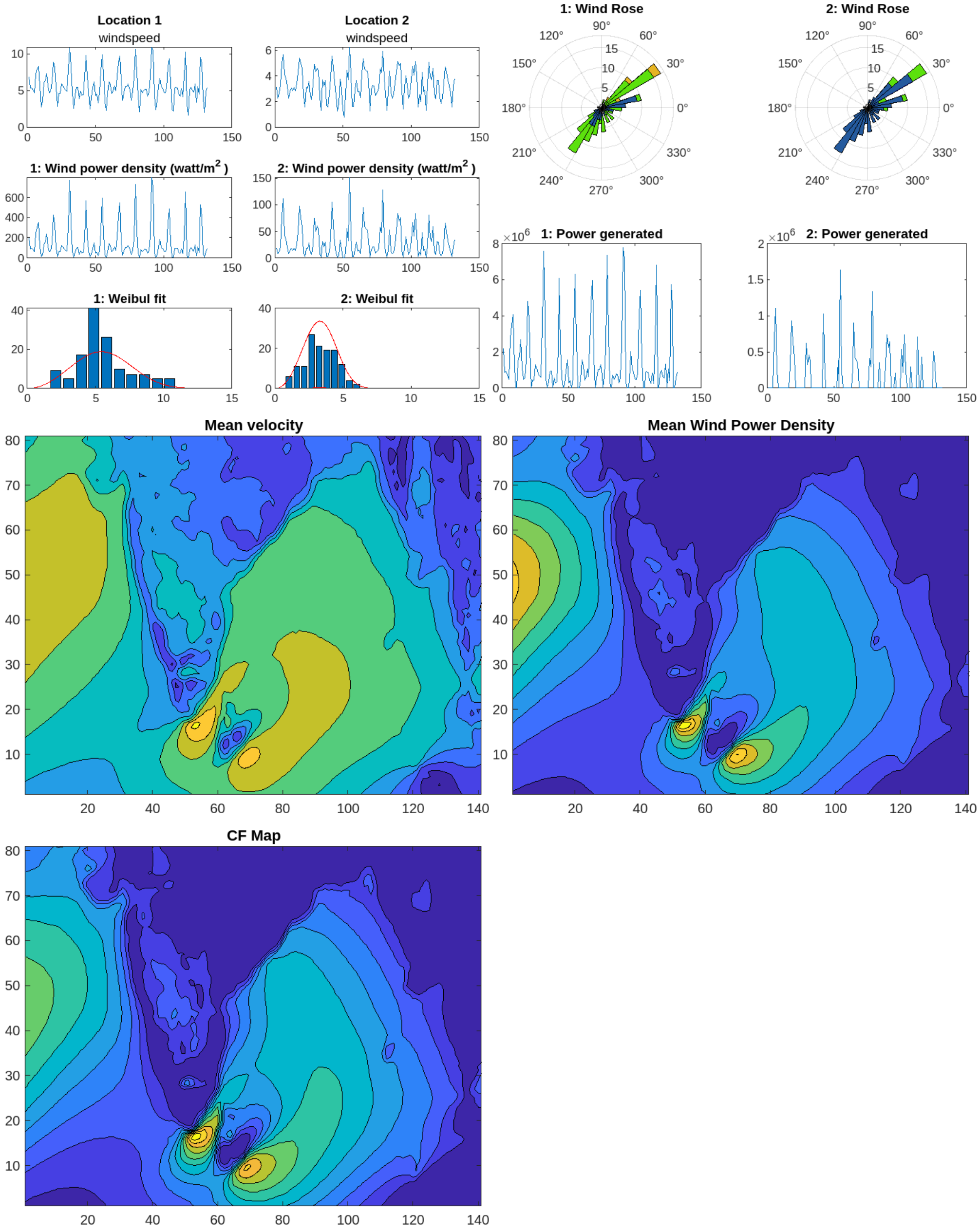
```
u100 = ncread("/MATLAB Drive/adaptor.mars.internal-1702029592.401762-29764-12-d9b88cdf-3135-40c5-
8094-43f8e64e1078.nc", 'u100');
v100 = ncread("/MATLAB Drive/adaptor.mars.internal-1702029592.401762-29764-12-d9b88cdf-3135-40c5-
8094-43f8e64e1078.nc", 'v100');
time = ncread("/MATLAB Drive/adaptor.mars.internal-1702029592.401762-29764-12-d9b88cdf-3135-40c5-
8094-43f8e64e1078.nc", 'time');
longitude = ncread("/MATLAB Drive/adaptor.mars.internal-1702029592.401762-29764-12-d9b88cdf-3135-
40c5-8094-43f8e64e1078.nc", 'longitude');
latitude = ncread("/MATLAB Drive/adaptor.mars.internal-1702029592.401762-29764-12-d9b88cdf-3135-
40c5-8094-43f8e64e1078.nc", 'latitude');

% calculate resultant and direction of velocity

v_resultant = sqrt(u100.^2+v100.^2);
v_dir       = atan2d(v100, u100);

% contour plot of wind speed
v_mean = mean(v_resultant, 3); % mean wrt time dimension
velocity_mean = v_mean(:, 81:-1:1);

rho=1.225;

% contour plot of wind power density
WPD_net = 0.5*rho*v_resultant.^3;
WPD_mean = mean(WPD_net,3);
WPD_resultant = WPD_mean(:, 81:-1:1);

% contour plot of capacity factor
Pwr = 8e6;
for i = 1:141
    for j = 1:81
        for k = 1:132
                U = v_resultant;
                if (U(i,j,k)<4)
                        Pw_net(i,j,k)=0;
                elseif (U(k)>14)
                        Pw_net(i,j,k) = Pwr;
                else
                        Pw_net(i,j,k) = (-1.306*(U(i,j,k)^4)+28.085*(U(i,j,k)^3)-88.859*
(U(i,j,k)^2)+30.288*(U(i,j,k))+30.808)*1000;
            end
        end
    end
end

Pwr_mean = mean(Pw_net,3);
Pwr_resultant = Pwr_mean(:, 81:-1:1);
CF_resultant = Pwr_resultant/Pwr;

% define locations (longitude, latitude)

location_1 = [29 21];
location_2 = [31 4];

% extract time series data from location

location_1_v = v_resultant(29, 21, :);
location_1_d = v_dir(31, 4, :);

location_2_v = v_resultant(31, 4, :); %% 4th row, 31st column (longitude, latitude)
location_2_d = v_dir(31, 4, :);

% change to single dimensional array

location_1_velocity     = location_1_v(:);
```

```
location_1_direction    = location_1_d(:);

location_2_velocity     = location_2_v(:);
location_2_direction    = location_2_d(:);

% weibul parameters

[a_1] = wblfit(location_1_velocity);
[a_2] = wblfit(location_2_velocity);


[h_1 x_1] = hist(location_1_velocity,10);
