```
%% Fitting ARMA For Non Stationary series
[a0,model0,c0] = PostulateARMA(number_people, 0.95)

r0 = roots(model0.a)
plot(r0, 'o')
hold on
n=1000;
theta = linspace(0, 2*pi, n);
x1 = cos(theta);
y1 = sin(theta);
plot(x1,y1)
%%
mo =roots(model0.c)
%hold
%% Fitting Linear model to remove the trend in the data
X = 0:484
linear_model = fitlm(X',number_people)
residuals_linear = linear_model.Residuals.Raw
RSS_linear = sum(residuals_linear.^2)
%% Checking for polynomal Non-linear (order 2) regression model
modelfun = @(b,x)b(1) + b(2)*x(:,1) + b(3)*x(:,1).^2
beta0 = [50 20 50]
nonlinear_model = fitnlm(X', number_people, modelfun, beta0)
residuals_nonlinear = nonlinear_model.Residuals.Raw
RSS_nonlinear = sum(residuals_nonlinear.^2)

%% Linear model is selected on the basis of F-test
%Use postualte ARMA to fit ARMA for linear residuals
[a,model,c] = PostulateARMA(residuals_linear,0.95)

autocorr(c)

r = roots(model.a)
plot(r, 'o')
hold on
n=1000;
theta = linspace(0, 2*pi, n);
x = cos(theta);
y = sin(theta);
plot(x,y)

%% For doing F-Test to check for significant stochastic seasonality

l = r
abs(l)
q = find(abs(l)<1.03 & abs(l)>0.9)
l1 =l(q) %only roots on unit circle

l2=[];
for i=(1:length(l1))
```

```
    if imag(l1(i))~=0
    l2 = [l2 ; l1(i)];
    end
end
l2 = l2(1:2:length(l2)-1) % only unique complex conjugate roots
w = angle(l2)
p =2*pi./w %all the possible stochastic periods
p1= round(p)

P=zeros(length(p1),1);
x = residuals_linear;
for j=(1:length(p1))

y=[];
for i=(3:length(x))
   y(i-2) = x(i)+ x(i-2)- (2*cos(2*pi/p1(j))*x(i-1));
end;


sys = armax(y',[(length(model.a)-2),(length(model.c))])

s=2;
r1=(length(model.a)-1) + (length(model.c)-1) + 1;
r2=(length(sys.a)-1) + (length(sys.c)-1) + 1;
dof1 = length(x)-r1;
dof2 = length(y')-r2;
RSSo(j) = (model.Report.Fit.MSE)*dof1;
RSSp(j) = (sys.Report.Fit.MSE)*dof2;


F(j) = ((RSSp(j) - RSSo(j))/s)/(RSSo(j)/dof1)

if F < finv(0.95,2,dof1)
    P(j)=1
    D =['seasonality of period',p1(j),'is confirmed'];
    disp(D)
end
end

%% Removing periods of 7,4 seasonalities from the model
x = residuals_linear
y=[];
for i=(3:length(x))
   y(i-2) = (x(i)+ x(i-2)- (2*cos(2*pi/7)*x(i-1)))*(x(i)+ x(i-2)- (2*cos(2*pi/4)*x(i-↙
1)));
end;

sys_total = armax(y',[(length(model.a)-5),(length(model.c)-1)])

root = roots(sys_total.a)
```

```
plot(root, 'o')
hold on
n=1000;
theta = linspace(0, 2*pi, n);
x1 = cos(theta);
y1 = sin(theta);
plot(x1,y1)


%% ARMAV Model (temperature, people)

data1=iddata(temperature,[number_people],1);
data2=iddata(number_people,[temperature],1);

data3 = iddata(People(1:505),[temp],1);

sys1=cell(1,15);
ct=1;
for n= 1:15
 sys1{ct}=armax(data1,[n n n-1 0]);
 ct=ct+1;

end
sys2=cell(1,15);
ct=1;
for n= 1:15
 sys2{ct}=armax(data2,[n  n n-1 0]);
 ct=ct+1;

end

mse1 = []
for n=1:15
    mse1(n) = sys1{1, n}.NoiseVariance
end

mse2 = []
for n=1:15
    mse2(n) = sys2{1, n}.NoiseVariance
end

%% IT is found from mse2 ARMAV(6,5) is the best model

ArmaV = sys2{1,6}
resid(data2, sys2{1,6})

test_data= iddata(People(486:505),[temp_predict],1)
[Pred, MSE_pred] = forecast(ArmaV, data2, 21)
```