LAB2_TS

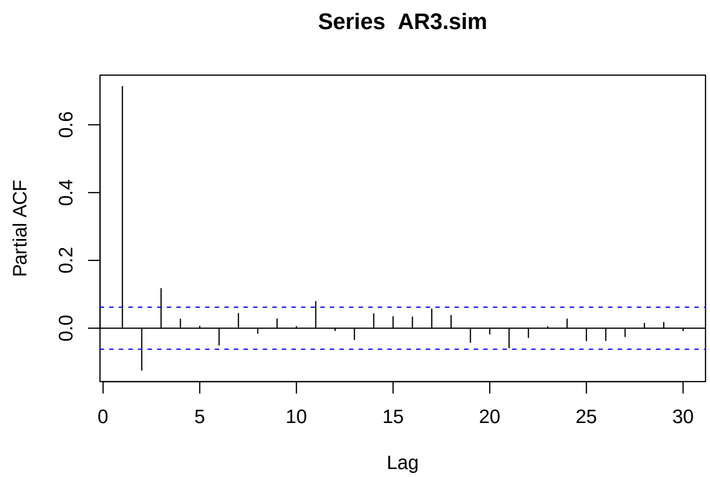
Andreas C Charitos(andch552),Ruben Muñoz (rubmu773)

9/30/2019

Assignment 1.Computations with simulated data

a)

Generate 1000 observations from AR(3) process with $\phi1$ = 0.8, $\phi2$ = -0.2, $\phi3$ = 0.1. Use these data and the definition of PACF to compute ϕ_{33} from the sample, i.e. write your own code that performs linear regressions on necessarily lagged variables and then computes an appropriate correlation. Compare the result with the output of function $pacf()$ and with the theoretical value of ϕ_{33}



The partial autocorrelation is the association between X_t and X_{t+k} with the linear dependence of X_{t+1} through X_{t+k-1} removed.Given by the formula :

$$pacf(X_t, X_{t+k}) = Corr(X_t, X_{t+k}|X_{t+1} = x_{t+1,...,}X_{t+k+1} = x_{t+k+1}$$

The results we obtain are similar calucating the corellation with linear regression between $X_t \sim X_{t-1} + X_{t-2}$ and $X_{t-3} \sim X_{t-1} + X_{t-2}$ and the output of the pacf() function for the simulated data and the theoretical PACF.

Est.Corr	Sim.PACF	Theo.PACF
0.1146076	0.1170643	0.1

b)

Simulate an AR(2) series with ϕ_1 = 0.8, ϕ_2 = 0.1 and n = 100. Compute the estimated parameters and their standand errors by using three methods: method of moments (Yule-Walker equations), conditional least squares and maximum likelihood (ML) and compare their results to the true values. Which method does seem to give the best result? Does theoretical value for ϕ_2 fall within confidence interval for ML estimate?

Table with the estimated coefficients

	phi1	phi2
YW	0.8571752	-0.0199902
OLS	0.9386075	-0.0910831
MLE	0.9015078	-0.0354404
TRUE	0.8000000	0.1000000

The above table gives the estimated coeeficients given the 3 methods.As we can see the Yule-Walker method seems to have the most accurate estimates.

```
## The esitmated interval is : -0.2192624 0.1483816
```

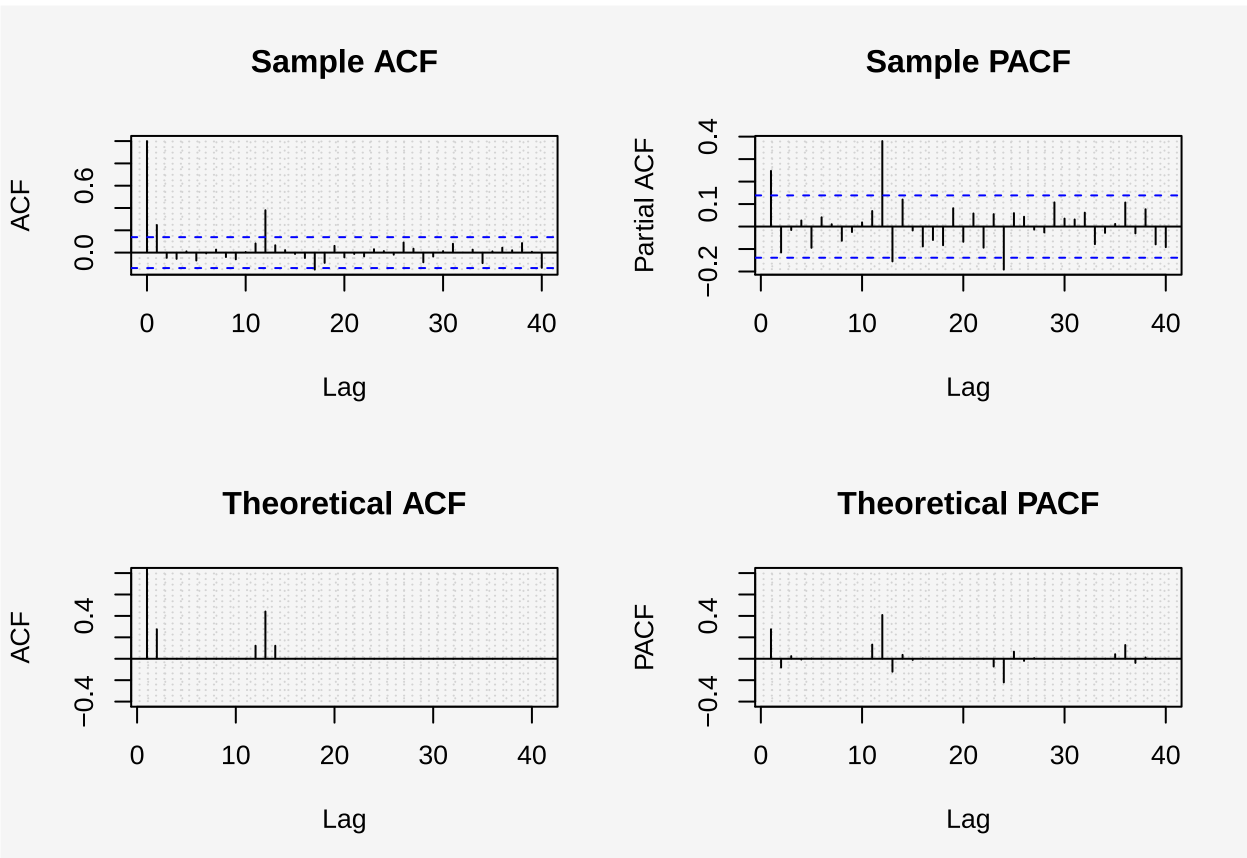
```
## The value of phi_2 is within the estimated interval?
```

```
## [1] TRUE
```

As we can see the value of ϕ_2 is within the CI for the ML estimate.

c)

Generate 200 observations of a seasonal $ARIMA(0,0,1)x(0,0,1)_{12}$ model with coefficients $\Theta = 0.6$ and $\theta = 0.3$ by using arima.sim(). Plot sample ACF and PACF and also theoretical ACF and PACF. Which patterns can you see at the theoretical ACF and PACF? Are they repeated at the sample ACF and PACF?

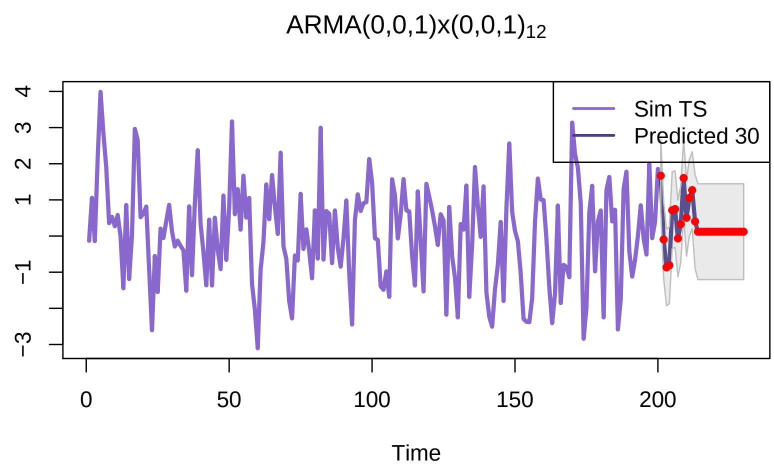


As we can see from the plots of the simulated and theoretical ACF the pattern at lag 1 and the pattern at lag 11-13 are observed on both plots. For the PACF plots we can see some seasonal patterns that are in the theoretical PACF are also present at the simulated PACF.

d)

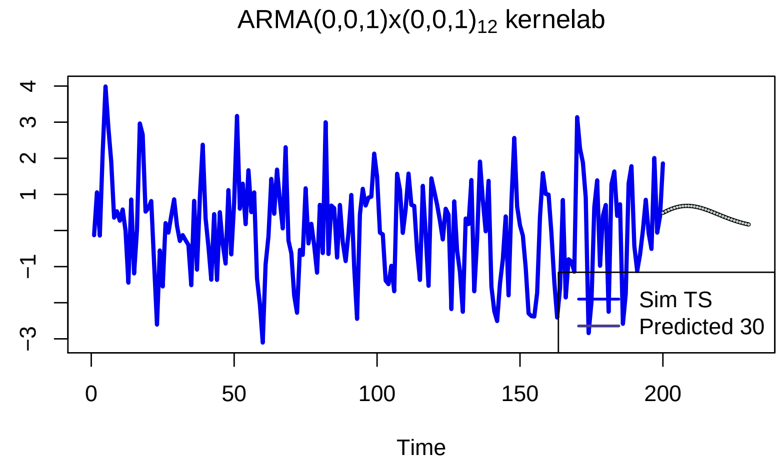
Generate 200 observations of a seasonal $ARIMA(0,0,1)Ö(0,0,1)_{12}$ model with coefficients $\Theta = 0.6$ and $\theta = 0.3$ by using arima.sim().Fit $ARIMA(0,0,1)Ö(0,0,1)_{12}$ model to the data, compute forecasts and a prediction band 30 points ahead and plot the original data and the forecast with the prediction band. Fit the same data with function gausspr from package kernlab (use default settings). Plot the original data and predicted data from t=1 to t=230.Compare the two plots and make conclusions.

Plot of predictions with $ARIMA(0,0,1)x(0,0,1)_{12}$



Plot of predictions with kernlab

```
## Using automatic sigma estimation (sigest) for RBF or laplace kernel
```

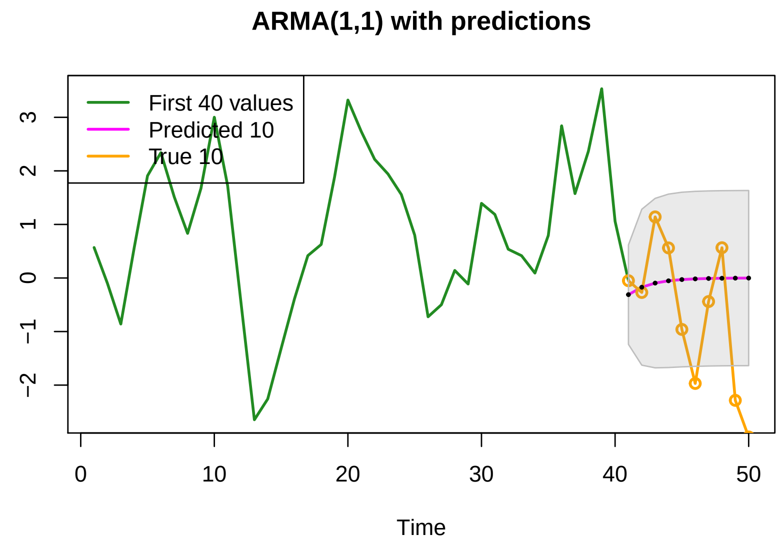


As we can see comparing the plots using the $ARIMA(0,0,1)x(0,0,1)_{12}$ we fitted and the results from the kernelab the $ARIMA(0,0,1)x(0,0,1)_{12}$ was able to produce better predictions compared to kernelab. This result may be explained due to the fact that kernelab wasn't able to capture the seasonality or trend because the prediction is based on the width of the kernel and might some previous values not include in the kernel estimate.Also the gaussian kernel which is symmetric returns the most probable prediction (or mean prediction).

e)

Generate 50 observations from $ARMA(1,1)$ process with $\phi = 0.7$, $\theta = 0.5$. Use first 40 values to fit an $ARMA(1,1)$ model with $\mu = 0$. Plot the data, the 95% prediction band and plot also the true 10 values that you initially dropped. How many of them are outside the prediction band? How can this be interpreted?

