Technical University of Denmark

Time Series Analysis

02417

Assignment 3

Author: (s164229)

November 21, 2019

Contents

1 Plotting

The data that will be used for this exercise is measurements of CO_2 concentrations over the years. The data is divided into training and test data (last 20 observations = test data). As illustrated in figure 1, that the concentration is oscillating and overall increasing. It can therefore conclude that the data is **not stationary, since the mean is increasing.** The data is monthly based therefore 12 observations pr year. By this, it is expected that the data contains a seasonal trend.

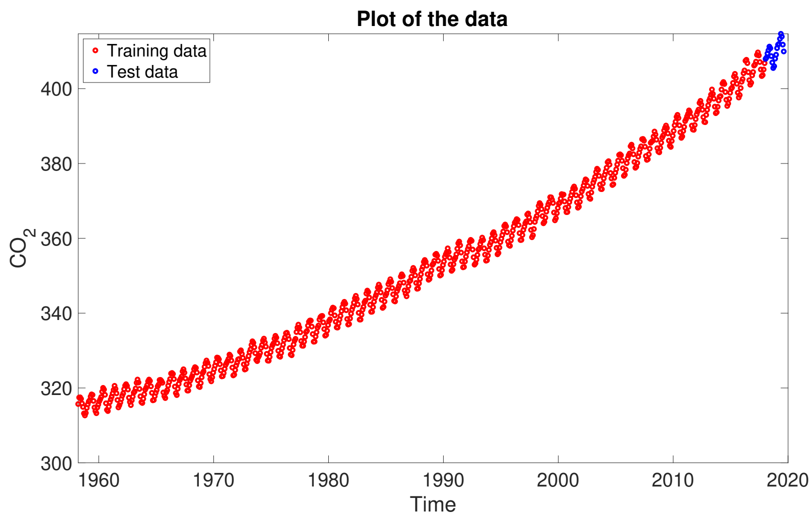

Figure 1

2 Correlation structure

First time running the ACF and PACF: It reveals that the data is not stationary, hence the ACF is slow converging in the ACF it can be seen on first column in figure 2. The top graph (same as figure 1) illustrates the data is non-stationary, as it has an increasing mean.

One lag differencing: Transformation or differencing is done for the purpose of removing slow variations such as this trend of increasing mean. According to (6.9) by removing the previous value

$$W_t = Y_t - aY_{t-1} \tag{1}$$

The ACF is still not converging sufficiently fast enough towards zero, at the same time revealing seasonal trend. This is shown in the second column in figure 2.

Seasonal differencing: It is obvious that there's oscillations and seasonal components in the ACF (figure 2) every 12Th month after one lag differencing. However the amplitudes in the ACF does not seem to decrease fast enough towards zero. The PACF has the highest positive amplitude at lag k=0, k=1, k=12 and so on. Both plots indicates that a seasonal differencing is needed. The result of seasonal differencing is shown in the last colunm in figure 2.

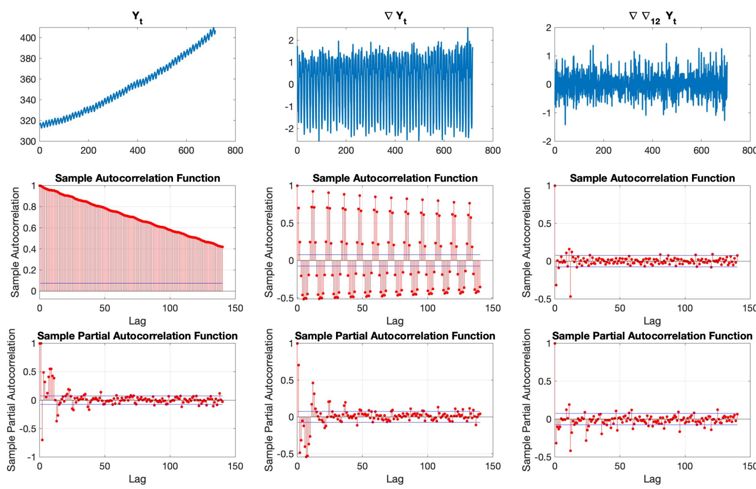

Figure 2: Showing the transformation steps

3 Procedure for identifying ARIMA model

For this assignment, the iterative procedure has been used for identification.