x__1=linspace(0,12);
q_1 = size(x__1);
wb_1=wblpdf(x__1,a_1(1,1),a_1(1,2));
wb_1=wb_1(:);

[h_2 x_2] = hist(location_2_velocity,10);
x__2=linspace(0,12);
q_2 = size(x__2);
wb_2=wblpdf(x__2,a_2(1,1),a_2(1,2));
wb_2=wb_2(:);




% power generation
Pwr = 8e6;

for k = 1:132
        U = location_1_velocity;
    if (U(k)<4)
                Pw_1(k)=0;
        elseif (U(k)>14)
                Pw_1(k) = Pwr;
        else
                Pw_1(k) = (-1.306*(U(k)^4)+28.085*(U(k)^3)-88.859*(U(k)^2)+30.288*
(U(k))+30.808)*1000;
    end
end

for k = 1:132
        U = location_2_velocity;
        if (U(k)<4)
                Pw_2(k)=0;
        elseif (U(k)>14)
                Pw_2(k) = Pwr;
        else
                Pw_2(k) = (-1.306*(U(k)^4)+28.085*(U(k)^3)-88.859*(U(k)^2)+30.288*
(U(k))+30.808)*1000;
    end
end

% capacity factor
CF_1 = Pw_1/Pwr;
CF_2 = Pw_2/Pwr;


% WPD
WPD_1 = 0.5*rho*location_1_velocity.^3
WPD_2 = 0.5*rho*location_2_velocity.^3

% plotting
tiledlayout(3,4)

nexttile([1 2])
plot(location_1_velocity)
title("Location 1", "windspeed")
nexttile([1 2])
plot(location_2_velocity)
```

```
title("Location 2", "windspeed")

nexttile([1 2])
plot(WPD_1)
title("1: Wind power density (watt/m^2 )")
nexttile([1 2])
plot(WPD_2)
title("2: Wind power density (watt/m^2 )")

nexttile([1 2])
bar(x_1,h_1)
hold on
plot(x__1,wb_1*q_1,'r')
title("1: Weibul fit")
nexttile([1 2])
bar(x_2,h_2)
hold on
plot(x__2,wb_2*q_2,'r')
title("2: Weibul fit")

figure
tiledlayout(2,4)
nexttile([1 2])
wind_rose(location_1_direction, location_1_velocity)
title("1: Wind Rose")
nexttile([1 2])
wind_rose(location_2_direction, location_2_velocity)
title("2: Wind Rose")

nexttile([1 2])
plot(Pw_1)
title("1: Power generated")
nexttile([1 2])
plot(Pw_2)
title("2: Power generated")

figure
contourf(velocity_mean');
title("Mean velocity")

figure
contourf(WPD_resultant');
title("Mean Wind Power Density")

figure
contourf(CF_resultant');
title("CF Map")
```


Location 1
windspeed
Location 2
windspeed
1: Wind Rose
2: Wind Rose
90°
60°
30°
0°
330°
300°
270°
240°
210°
180°
150°
120°
5
10
15
1: Wind power density (watt/m^2)
2: Wind power density (watt/m^2)
$\times 10^6$ 1: Power generated
$\times 10^6$ 2: Power generated
1: Weibul fit
2: Weibul fit
Mean velocity
Mean Wind Power Density
CF Map
0
50
100
150
20
40
60
80
100
120
140