PLot of the predictions for ARMA(1,1)



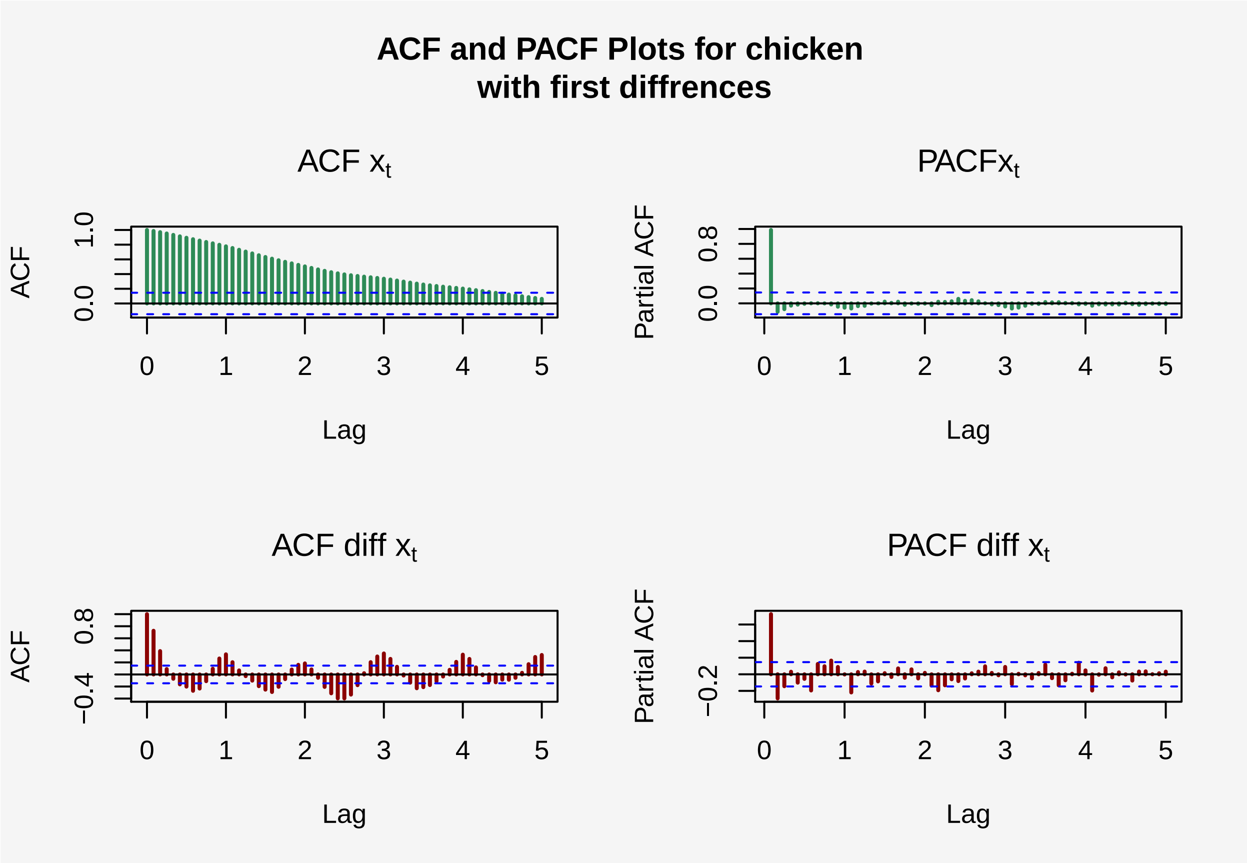

```
## The number of the values outside the prediction band is: 3
```

Assingment 2.ACF and PACF diagnostics

a)

For data series chicken in package astsa (denote it by x_t), plot 4 following graphs up to 40 lags: $ACF(x_t)$, $PACF(x_t)$, $ACF(\nabla x_t)$, $PACF(\nabla\ x_t)$ (group them in one graph). Which $ARIMA(p,d,q)$ or $ARIMA(p,d,q)x(P,D,Q)_s$ models can be suggested based on this information only? Motivate your choice.

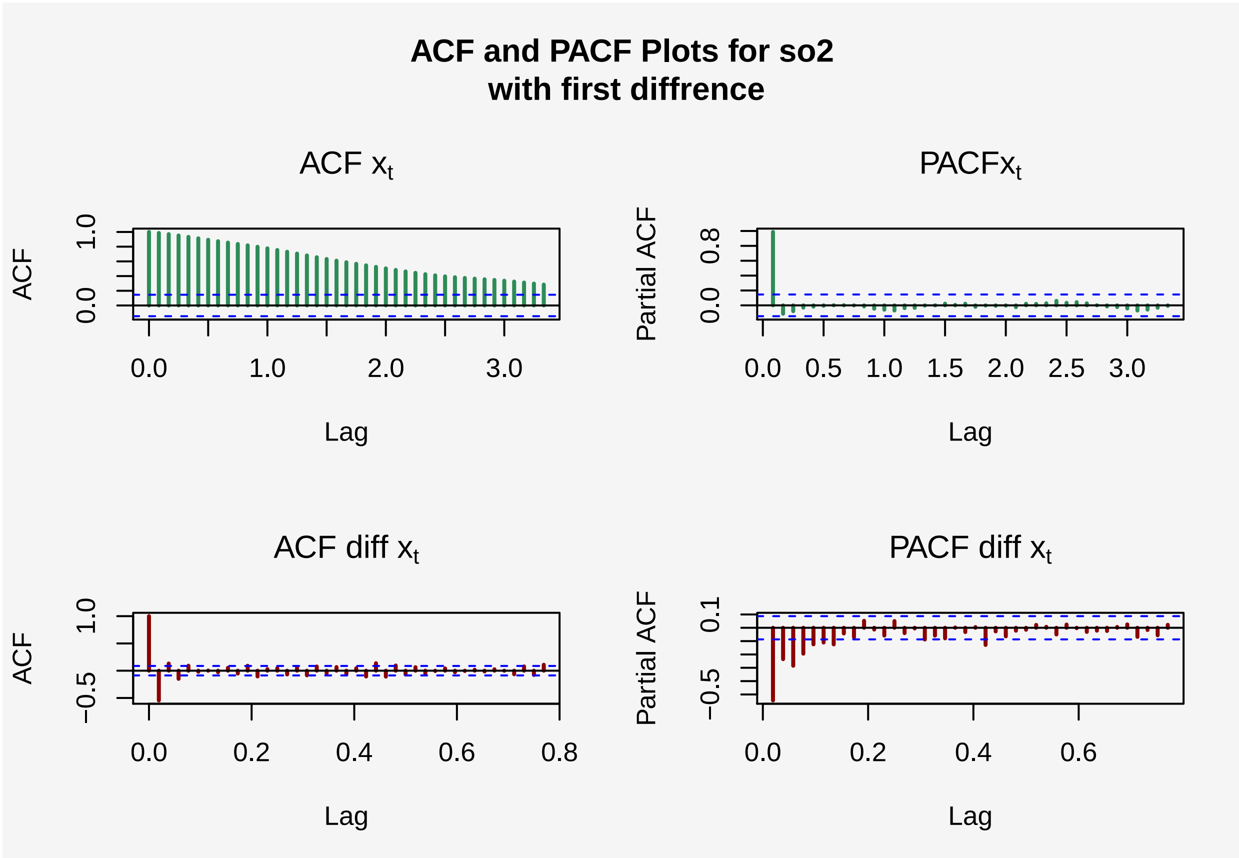
ACF-PACF Plots for chicken data



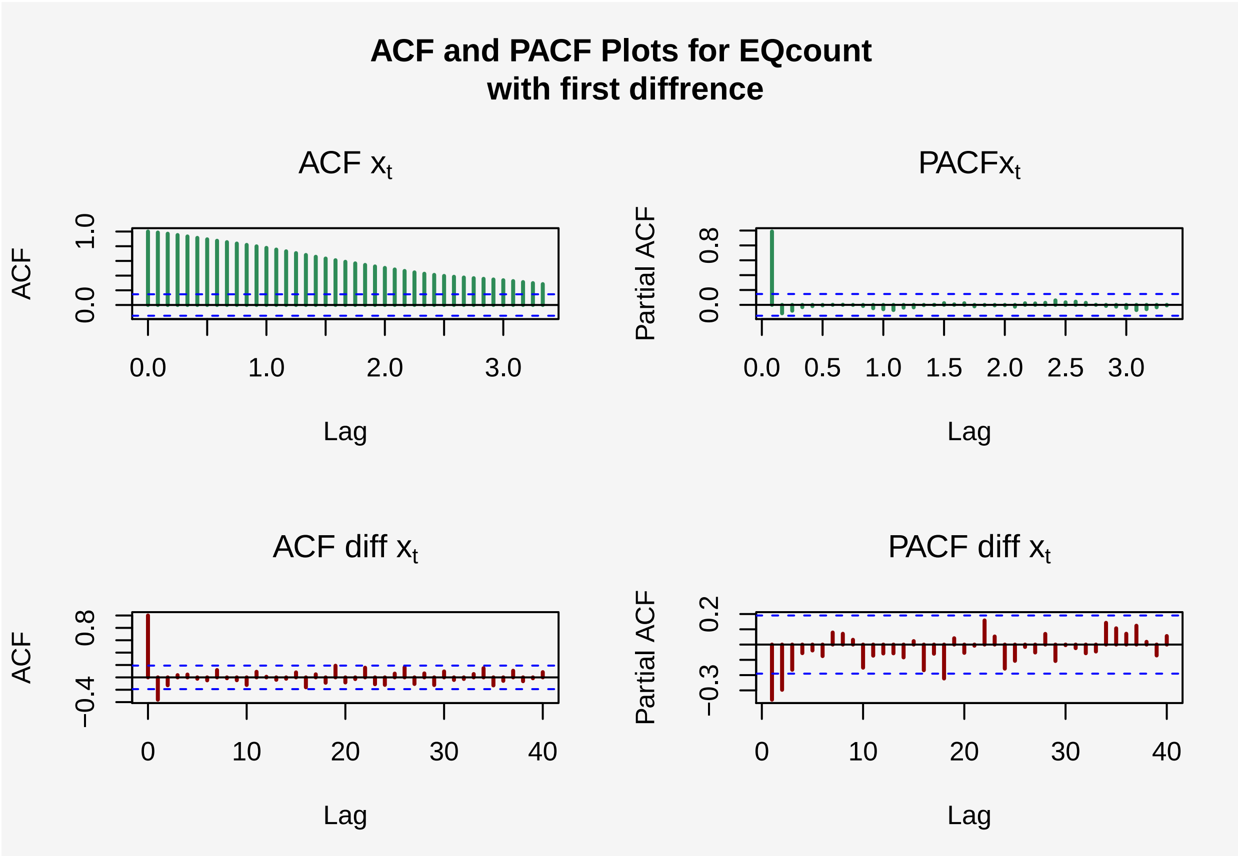
b)