3.1 Expectations

After the transformations, it is defined that the model is a seasonal model with s=12 after transformed with one-lag differencing (d=1). These parameters will be fixed throughout the rest of model identification.
From figure 2 for $\nabla\nabla_{12}Y_t$ it is clear that both the ACF and PACF have peaks that can indicate pure MA model (MA(1) or MA(12) because ACF can be considered zero after lag 12) or ARMA model since both the ACF and PACF could indicate exponentially decaying behavior. The procedure is therefore easier to start with the lowest order and verify throughout the residual analysis if there is any need for increasing model orders. The residual analysis is based on following criteria:

3.2 Model checking Criteria

General white noise signal is evaluated in order for later comparison:

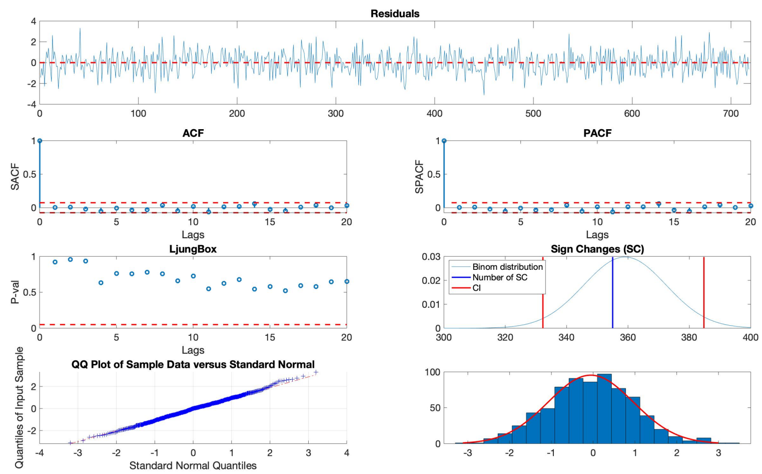

Figure 3: This is for illustrating **pure white noise signal** generated directly from build in function in MATLAB.

Following steps shows what there has been done in order to select the model order and the verification steps:

- The residuals should resemble white noise (see figure 3).
- The residuals should not have any trends (e.g seasonal). This can be verified through the first plot residuals over time or the correlation functions- ACF & PACF e.g they should be zero / within the confidence intervals.
- **Ljung Box test** Is a test for testing auto correlation function on the basis of null-hypothesis that the residuals have no auto-correlation. So if the p-values are below 5% we have to reject hypothesis. When p-values are above, there is not enough evidence to reject the null hypothesis. - This is expected for the case of white noise, that for all lags, the p-values should be above 5% (see figure 3).
- White noise / residuals should be normal distributed with mean = zero. This can be verified through histogram plot, that the curve resembles a standard normal distribution function. The QQplot also reveals if the residuals are normal distributed.
- The number of **sign changes** should be approximately half of the number of observations. For this we calculate if the number is within the 95% confidence interval.

- **Model comparison** When comparing models, **Maximum likelihood** can be used for those models that are nested. For non-nested models **AIC** (Akaike's Information Criterion) value, model with the smallest AIC value will be the most accurate.[1] - This comparison can done in order to determine between models that are similar after several models has been estimated.

All these criteria is applied on the white noise signal shown in figure 3. The code *Diagnostics_ARIMA.m* is attached in appendix.

4 ARIMA model

ARIMA	Test result	AIC	ML	Sign change
$(0,1,1)(1,0,0)_{12}$	Failing Ljung Box	546.4607	-270.2304	387.0
$(1,1,1)(1,1,0)_{12}$	Failing Ljung Box	544.0789	-269.0395	371.0
$(1,1,1)(1,1,1)_{12}$	Perfect	390.3895	-191.1948	363.0
$(1,1,1)(0,1,1)_{12}$	Perfect	391.5998	-192.7999	366.0
$(0,1,1)(0,1,1)_{12}$	Failing Ljung Box	397.4782	-195.7391	368.0

Test result is the performance according to the plots shown in figure 5. In terms of comparing the models shown in table **maximum log likelihood** can be used since all of the models are **nested**.

4.1 $ARIMA(1,1,1)(1,1,1)_{12}$

The first two selection of model revealed for lag = 12 both in ACF, PACF and Ljung box test that there were correlation. This issue is solved by introducing more terms in the seasonal order. From the table and further analysis it shows that the model number 3 is the best performing (see figure 5). There's no correlation in ACF and PACF, and no reason for rejecting the null-hypothesis based on Ljung box test. Histogram and QQ-plot confirms the normality. Lastly the sign test showing that the number of sign changes for this model is within the confidence interval.

```
                Value      StandardError    TStatistic      PValue
              _________    _____________    __________    ___________

Constant              0                0           NaN            NaN
AR{1}           0.19625         0.072131        2.7207      0.0065135
SAR{12}       -0.078547         0.039356       -1.9958       0.045955
MA{1}          -0.58359         0.061496       -9.4898     2.3145e-21
SMA{12}        -0.78358         0.025853       -30.309    8.6214e-202
Variance       0.099729        0.0046935        21.248    3.4228e-100
```

Figure 4: Model 3 output $ARIMA(1,1,1)(1,1,1)_{12}$

[1] According to section 6.5.3 in Text book (Time Series Analysis by Henrik Madsen)

Although the numbers shown in the table seem good, but when looking at the standard error for SAR in figure 4, the magnitude of standard error is close to parameter value itself $\frac{0.078547}{0.039356} \approx 2$. Therefore model number 4 SAR term is excluded for the purpose of testing whether there's a difference between model 3 and 4. This resulted in a slight decrease in the likelihood without any significant change in the diagnostic plots.

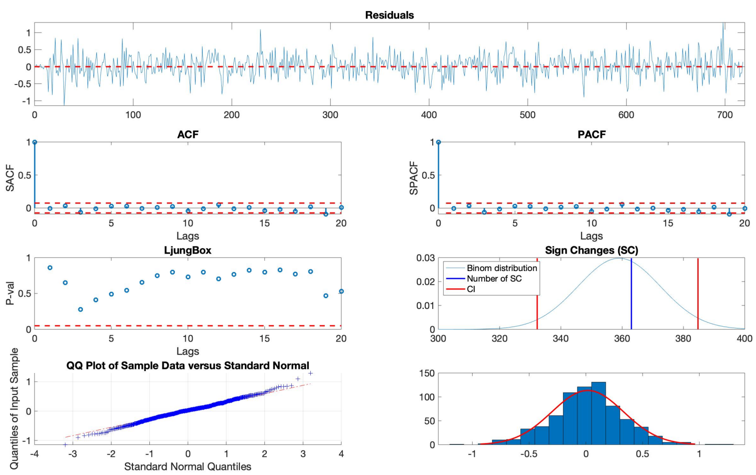

Figure 5: Model $ARIMA(1,1,1)(1,1,1)_{12}$, looks almost just identical to the figure 3, in terms of the tests, this model is performing very well.

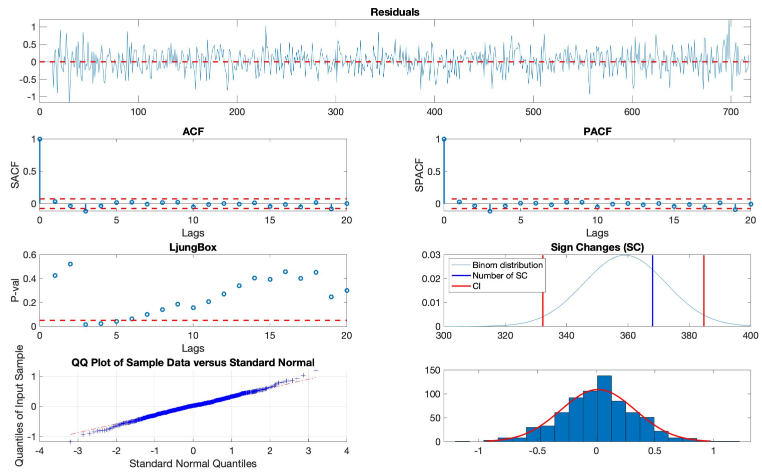

Figure 6: This is the last model $(0,1,1)(0,1,1)_{12}$, which shows the Ljung box test is significant for lag 3, 4 and 5. From ACF there's also a small peak at lag 4, which is quite close to the boundaries. When looking at the Ljung box test at these lags which indicates the null-hypothesis should be rejected for exactly these lags.