Repeat step 1 for the following datasets: so2, EQcount, HCT in package astsa

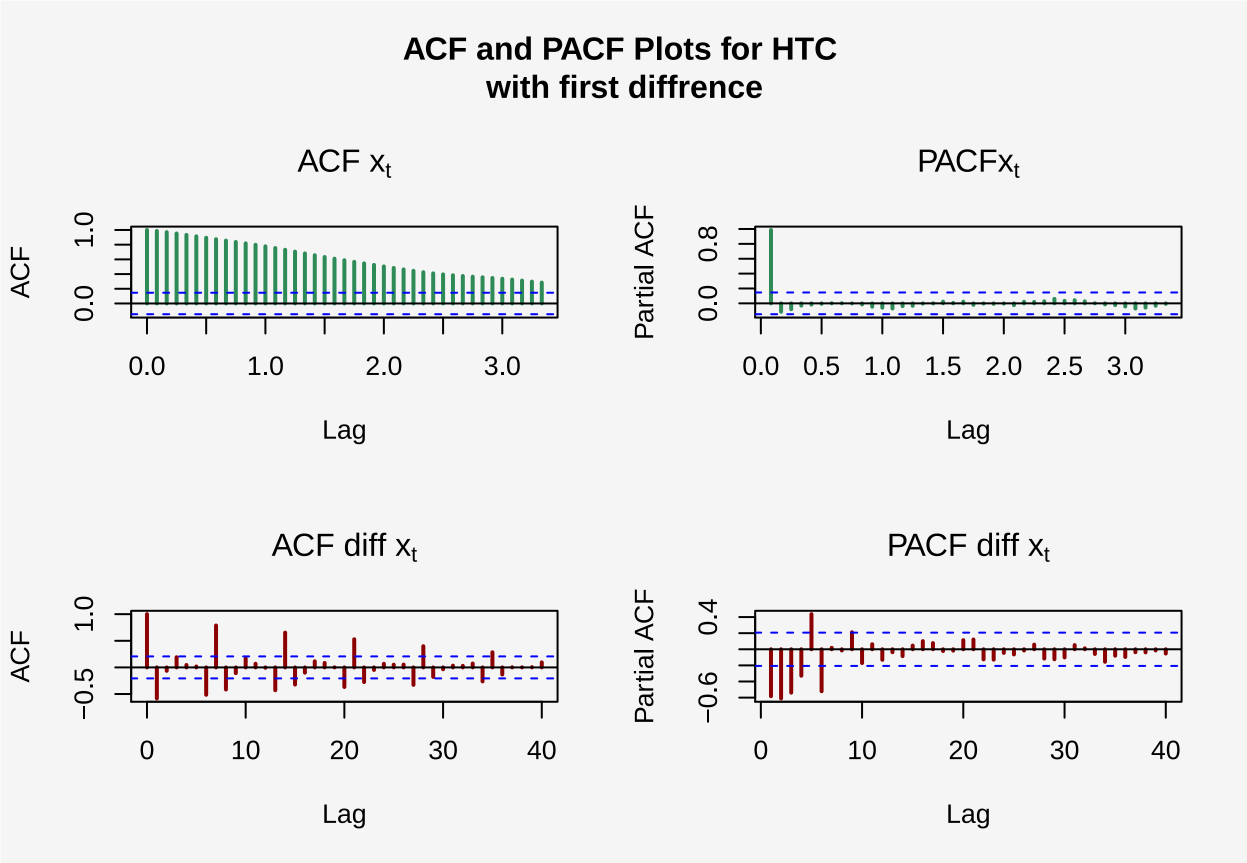
ACF-PACF PLots for so2 data



ACF-PACF PLots for EQcount data



ACF-PACF PLots for HCT data



Summary table for the ACF-PACF

	chicken	diff(chicken)	so2	diff(so2)
ACF	slow decay differencing needed	sesonal cycle patter presen	fast decay but need differencing	tails off after lag 0.02
PACF	seasonal pattern present	cut off after lag1	tails off quickly	tails off after 0.18

	EQcount	diff(EQcount)	HCT	diff(ECT)
ACF	tails off after lag 8	tails off after lag 1	tails off after lag 18	slow decay tails off after lag 1
PACF	the bars are in the boarders	tails off after lag1	tails off after lag7	tails off after lag5

From the ACF-PACF Plots and the above table we can propose the following models :

- For the chicken data Starting from the nonseasonal part we can suggest an AR(2) from ACF-PACF and for the seasonal s=12 an MA(1) The final model is a $SARIMA(2,1,0)x(1,0,0)_{12}$

- For the so2 data It's very hard to distinguiss a model but maybe and $ARMA(1,1,1)$ according to ACF-PACF plots of difference.
- For the EQcount From AFC of EQcount difference is needed and according to ACF of difference and MA(1).The final model is $ARIMA(0,1,1)$
- For the HCT According to ACF of HCT difference is needed.From PACF of difference we can suggest an $AR(5)$ and form ACF an $MA(1)$. The final model is $ARIMA(5,1,1)$

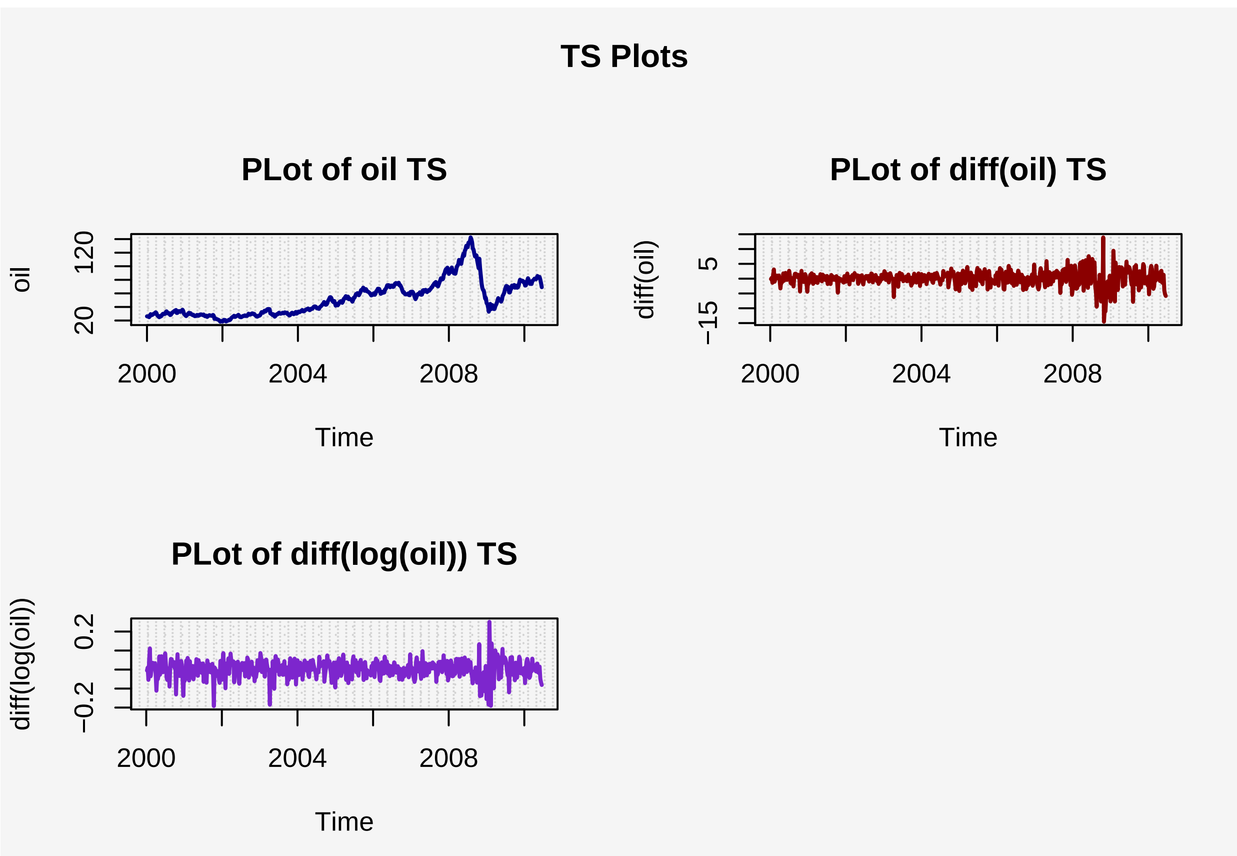
Assignment 3.Arima modeling cycle

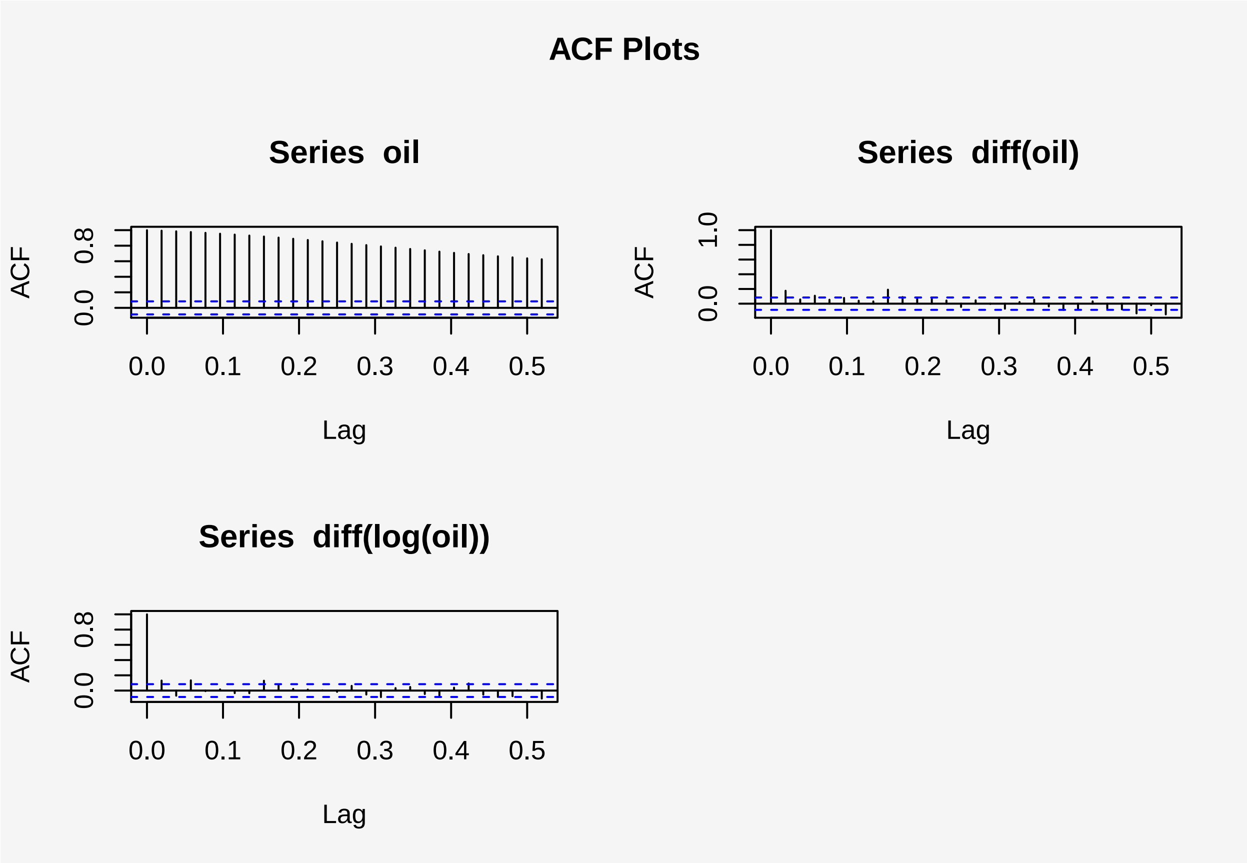
a)