As the Standard error for model 4 is large compared to the estimated AR parameter value, the AR term was excluded in model 5, this attempt was a deterioration compared to previous models (see figure 6), this made up the stopping criteria, that it is not possible to obtain better models by decreasing the order, and no reason to increase model order when model 3 is enough for making the residuals resemble white noise.

Based on the maximum log likelihood and test results as main criteria for selecting ideal model, **then the best performing model will be number 3** ARIMA$(1,1,1)(1,1,1)_{12}$.

5 Predictions

The model that is used here is ARIMA $(1,1,1)(1,1,1)_{12}$. The predictions for 1,2,6,12 and 20 months ahead is shown in following table.

Months	Test Data	Predicted	MSE	Lower CI	Upper CI
1	407.96	408.0066	0.12388	407.3167	408.2494
2	408.32	408.9407	0.16991	408.1328	409.2737
6	410.79	411.5079	0.31909	410.4007	412.1333
12	409.07	409.2803	0.54074	407.839	410.3402
20	409.95	409.8896	1.257	407.6921	412.3533

The 48 months prediction is illustrated in figure 7. As expected the predicted values follow same pattern as the test data, both the oscillations and the increasing trend. The predictions are quite accurate when comparing with test data. Confidence interval width is increasing with the months as the error increases with step size. This also indicates that we get a more inaccurate estimate when the month step increases.

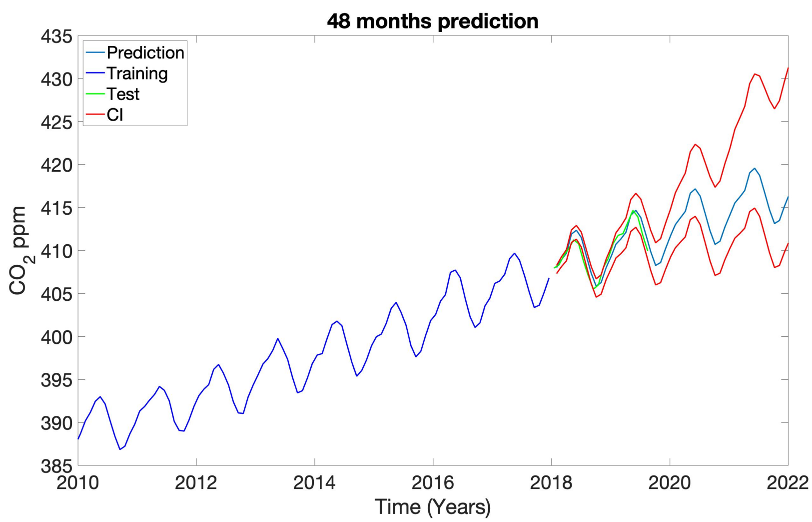

Figure 7: Predictions for 48 months ahead compared with test data.

6 Attaining 460 ppm

Same model (ARIMA $(1,1,1)(1,1,1)_{12}$) has been used to make prediction shown in figure 8. When investigating the prediction it is noticeable that the confidence level is very large and MSE is as well too large in terms of an ideal MSE is zero.(see table below the figure)