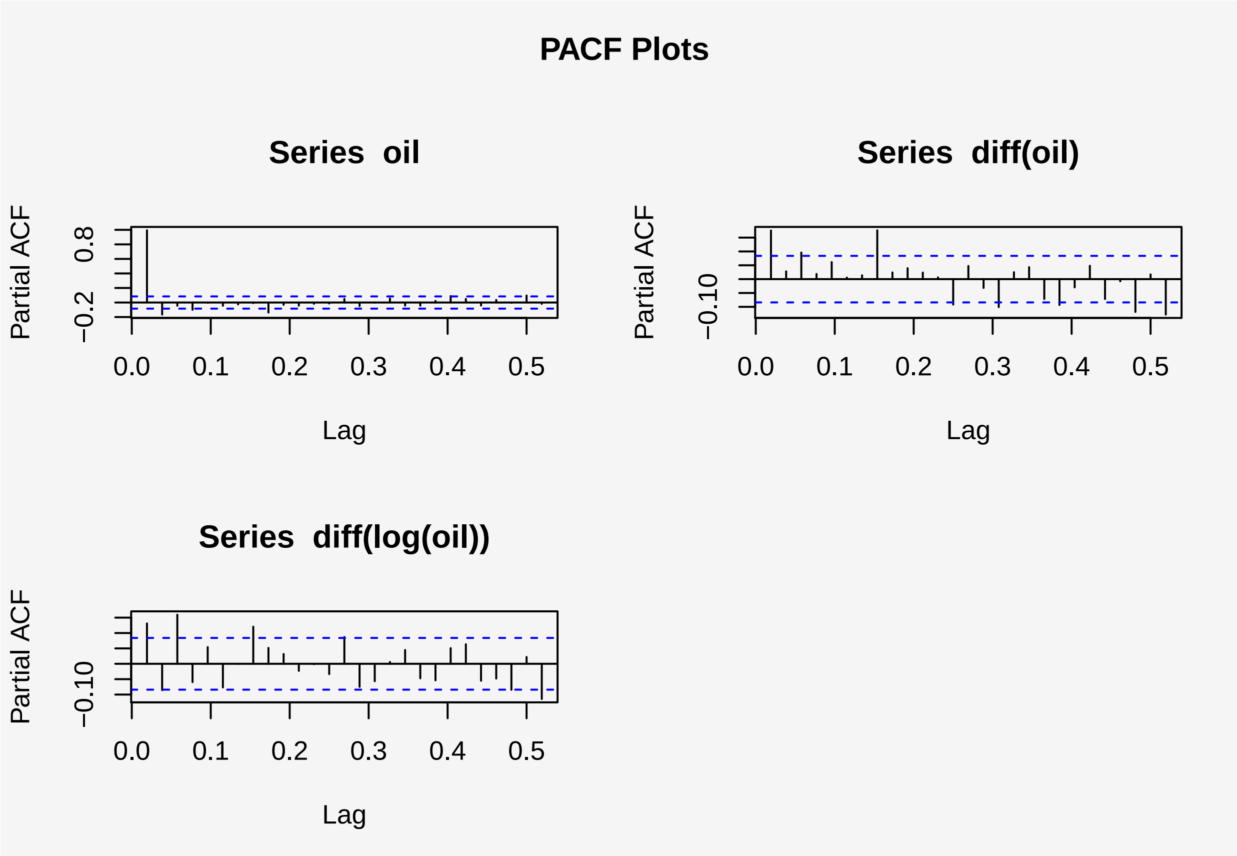
Find a suitable $ARIMA(p,d,q)$ model for the data set oil present in the library astsa. Your modeling should include the following steps in an appropriate order: visualization, unit root test, detrending by differencing (if necessary), transformations (if necessary), ACF and PACF plots when needed, EACF analysis, Q-Q plots, Box-Ljung test, ARIMA fit analysis, control of the parameter redundancy in the fitted model. When performing these steps, always have 2 tentative models at hand and select one of them in the end. Validate your choice by AIC and BIC and write down the equation of the selected model. Finally, perform forecasting of the model 20 observations ahead and provide a suitable plot showing the forecast and its uncertainty.

TS -ACF-PACF Plots

We start by making some diagnostic plots for the original time series ,the first diffeence and the log first diffrence.



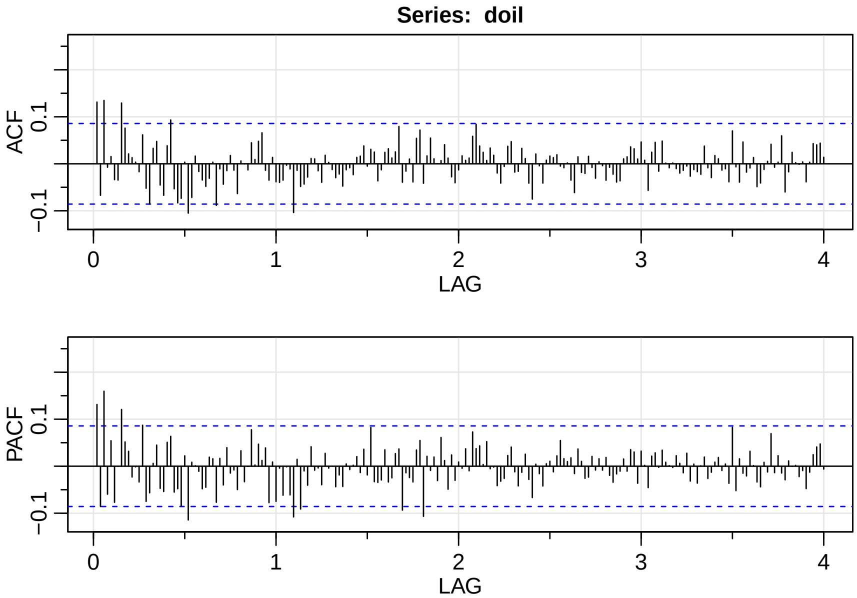

ACF Plots
Series oil
Series diff(oil)
Series diff(log(oil))
ACF
Lag
0.0
0.8
1.0
0.1
0.2
0.3
0.4
0.5



Fro the plots we conclude that working with the log of first diffrence of the original data seems reasonable because we have a stationaly process. Here we report the Dickey-Fuller test for stationarity and as we can see the p-value suggests that data are stationary.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  doil
## Dickey-Fuller = -6.3708, Lag order = 8, p-value = 0.01
## alternative hypothesis: stationary
```

Plot of the ACF-PACF for $\nabla log(x_t)$



Now we are going to use the eacf in order to identify best model combinations.

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x o x o o o o x o o o  o  o  o
## 1 x o x o o o o x o o o  o  o  o
## 2 x x x o o o o x o o o  o  o  o
## 3 x x x o o o o x o o o  o  o  o
## 4 x o x o o o o x o o o  o  o  o
## 5 x x x o x o o x o o o  o  o  o
## 6 o x x o x x o x o o o  o  o  x
## 7 o x x x x x x x o x o  o  o  o
```

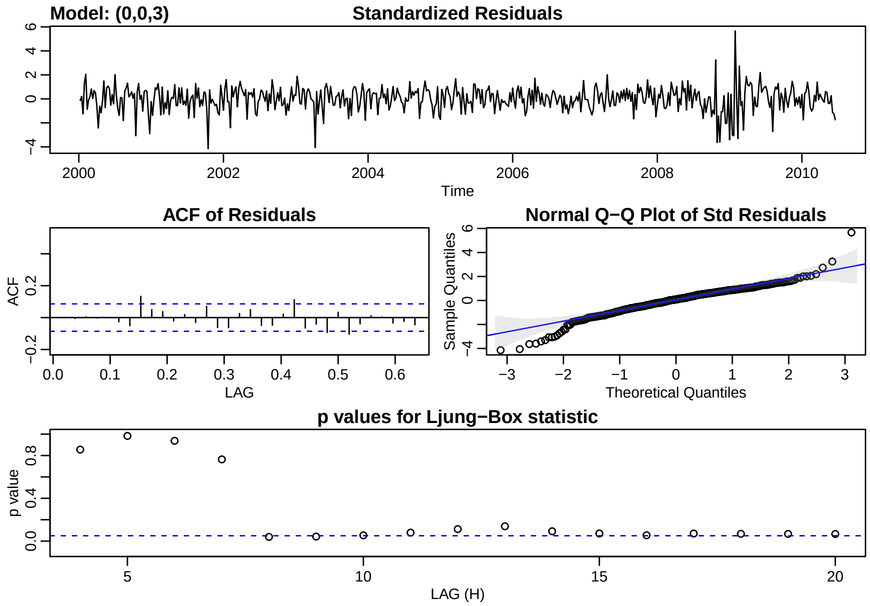
From the matrix we distinguiss 2 models 1.ARMA(0,3) and ARMA(1,1).We are going to investigate each seperately

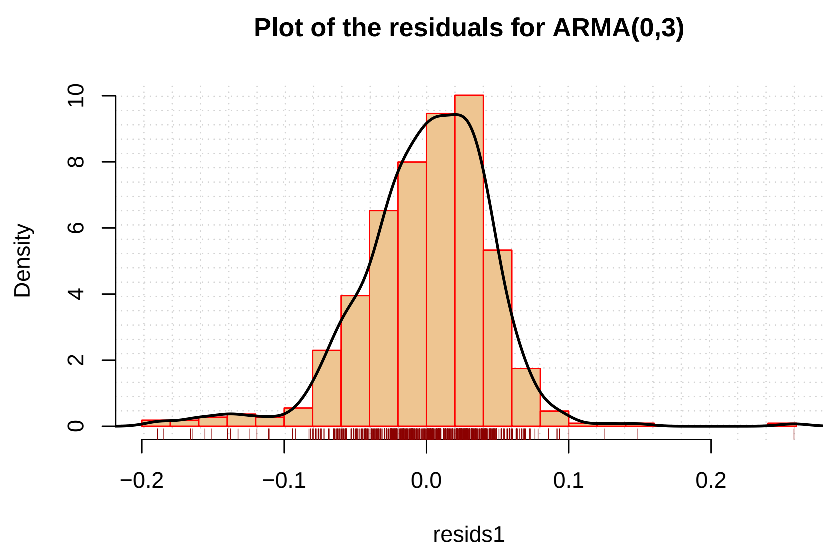
ARMA(0,3)

Diagnostic Plots for residuals

```
## initial  value -3.058495
## iter   2 value -3.086110
## iter   3 value -3.086980
## iter   4 value -3.087501
## iter   5 value -3.087521
## iter   6 value -3.087521
## iter   7 value -3.087522
## iter   8 value -3.087522
## iter   9 value -3.087522
## iter   9 value -3.087522
## iter   9 value -3.087522
```

```
## final  value -3.087522
## converged
## initial  value -3.087448
## iter   2 value -3.087448
## iter   3 value -3.087449
## iter   3 value -3.087449
## iter   3 value -3.087449
## final  value -3.087449
## converged
```





The Ljung-Box p-value is significant until lag 7 and from the Q-Q plot some sample residuals are not in the line of the theoretical ones as we see on the tail of the plot.The histogram of the residuals seems quit normal although the tails are very long.

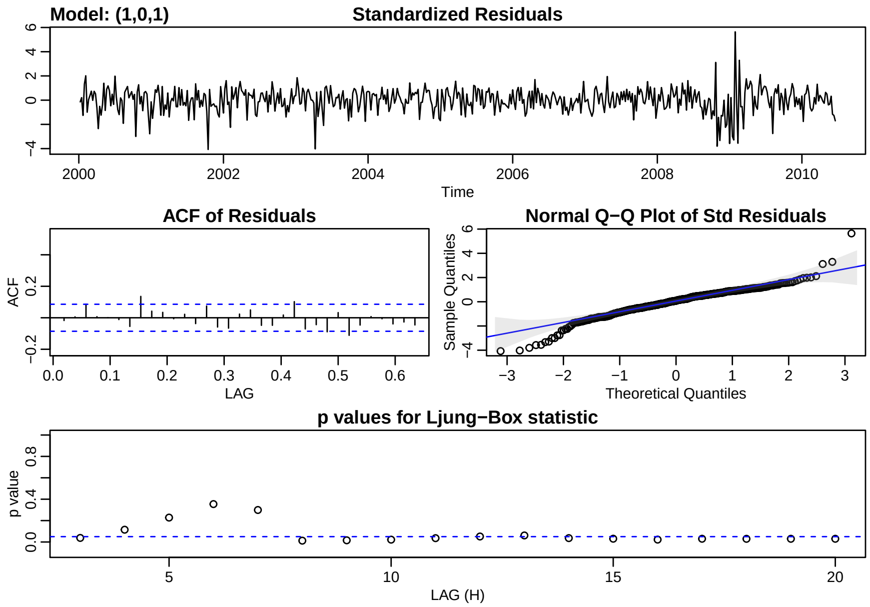
Next we perform runs test for independence

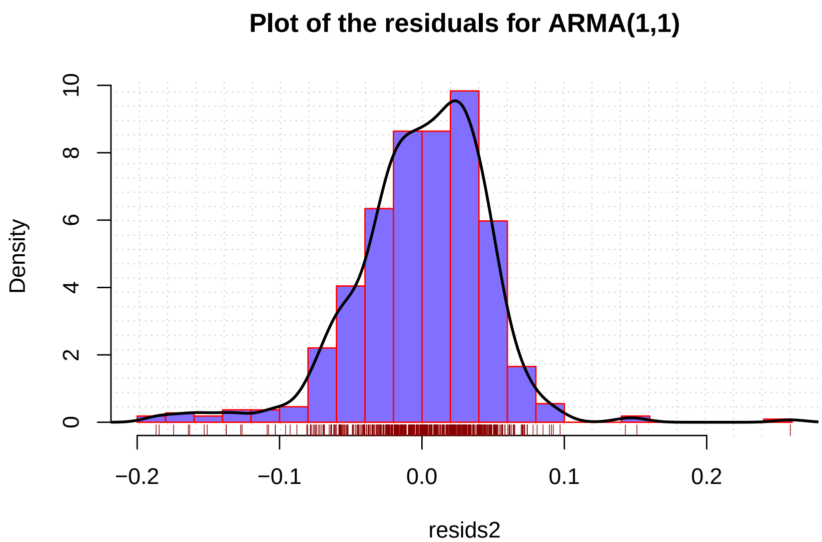
	x
pvalue	0.7940
observed.runs	267.0000
expected.runs	270.5147
n1	246.0000
n2	298.0000
k	0.0000
The p-value is q	uite high suggesting that the series is dependent

ARMA(1,1)

We proceed now for the next model

```
## initial  value -3.057594
## iter   2 value -3.061420
## iter   3 value -3.067360
## iter   4 value -3.067479
## iter   5 value -3.071834
## iter   6 value -3.074359
## iter   7 value -3.074843
## iter   8 value -3.076656
## iter   9 value -3.080467
## iter  10 value -3.081546
## iter  11 value -3.081603
## iter  12 value -3.081615
## iter  13 value -3.081642
## iter  14 value -3.081643
## iter  14 value -3.081643
## iter  14 value -3.081643
## final  value -3.081643
## converged
## initial  value -3.082345
## iter   2 value -3.082345
## iter   3 value -3.082346
## iter   4 value -3.082346
## iter   5 value -3.082346
## iter   5 value -3.082346
## iter   5 value -3.082346
## final  value -3.082346
## converged
```





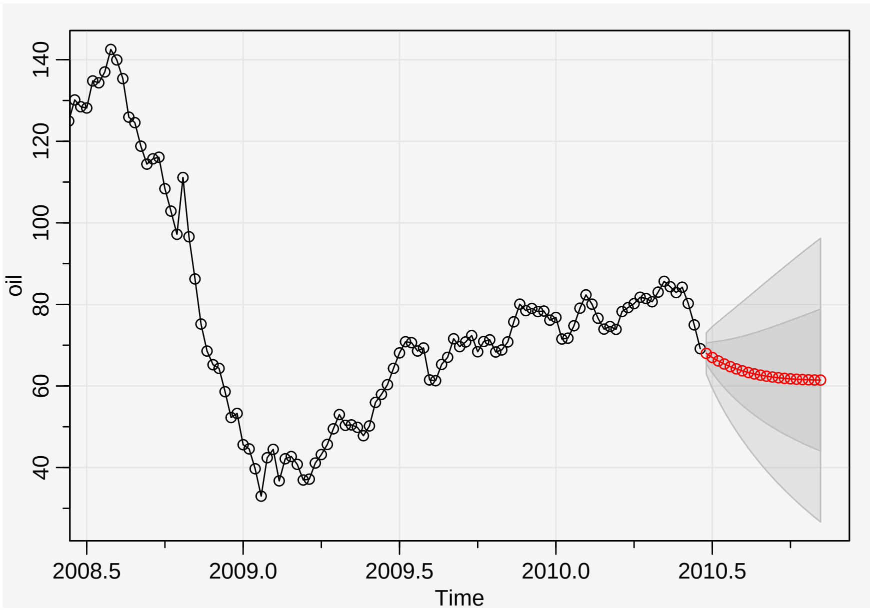
As we can see from the Q-Q plot ,the plot the Ljung-Box and the plot of the histogram of the residuals we obtain quite similar results with the previous model. Testing again for independence again the p-value suggests that we have dependence.

	x
pvalue	0.7880
observed.runs	275.0000
expected.runs	271.3787
n1	251.0000
n2	293.0000
k	0.0000

We proceed comparing the AIC and BIC of the 2 models.

	AIC	BIC
ARMA(0,3)	-3.318638	-3.279125
ARMA(1,1)	-3.312109	-3.280499

From the above table we conclude that the ARMA(1,1,1) seems to performe a little better and we use this for the predictions.

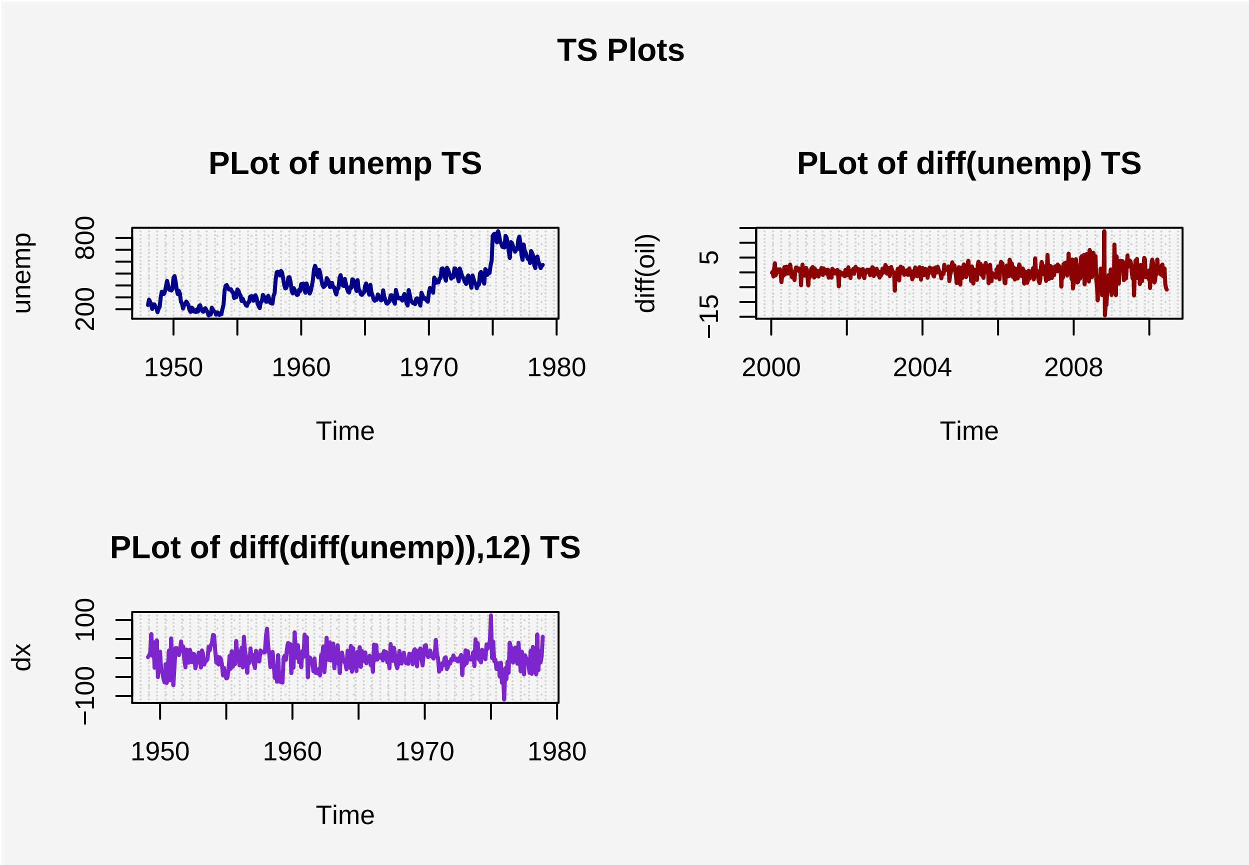


b)

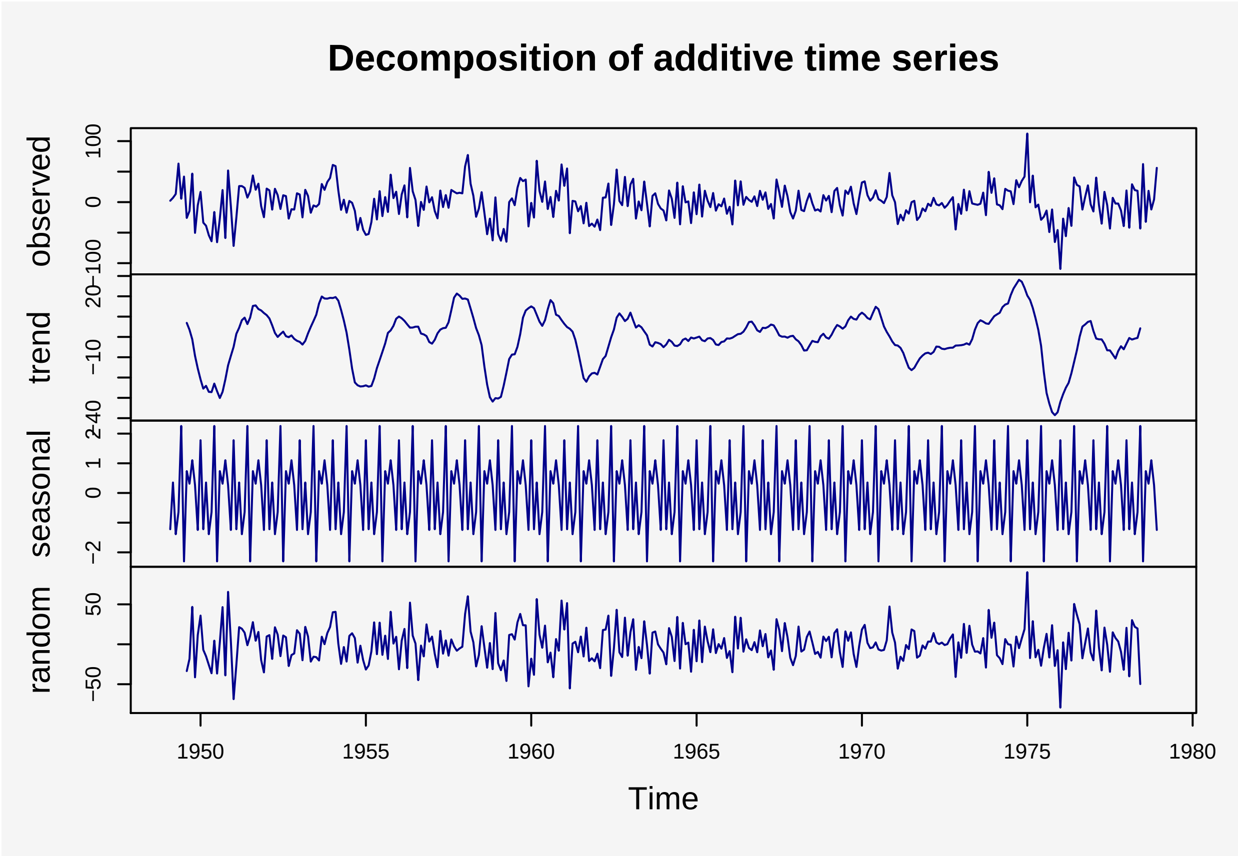
Find a suitable $ARIMA(p,d,q)x(P,D,Q)_s$ model for the data set unemp present in the library astsa. Your modeling should include the following steps in an appropriate order: visualization, detrending by differencing (if necessary), transformations (if necessary), ACF and PACF plots when needed, EACF analysis, Q-Q plots, Box-Ljung test, ARIMA fit analysis, control of the parameter redundancy in the fitted model. When performing these steps, always have 2 tentative models at hand and select one of them in the end. Validate your choice by AIC and BIC and write down the equation of the selected model (write in the backshift operator notation without expanding the brackets). Finally, perform forecasting of the model 20 observations ahead and provide a suitable plot showing the forecast and its uncertainty.