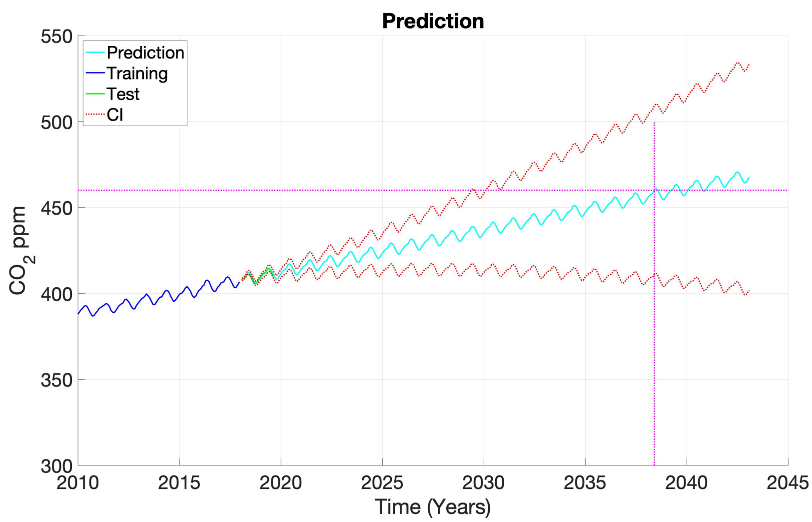

Figure 8: Illustrating the prediction further ahead in order to find the month where the CO

According to this ARIMA model chosen in question 3, the CO_2 concentration is expected to attain 460 around July 2038 which corresponds to 244 months ahead from the last training data:

Months ahead	**Value**	**MSE**	**Lower CI**	**Upper CI**
244	460.42	624.6223	411.4348	509.4052

And from what we do expect, is that we see the upper confidence interval do hit the 460 boundary already at 2030. (see figure 8) ***Which means that we from 2030 there is an increasing chance of obtaining the 460ppm CO_2.***

7 Appendix (code)

Listing 1: *Diagnostics_ARIMA.m*

```
function Diagnostics_ARIMA(data)
% calculate residuals
figure('WindowState','maximized')
subplot(4,2,[1 2])
plot(data); hold on
xlim([0 720])
line(xlim,[0 0],'Color','red','LineStyle','--','
    LineWidth',2);
title('Residuals');
set(gca, 'fontsize',15)

% Autocorrelation (ACF)
subplot(4,2,3)
[acf,lags,bounds] = autocorr(data);
stem(lags,acf,'LineWidth',2 ); hold on
line(xlim, [bounds(1) bounds(1)] ,'Color','red','
    LineStyle','--','LineWidth',2 )
line(xlim, [bounds(2) bounds(2)] ,'Color','red','
    LineStyle','--','LineWidth',2 )
title('ACF'); xlabel('Lags'); ylabel('SACF')

set(gca, 'fontsize',15)

% Partial-ACF
subplot(4,2,4)
[pacf,lagsp,boundsp] = parcorr(data);
stem(lagsp,pacf,'LineWidth',2 ); hold on
line([0.5 20], [boundsp(1) boundsp(1)] ,'Color','red',
    'LineStyle','--','LineWidth',2 )
line([0.5 20], [boundsp(2) boundsp(2)] ,'Color','red',
    'LineStyle','--','LineWidth',2 )
title('PACF') ; xlabel('Lags'); ylabel('SPACF')
set(gca, 'fontsize',15)


subplot(4,2,5)
[~,pValue] = lbqtest(data,'lags',1:20);
plot(pValue,'o','LineWidth',2); title('LjungBox') ;
    hold on
line(xlim,[0.05 0.05],'Color','red','LineStyle','--','
    LineWidth',2);
```

```
xlabel('Lags'); ylabel('P-val')
set(gca, 'fontsize',15)

subplot(4,2,6)
% signtest
Sign=(data(2:end).*data(1:end-1)>0);
% stem(Sign);
CI_sign=(length(data)-1)/2+[-1 1]*1.96*sqrt((length(
    data)-1)/4);
N=length(data);
y=binopdf(300:400,N,0.5);
plot(300:400,y); hold on
plot([sum(Sign) sum(Sign)],ylim, 'b', 'LineWidth',2)
plot([CI_sign(1) CI_sign(1) ], ylim, 'r-','LineWidth'
    ,2)
plot([CI_sign(2) CI_sign(2) ], ylim, 'r-', 'LineWidth'
    ,2)
legend('Binom distribution','Number of SC','CI','
    Location','NW')
%ylabel('Logical Yes=1 when sign changes')
xlim([300 400])
title('Sign Changes (SC)');
set(gca, 'fontsize',15)

subplot(4,2,7)
%QQplot
qqplot(data)
grid on
set(gca, 'fontsize',15)

subplot(4,2,8)
%histogram(data); hold on
histfit(data,20,'normal')
set(gca, 'fontsize',15)

fprintf(' The number of sign changes are %4.1f ',sum(
    Sign))
end
```