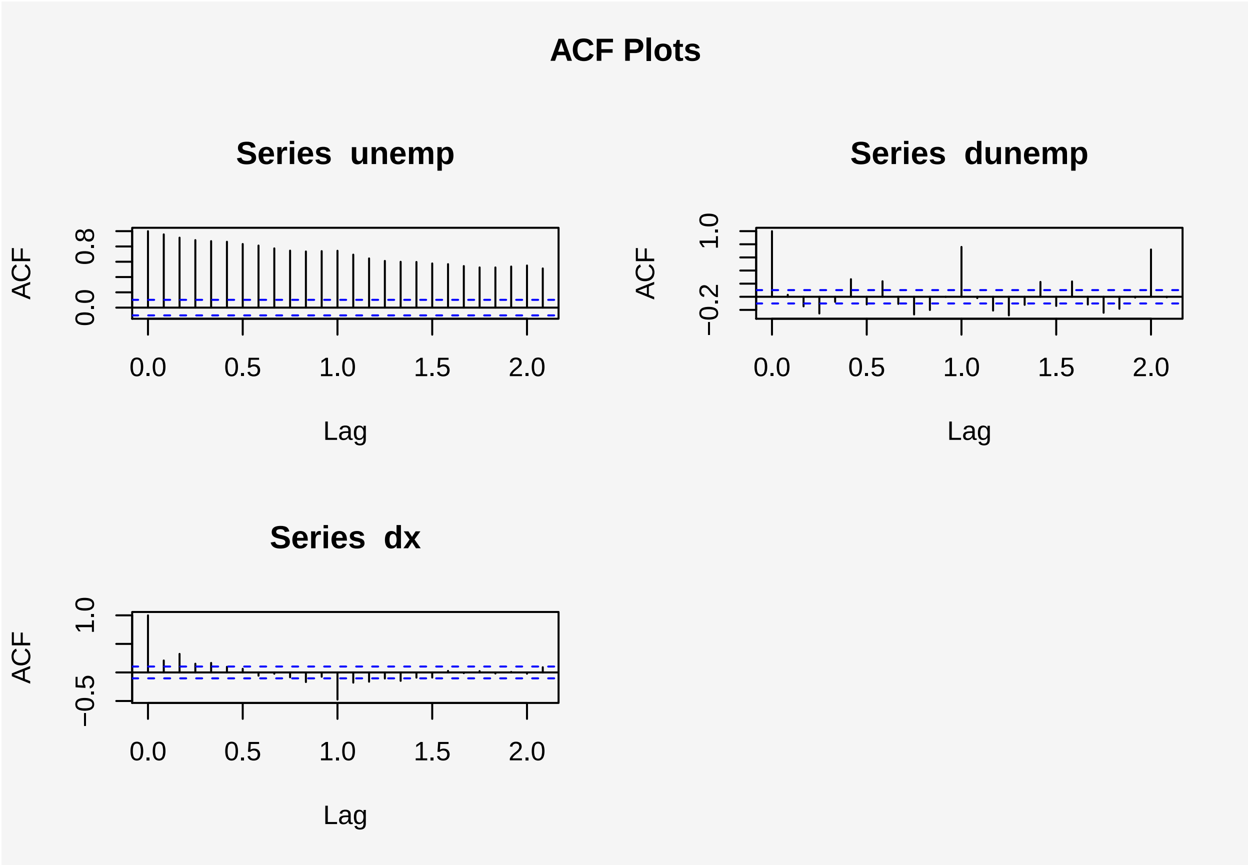
TS -ACF-PACF Plots

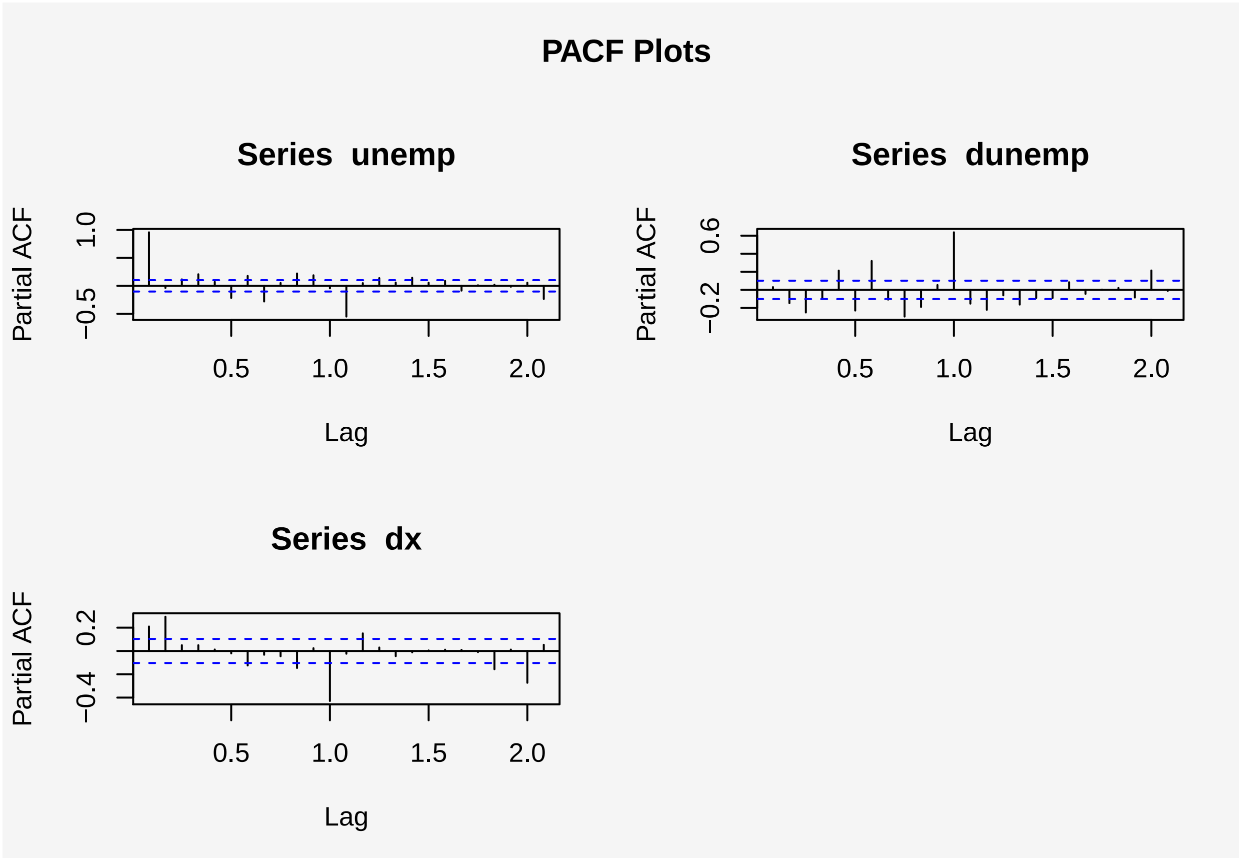
We start by making some diagnostic plots for the original time series ,the first diffeence and the log first diffrence.



We report also the decomposition of the $\nabla^{12}\nabla unemp$


Decomposition of additive time series
observed
trend
seasonal
random
100
0
−100
20
−10
−40
2
1
0
−2
50
−50
1950
1955
1960
1965
1970
1975
1980
Time


ACF Plots
Series unemp
ACF
0.0
0.8
0.0
0.5
1.0
1.5
2.0
Lag
Series dunemp
ACF
−0.2
1.0
0.0
0.5
1.0
1.5
2.0
Lag
Series dx
ACF
−0.5
1.0
0.0
0.5
1.0
1.5
2.0
Lag



From the plots we conclude that is suggesting to work with the $\nabla^{12}\nabla unemp$ transformation.Also the plot suggest an $ARMA(1,1,1)$ for the first difference data. Here we report the Dickey-Fuller test for stationarity and as we can see the p-value suggests that data are non stationary.

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dx
## Dickey-Fuller = -6.171, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Now we are going to use the eacf in order to identify best model combinations.

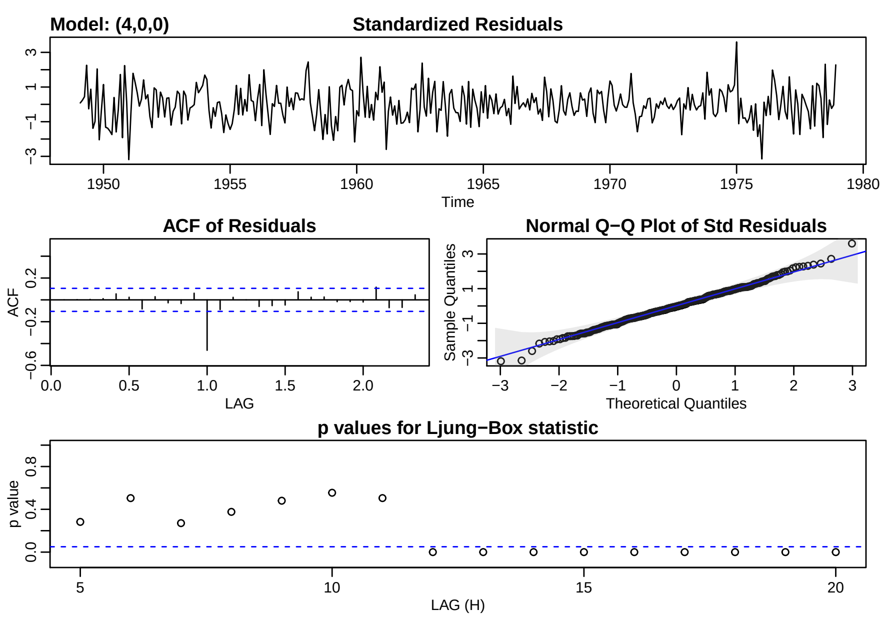
```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o o o o x o  x  x  x
## 1 x x x o o o o o o x o  x  x  o
## 2 x x o o o o o o o o o  x  o  x
## 3 x x o o o o o o o o o  x  o  x
## 4 x x o o o o o o o o o  x  o  o
## 5 x o x o x o o o o o o  x  x  o
## 6 x x x x x o o o o o o  x  o  o
## 7 x x x x x o o o o o o  x  o  o
```

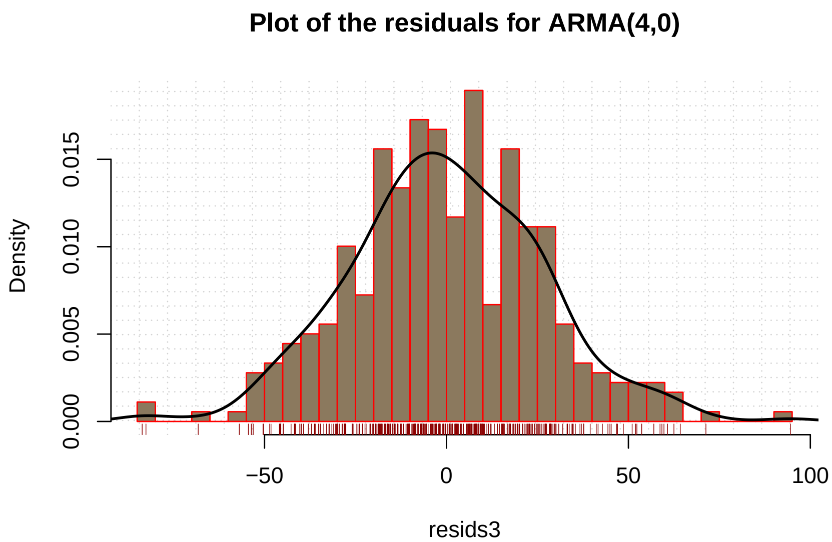
From the matrix we distinguiss 2 models 1.ARMA(4,0) and ARMA(2,2).We are going to investigate each seperately

ARMA(4,0)

Diagnostic Plots for residuals

```
## initial  value 3.336140
## iter   2 value 3.298515
## iter   3 value 3.265469
## iter   4 value 3.265171
## iter   5 value 3.265162
## iter   6 value 3.265161
## iter   7 value 3.265157
## iter   8 value 3.265157
## iter   9 value 3.265156
## iter   9 value 3.265156
## iter   9 value 3.265156
## final  value 3.265156
## converged
## initial  value 3.267503
## iter   2 value 3.267493
## iter   3 value 3.267473
## iter   4 value 3.267464
## iter   5 value 3.267456
## iter   6 value 3.267454
## iter   7 value 3.267454
## iter   7 value 3.267454
## iter   7 value 3.267454
## final  value 3.267454
## converged
```





The Ljung-Box p-value is significant until lag 11 and from the Q-Q plot some sample residuals are not in the line of the theoretical ones as we see on the tail of the plot.The histogram of the residuals seems quit normal.

Next we perform runs test for independence

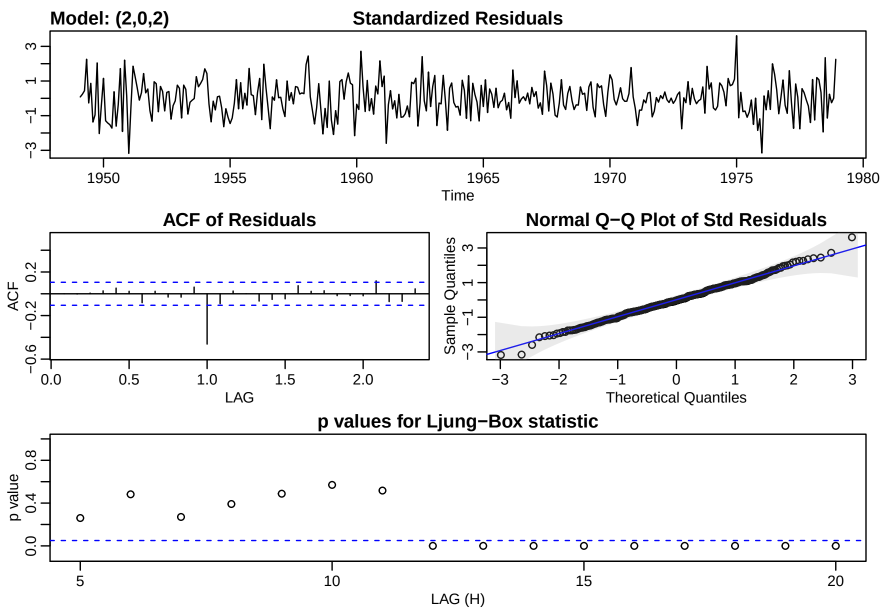
	x
pvalue	0.9800
observed.runs	181.0000
expected.runs	180.2646
n1	186.0000
n2	173.0000
k	0.0000

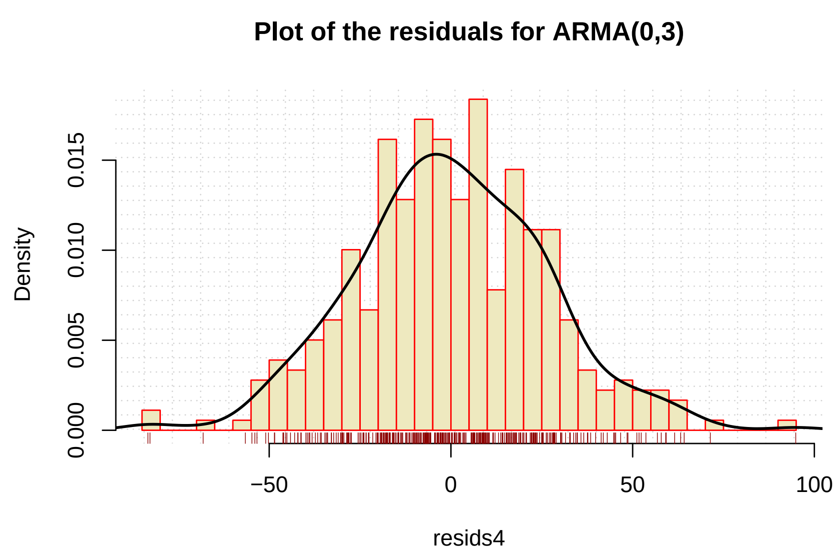
ARMA(2,2)

Diagnostic Plots for residuals

```
## initial  value 3.340771
## iter   2 value 3.295757
## iter   3 value 3.279401
## iter   4 value 3.276527
## iter   5 value 3.275399
## iter   6 value 3.271435
## iter   7 value 3.270544
## iter   8 value 3.270026
## iter   9 value 3.269973
## iter  10 value 3.269972
## iter  11 value 3.269972
## iter  12 value 3.269971
## iter  12 value 3.269971
## final  value 3.269971
## converged
```

```
## initial  value 3.267802
## iter   2 value 3.267800
## iter   3 value 3.267796
## iter   4 value 3.267792
## iter   5 value 3.267791
## iter   6 value 3.267791
## iter   7 value 3.267791
## iter   8 value 3.267791
## iter   9 value 3.267791
## iter  10 value 3.267791
## iter  11 value 3.267791
## iter  12 value 3.267791
## iter  12 value 3.267791
## final  value 3.267791
## converged
```





The results are similar with the previous model

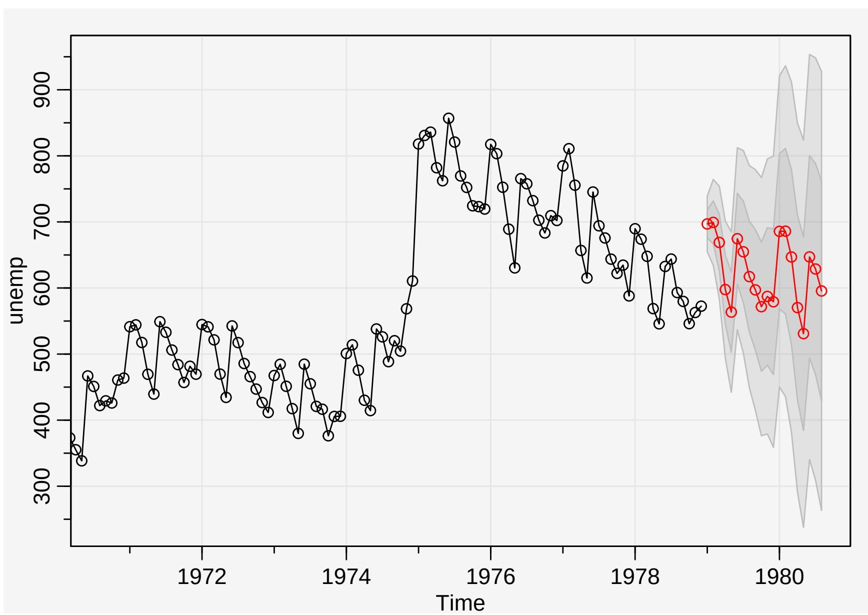
Next we perform runs test for independence

	x
pvalue	0.9900
observed.runs	181.0000
expected.runs	180.3872
n1	184.0000
n2	175.0000
k	0.0000

We proceed comparing the AIC and BIC of the 2 models.

	AIC	BIC
ARMA(4,0)	9.406211	9.471113
ARMA(2,2)	9.406885	9.471787

From the above table we conclude that the $SARMA(4,1,0)_{12}$ seems to performe a little better and we use this for the predictions. The final model is $SARMA(1,1,1)x(4,1,0)_{12}$ and we use this for the prediction shown below.