Listing 2: First part

```
%% Time series analysis
% Assignment 3
% Jiayi Han F19
C = textscan(fopen('A1_co2.txt'), '%f %f %f %f','
    delimiter',' ');
```

```
AttributeNames = {'year','month','time','co2'};
Data = cell2struct(C,AttributeNames,2);
%% Divide the data into test and training
Data.test=Data.co2(end-19:end);
Data.train=Data.co2(1:end-20);
Data.testTime=Data.time(end-19:end);
Data.trainTime=Data.time(1:end-20);
%% Diff of one order:
tt=diff(Data.train);
autocorr(tt, 'NumLags',140)
%% Question 3.1: Plotting
figure('Windowstate','maximized')
plot(Data.trainTime, Data.train,'ro','LineWidth',3); hold on
plot(Data.testTime, Data.test,'bo','LineWidth',3);
legend('Training data','Test data','Location','northwest')
title('Plot of the data');
xlabel('Time'); ylabel('CO_2'); set(gca, 'fontsize',30)
hgexport(gcf,'Ex3_1.eps')
%% Question 3.2: Correlation structure
% Plot the autocorrelation function and
% the partial autocorrelation function of the CO2 concentration.
figure('Windowstate','maximized')
subplot(2,1,1)
autocorr(Data.train,'NumLags',140);
set(gca, 'fontsize',20)
subplot(2,1,2)
parcorr(Data.train,'NumLags',140);
set(gca, 'fontsize',20)
%% Differencing, seasonal and lag 1
Differen_one_lag=Data.train(2:end)-Data.train(1:end-1);
%plot(Differen_one_lag)
figure('Windowstate','maximized')
subplot(2,1,1)
autocorr(Differen_one_lag,'NumLags',140);
set(gca, 'fontsize',20)
subplot(2,1,2)
parcorr(Differen_one_lag,'NumLags',140);
set(gca, 'fontsize',20)
%% AR(1) model with 1-lag differencing
model = arima(1,0,0);
TestModel1=estimate(model,Differen_one_lag);
```

```
mdl = arima('AR',TestModel1.AR ,'Constant',0,'Variance
    ',1) ;
Y = simulate(mdl,700);
plot(Y)
figure('Windowstate','maximized')
subplot(2,1,1)
autocorr(Y,'NumLags',140);
set(gca, 'fontsize',20)
subplot(2,1,2)
parcorr(Y,'NumLags',140);
set(gca, 'fontsize',20)
%% Auto-correlation and partial correlation
Differen_tw_lag=Data.train(12:end)-Data.train(1:end
    -11);
figure('Windowstate','maximized')
subplot(2,1,1)
autocorr(Differen_tw_lag,'NumLags',140);
set(gca, 'fontsize',20)
subplot(2,1,2)
parcorr(Differen_tw_lag,'NumLags',140);
set(gca, 'fontsize',20)
%% ACF PACF 12-lag
model = arima(2,0,2);
TestModel2=estimate(model,Differen_tw_lag,'Display','
    off');
mdl = arima('AR',TestModel2.AR,'MA', TestModel2.MA ,'
    Constant',0,'Variance',1) ;
Y = simulate(mdl,700);
%plot(Y)
figure('Windowstate','maximized')
subplot(2,1,1)
autocorr(Y,'NumLags',140);
set(gca, 'fontsize',30)
subplot(2,1,2)
parcorr(Y,'NumLags',140);
set(gca, 'fontsize',30)
%% Compare data
figure('WindowState','Maximized')
subplot(2,1,1)
plot(Differen_one_lag)
ylim([-6 6])
title('1-lag differencing')
set(gca, 'fontsize',30)
subplot(2,1,2)
plot(Differen_tw_lag)
ylim([-6 6])
```

```
title('12-lag differencing')
set(gca, 'fontsize',30)

%%
TL=1;
Tv=[];
for k=13:length(Differen_one_lag)
    Tv(TL)=Differen_one_lag(k)-Differen_one_lag(k-12);
    TL=TL+1;
end
%% Plot ACF and PACF for the data that has been
   differenced:
figure('WindowState','maximize')
subplot(3,3,1)
plot(Data.train,'LineWidth',2)
title('Y_t')
set(gca, 'fontsize',15)

subplot(3,3,2)
plot(Differen_one_lag,'LineWidth',2)
title('\nabla Y_t')
set(gca, 'fontsize',15)

subplot(3,3,3)
plot(Tv,'LineWidth',2)
title('\nabla \nabla_1_2  Y_t')
set(gca, 'fontsize',15)

subplot(3,3,4)
autocorr(Data.train,'NumLags',140);
set(gca, 'fontsize',15)

subplot(3,3,5)
autocorr(Differen_one_lag,'NumLags',140);
set(gca, 'fontsize',15)

subplot(3,3,6)
autocorr(Tv,'NumLags',140);
set(gca, 'fontsize',15)

subplot(3,3,7)
parcorr(Data.train,'NumLags',140);
set(gca, 'fontsize',15)

subplot(3,3,8)
parcorr(Differen_one_lag,'NumLags',140)
```

```
set(gca, 'fontsize',15)

subplot(3,3,9)
parcorr(Tv,'NumLags',140)
set(gca, 'fontsize',15)

%% Test of random white noise
TT=randn(718,1);
Diagnostics_ARIMA(TT)
```

Listing 3: Second part

```
%% Models
Models=[];
% (0,1,1)(1,0,0)12 - model
Models.M1 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1,'SARLags',12);
% (1,1,1)(1,1,0)12
Models.M2 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1,'ARLags',1,'SARLags',12);
%(1,1,1)(1,1,1)12
Models.M3 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1,'ARLags',1,'SARLags',12, 'SMALags',12);
%(1,1,1)(0,1,1)12
Models.M4 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1,'ARLags',1, 'SMALags',12);
%(0,1,1)(0,1,1)12
Models.M5 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1, 'SMALags',12);
%(0,1,1)(1,1,1)12
Models.M6 = arima('Constant',0,'D',1,'Seasonality',12,
    'MALags',1, 'SMALags',12);

%%
[EstMdl,EstParamCov,logL,info] = estimate(Models.M3,
    TTTT);
numParam=3; % 4 if it's model 3
numObs=length(TTTT);
[aic,bic] = aicbic(logL,numParam,numObs);
% Residuals:
residual=infer(EstMdl,TTTT);
stdRes = residual/sqrt(EstMdl.Variance); %
    Standardized residuals
% Plot ACF and PACF
Diagnostics_ARIMA(residual)
%%
```

```
T_stat=0.078547/ 0.039356 ;
p = 1-tcdf(T_stat,N-4) % Probability of larger t-
    statistic
```

Listing 4: Last part

```
%% Model estimation:
% D=1
% seasonal model with 12 (monthly model)
% reference to section (5.6.2)
p=1; q=1;
%TTTT= Differen_one_lag;
TTTT= Data.train;
model = arima('Constant',0,'ARLags',p,'SARLags',12,'D'
    ,1,'Seasonality',12,'MALags',q,'SMALags',12);
%model = arima('Constant',0,'ARLags',p,'SARLags',12,'
    Seasonality',12,'MALags',q,'SMALags',12);
%model = arima('Constant',0,'ARLags',p,'SARLags',12,'D
    ',1,'Seasonality',12);
%model = arima('Constant',0,'ARLags',p,'SARLags',12,'
    Seasonality',12);
%model = arima('Constant',0,'ARLags',p,'SARLags',12,'D
    ',1,'Seasonality',12,'MALags',q,'SMALags',12);
%model = arima('Constant',0,'ARLags',p,'SARLags',1,'D
    ',1,'Seasonality',12 );
[EstMdl,EstParamCov,logL,info] = estimate(model,TTTT);
numParam=length(p)+length(q)+1;
numObs=length(TTTT);
[aic,bic] = aicbic(logL,numParam,numObs);
% Residuals:
residual=infer(EstMdl,TTTT);
stdRes = residual/sqrt(EstMdl.Variance); %
    Standardized residuals
% Plot ACF and PACF
Diagnostics_ARIMA(residual)
%%
[Y,YMSE] = forecast(EstMdl,48, TTTT);
% mean squared error (MSE)  use to calculate
    confidence interval
upper= Y + 1.96*YMSE; lower= Y - 1.96*sqrt(YMSE);
Time1=linspace(2.018085106382979e
    +03,2022.018085106382979,48);
figure('WindowState','maximized')
plot(Time1,Y,'LineWidth',2); hold on
plot(Data.trainTime,Data.train,'b-','LineWidth',2)
plot(Data.testTime,Data.test,'g-','LineWidth',2)
```

```
plot(Time1,lower,'r-','LineWidth',2)
plot(Time1,upper,'r-','LineWidth',2)
xlim([2010 2022])
legend('Prediction','Training', 'Test','CI','Location'
    ,'NW')
set(gca, 'fontsize',30)
xlabel('Time (Years)'); ylabel('CO_2 ppm')
title('48 months prediction')
%%
mm=[1 2 6 12 20]';
ExportTab=[mm Data.test([1 2 6 12 20]) Y([1 2 6 12
    20]) YMSE([1 2 6 12 20]) lower([1 2 6 12 20]) upper
    ([1 2 6 12 20])];
columnLabels = {'Months ahead','Test Data','Predicted'
    ,'MSE','Lower CI','Upper CI'};
matrix2latex(ExportTab, 'Prediction.tex', '
    columnLabels', columnLabels)
%%
[Y_future,YMSE_future] = forecast(EstMdl,300, TTTT);
Time2=linspace(2.018085106382979e+03,2.018085106382979
    e+03+25,300);
upperF= Y_future + 1.96*sqrt(YMSE_future); lowerF=
    Y_future - 1.96*sqrt(YMSE_future);

% Plot results of prediction 460
% time point
figure('Windowstate','maximized')
hold on;
plot(Time2,Y_future,'c','LineWidth',2);
plot(Data.trainTime,Data.train,'b-','LineWidth',2)
plot(Data.testTime,Data.test,'g-','LineWidth',2)
plot(Time2,upperF,'r:','LineWidth',2);
plot(Time2,lowerF,'r:','LineWidth',2)
grid on
plot([Time2(min(find(Y_future>460))) Time2(min(find(
    Y_future>460)))], [300 500], 'm:', 'LineWidth',2);
plot(xlim, [460 460],'m:','LineWidth',2)
xlim([2010 2045])
legend('Prediction','Training', 'Test','CI','Location'
    ,'NW')
title('Prediction')
xlabel('Time (Years)'); ylabel('CO_2 ppm')
set(gca, 'fontsize',30)

%%
Exp=[min(find(Y_future>460)) Y_future(min(find(
```

```
    Y_future>460))) ...
     YMSE_future(min(find(Y_future>460))) lowerF(min(
        find(Y_future>460)))...
     upperF(min(find(Y_future>460))) ];
columnLabels_2 = {'Months ahead','Value','MSE','Lower
    CI','Upper CI'};
matrix2latex(Exp, 'Prediction460.tex', 'columnLabels',
     columnLabels_2)
%% Sign test CI
CI_sign=(length(Data.train)-1)/2+[-1 1]*1.96*sqrt((
    length(Data.train)-1)/4);
N=length(Data.train);
y=binopdf(300:400,N,0.5);
plot(300:400,y); hold on
plot([CI_sign(1) CI_sign(1) ], ylim, 'r-','LineWidth'
    ,2)
plot([CI_sign(2) CI_sign(2) ], ylim, 'r-', 'LineWidth'
    ,2)
```