Appendix

```r
## ----setup,
## include=FALSE--------------------------------------------
## knitr::opts_chunk$set(echo = F, message = F,warning =
## F)
# Libraries
# ------------------------------------------------------------
library(astsa)
library(ggplot2)
library(knitr)
# ----------------------------------------------------------------------
# Assignment 1
# ----------------------------------------------------------
# a) ------------------------------------
set.seed(12345)
# sim AR(3) n=1000
AR3.sim = arima.sim(list(ar = c(0.8, -0.2, 0.1)), n = 1000)
# bind ts for up to 3 lags
data1 = ts.intersect(x = AR3.sim, x1 = lag(AR3.sim, -1),
    x2 = lag(AR3.sim, -2), x3 = lag(AR3.sim, -3), dframe = T)
# linear regressions
res1 = lm(x ~ x1 + x2, data = data1)
res2 = lm(x3 ~ x1 + x2, data = data1)
# calculate residuals
resids1 = residuals(res1)
resids2 = residuals(res2)
# calculate correlation for the residuals
estimatedCorr = cor(cbind(resids1, resids2))
# theoretical pacf
theoreticalPacf = ARMAacf(ar = c(0.8, -0.2, 0.1), pacf = T)
simulatedPacf = pacf(AR3.sim)
# make a table with the corr and pacf
sum_tab = cbind(estimatedCorr[1, 2], simulatedPacf[3][[1]],
    theoreticalPacf[3])
colnames(sum_tab) = c("Est.Corr", "Sim.PACF", "Theo.PACF")
## ----------------------------
kable(sum_tab)
# ----------------------------
## --------------------------- b)
## ------------------------------------ simulate AR(2)
## n=100
AR2.sim = arima.sim(list(ar = c(0.8, 0.1)), n = 100)
# perform each method
AR_yw = ar(AR2.sim, order.max = 2, aic = F)  # Yule-Walker
AR_ols = ar(AR2.sim, method = "ols", order.max = 2, aic = F)  # OLS
AR_mle = ar(AR2.sim, method = "mle", order.max = 2, aic = F)  # ML
# make a table with the coefficients
d = rbind(AR_yw$ar, AR_ols$ar, AR_mle$ar, c(0.8, 0.1))
rownames(d) = c("YW", "OLS", "MLE", "TRUE")
colnames(d) = c("phi1", "phi2")
kable(d)
## -------------------------------- check iF phi2 in CI
```

```r
## for ML estimate compute the CI
lower_CI = AR_mle$ar[2] - sqrt(AR_mle$asy.var.coef[2, 2]) *
    1.96  # lower limit
upper_CI = AR_mle$ar[2] + sqrt(AR_mle$asy.var.coef[2, 2]) *
    1.96  # upper limit
# check if the phi2=0.1 in the CI
cat("The esitmated interval is :", c(lower_CI, upper_CI),
    "\n")
cat("The value of phi_2 is within the estimated interval?\n")
0.1 %in% round(seq(as.numeric(lower_CI[1]), as.numeric(upper_CI[1]),
    length.out = 1000), 2)  # returns T,F
# PACF = ARMAacf(ar=c(0.8,0.1), pacf=TRUE) ; PACF #
# theoretical PACF
# --------------------------------------------------------------------------
# c) ------------------------------------ simulate
# ARIMA(0,0,1)x(0,0,1)12 n=200
seasonal.sim = arima.sim(list(order = c(0, 0, 13), ma = c(0.3,
    rep(0, 10), 0.6, (0.6 * 0.3))), n = 200)
# plot of the simulated PACF and ACF for simulation
par(mfrow = c(2, 2), bg = "whitesmoke")
acf(seasonal.sim, main = "Sample ACF", panel.first = grid(25,
    25), lag.max = 40)
pacf(seasonal.sim, main = "Sample PACF", panel.first = grid(25,
    25), lag.max = 40)
# compute theoretical ACF and PACF
ACF = ARMAacf(ma = c(0.3, rep(0, 10), 0.6, (0.6 * 0.3)),
    lag.max = 40)
PACF = ARMAacf(ma = c(0.3, rep(0, 10), 0.6, (0.6 * 0.3)),
    pacf = TRUE, lag.max = 40)
# plot of the theoretical ACF and PACF
plot(ACF, type = "h", xlab = "Lag", ylim = c(-0.4, 0.8),
    main = "Theoretical ACF", panel.first = grid(25, 25))
abline(h = 0)
plot(PACF, type = "h", xlab = "Lag", ylim = c(-0.4, 0.8),
    main = "Theoretical PACF", panel.first = grid(25, 25))
abline(h = 0)
# --------------------------------------------------------------------------
# d) ------------------------------------
# using simulation ----------------
# simulate ARIMA(0,0,1)x(0,0,1)12
seasonal.sim1 = arima.sim(list(order = c(0, 0, 13), ma = c(0.3,
    rep(0, 10), 0.6, (0.6 * 0.3))), n = 200)
# forecast using the sarima.for and auto plot
# fore.sar=sarima.for(seasonal.sim1,n.ahead=30,p=0,d=0,q=1,P=0,D=0,Q=1,S=12)
# the same results but using predict forecast using the
# predict and plot of the forecasts
fore = predict(arima(seasonal.sim1, order = c(0, 0, 1),
    seasonal = list(order = c(0, 0, 1), period = 12)), n.ahead = 30)
# plot of ts and predictions and band
cols = c("mediumpurple3", "darkslateblue")
par(mfrow = c(1, 1), oma = c(0, 0, 3, 0))
ts.plot(seasonal.sim1, fore$pred, col = cols, lwd = 3)
U = fore$pred + fore$se
```

```r
L = fore$pred - fore$se
xx = c(time(U), rev(time(U)))
yy = c(L, rev(U))
polygon(xx, yy, border = 8, col = gray(0.6, alpha = 0.2))
points(fore$pred, pch = 20, col = "red")
legend("topright", legend = c("Sim TS", "Predicted 30"),
    col = c(cols[1], cols[2]), lty = 1, lwd = 2)
title(expression("ARMA(0,0,1)x(0,0,1)"[12]))
# using the kernelab -------------- kenel fit not import
# the kernelab because of the predict
kernel_fit = kernlab::gausspr(x = c(1:200), seasonal.sim1)
# make predictions
kernels_preds = kernlab::predict(kernel_fit, c(200:230))
# plot the ts and predictions
cols1 = c("blue2", "black")
par(mfrow = c(1, 1), oma = c(0, 0, 3, 0))
ts.plot(seasonal.sim1, ts(kernels_preds, start = 200, end = 230),
    col = cols1, lwd = 3)
points(ts(kernels_preds, start = 200, end = 230), pch = 20,
    col = "azure3", cex = 0.2)
legend("bottomright", legend = c("Sim TS", "Predicted 30"),
    col = c(cols1[1], cols[2]), lty = 1, lwd = 2)
title(expression("ARMA(0,0,1)x(0,0,1)"[12] * " kernelab"))
# ------------------------------------------------------------------------
# e) ------------------------------------ simulate
# ARMA(1,1) n=50
arma.sim <- arima.sim(list(order = c(1, 0, 1), ar = 0.7,
    ma = 0.5), n = 50)
# make predictions with the sarima.for
# sarima.for(arma.sim[1:40],n.ahead=10,1,0,1,0,0,0,0,no.constant
# = T)
# the same results but using predict forecast using the
# predict and plot of the forecasts
fore1 = predict(arima(arma.sim[1:40], order = c(1, 0, 1),
    include.mean = F), n.ahead = 10)
col1 = "forestgreen"
col2 = "magenta1"
col3 = "orange1"
ts.plot(as.ts(arma.sim[1:41]), fore1$pred, col = c(col1,
    col2), lwd = 2, main = "ARMA(1,1) with predictions")
lines(ts(arma.sim[41:50], start = 41, end = 50), col = col3,
    lwd = 2, type = "o")
U1 = fore1$pred + fore1$se
L1 = fore1$pred - fore1$se
xx1 = c(time(U1), rev(time(U1)))
yy1 = c(L1, rev(U1))
polygon(xx1, yy1, border = 8, col = gray(0.6, alpha = 0.2))
points(fore1$pred, pch = 20, col = "black", cex = 0.5)
legend("topleft", legend = c("First 40 values", "Predicted 10",
    "True 10"), col = c(col1, col2, col3), lty = 1, lwd = 2)
## ------------------- Note
require(dplyr)  # we load the library here beacaus it masks the lag and had problems in Ass1.a
# count how many points are not in the band
```

```
mat <- cbind(as.vector(U1), as.vector(L1), as.vector(arma.sim[41:50]))
mat <- as.data.frame(mat)  # ; mat
mat = mat %>% mutate(res = ifelse(((mat$V3 > mat$V1) | (mat$V3 <
    mat$V2)), 1, 0))
cat("The number of the values outside the prediction band is:",
    sum(mat$res))
# --------------------------------------------------------------------------
# Assignment 2
# ------------------------------------------------------------
# data chicken -------------------
data(chicken)
# girst diffrence
diff.xt <- diff(chicken)
my_colors = c("seagreen4", "red4")
# ACF and PACF plots
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(chicken, 60, col = my_colors[1], main = expression("ACF x"[t]),
    lwd = 2)
pacf(chicken, 60, col = my_colors[1], main = expression("PACFx"[t]),
    lwd = 2)
acf(diff.xt, 60, col = my_colors[2], main = expression("ACF diff x"[t]),
    lwd = 2)
pacf(diff.xt, 60, col = my_colors[2], main = expression("PACF diff x"[t]),
    lwd = 2)
title("\nACF and PACF Plots for chicken \nwith first diffrences",
    outer = TRUE)
# mtext(c('ACF~PACF Plots chicken', 'ACF~PACF PLots for
# diff(chicken)'), side = 3, line = c(-2,-18), outer =
# TRUE)
# -------------------------------
# data so2 ----------------------
data(so2)
# first diffrence
diff.xt <- diff(so2)
# ACF and PACF plots
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(chicken, 40, col = my_colors[1], main = expression("ACF x"[t]),
    lwd = 2)
pacf(chicken, 40, col = my_colors[1], main = expression("PACFx"[t]),
    lwd = 2)
acf(diff.xt, 40, col = my_colors[2], main = expression("ACF diff x"[t]),
    lwd = 2)
pacf(diff.xt, 40, col = my_colors[2], main = expression("PACF diff x"[t]),
    lwd = 2)
title("\nACF and PACF Plots for so2 \nwith first diffrence",
    outer = TRUE)
# mtext(c('ACF~PACF Plots so2', 'ACF~PACF PLots for
# diff(so2)'), side = 3, line=c(0,-19),outer = T)
# -------------------------------
# data EQcount ------------------
data("EQcount")
```

```r
# first diffrence
diff.xt <- diff(EQcount)
# plots
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(chicken, 40, col = my_colors[1], main = expression("ACF x"[t]),
    lwd = 2)
pacf(chicken, 40, col = my_colors[1], main = expression("PACFx"[t]),
    lwd = 2)
acf(diff.xt, 40, col = my_colors[2], main = expression("ACF diff x"[t]),
    lwd = 2)
pacf(diff.xt, 40, col = my_colors[2], main = expression("PACF diff x"[t]),
    lwd = 2)
title("\nACF and PACF Plots for EQcount \nwith first diffrence",
    outer = TRUE)
# mtext(c('ACF~PACF Plots EQcount', 'ACF~PACF PLots for
# diff(EQcount)'), side = 3, line = c(-2,-18), outer =
# TRUE)
# --------------------------------
# HCT ----------------------------
data(HCT)
# first diffrence
diff.xt <- diff(HCT)
# plots
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(chicken, 40, col = my_colors[1], main = expression("ACF x"[t]),
    lwd = 2)
pacf(chicken, 40, col = my_colors[1], main = expression("PACFx"[t]),
    lwd = 2)
acf(diff.xt, 40, col = my_colors[2], main = expression("ACF diff x"[t]),
    lwd = 2)
pacf(diff.xt, 40, col = my_colors[2], main = expression("PACF diff x"[t]),
    lwd = 2)
title("\nACF and PACF Plots for HTC \nwith first diffrence",
    outer = TRUE)
# mtext(c('ACF~PACF Plots EQcount', 'ACF~PACF PLots for
# diff(EQcount)'), side = 3, line = c(-2,-35), outer =
# TRUE)
# ----------------------------------
# Assignment 3
# -----------------------------------------------------------
# a) ------------------------------------
data(oil)
# plots
par(mfrow = c(2, 2), oma = c(0, 0, 3, 0), bg = "whitesmoke")
plot.ts(oil, col = "darkblue", lwd = 2, panel.first = grid(25,
    25), main = "PLot of oil TS")
plot.ts(diff(oil), col = "darkred", lwd = 2, panel.first = grid(25,
    25), main = "PLot of diff(oil) TS")
plot.ts(diff(log(oil)), col = "purple3", lwd = 2, panel.first = grid(25,
    25), main = "PLot of diff(log(oil)) TS")
title("TS Plots", outer = T)
```

```r
## -----------------------------------------------
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(oil)
acf(diff(oil))
acf(diff(log(oil)))
title("ACF Plots", outer = T)
## -----------------------------------------------
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
pacf(oil)
pacf(diff(oil))
pacf(diff(log(oil)))
title("PACF Plots", outer = T)
## ----------------------------------------------- we work
## with diff(log(oil)) for the analysis
doil = diff(log(oil))
# test the p-value
tseries::adf.test(doil)
plots = acf2(doil)
## ------------------------------------------------------------------------
## eacf test
TSA::eacf(doil)  # 2 choices AR(0,3) ARMA(1,1)
# Start with ARMA(0,3) ---------
# fit the model
arma.fit = sarima(doil, p = 0, d = 0, q = 3, details = T)
# tsdiag(arma.fit$fit) diagnostic plots
resids1 = residuals(arma.fit$fit)  # calculate residuals
# plot the residuals withe the forecast package
# resids1%>%forecast::ggtsdisplay(main='TS-ACF-PACF for
# residuals', col=sample(colors(),1),theme=theme_gray())
# forecast::gghistogram(resids1,add.normal = T,add.kde =
# T) # plot of residuals with gghistogram plot the
# histtogram of the residuals with basic
hist(resids1, 30, col = sample(colors(), 1), border = "red",
    panel.first = grid(25, 25), freq = F, main = "Plot of the residuals for ARMA(0,3)")
lines(density(resids1), lwd = 2)
rug(resids1, col = "red4")
## ---------------------------------------------- Test the
## independence of a sequence of random variables
kable(unlist(TSA::runs(resids1)))
# Then with ARMA(1,1) --------- fit model
arma.fit1 = sarima(doil, 1, 0, 1)
resids2 = residuals(arma.fit1$fit)
# plot of residuals with the forecast package
# resids2%>%forecast::ggtsdisplay(main='TS-ACF-PACF for
# residuals', col=sample(colors(),1),theme=theme_gray())
# plot of the histogram with the basic
hist(resids2, 30, col = sample(colors(), 1), border = "red",
    panel.first = grid(25, 25), freq = F, main = "Plot of the residuals for ARMA(1,1)")
lines(density(resids2), lwd = 2)
rug(resids2, col = "red4")
## ---------------------------------
```

```r
kable(unlist(TSA::runs(resids2)))
## ------------------------------------- ckeck the AIC
## and BIC for each model
AIC_BIC_mat = cbind(c(arma.fit$AIC, arma.fit1$AIC), c(arma.fit$BIC,
    arma.fit1$BIC))  # ; AIC_mat
colnames(AIC_BIC_mat) = c("AIC", "BIC")
rownames(AIC_BIC_mat) = c("ARMA(0,3)", "ARMA(1,1)")
kable(AIC_BIC_mat)
# we choose the ARMA(1,1) and make predictions
par(mfrow = c(1, 1), bg = "whitesmoke")
S1 <- sarima.for(oil, n.ahead = 20, p = 1, d = 1, q = 1,
    P = 0, D = 0, Q = 0, S = 0)
# the same results but using predict
# fore2=predict(arima(oil,order=c(1,0,1),include.mean =
# F),n.ahead = 20)
# cols4=sample(colors(),2)
# ts.plot(oil,fore2$pred,col=cols4,
# lwd=3,main='ARMA(1,1) with predictions') U2 =
# fore2$pred+fore2$se; L2 = fore2$pred-fore2$se xx2 =
# c(time(U2), rev(time(U2))); yy2 = c(L2, rev(U2))
# polygon(xx2, yy2, border = 8, col = gray(.6, alpha =
# .2)) points(fore2$pred, pch=19, col=2,cex=0.3)
# legend('topleft',legend=c('Oil TS','Predicted Oil 20
# ahead'), col=cols1,lty=1,lwd=2)
# b) ------------------------------------
## -------------------------------------
data(unemp)
# first difference
dunemp = diff(unemp)
# remove seasonal by 12 difference
dx = diff(dunemp, 12)
# plots
par(mfrow = c(2, 2), oma = c(0, 0, 3, 0), bg = "whitesmoke")
plot.ts(unemp, col = "darkblue", lwd = 2, panel.first = grid(25,
    25), main = "PLot of unemp TS")
plot.ts(diff(oil), col = "darkred", lwd = 2, panel.first = grid(25,
    25), main = "PLot of diff(unemp) TS")
plot.ts(dx, col = "purple3", lwd = 2, panel.first = grid(25,
    25), main = "PLot of diff(diff(unemp)),12) TS")
title("TS Plots ", outer = T)
## ----------------------------------------------------------------------
par(mfrow = c(1, 1), bg = "whitesmoke")
plot(decompose(dx), col = "darkblue")
## ----------------------------------------------------------------------
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
acf(unemp)
acf(dunemp)
acf(dx)
title("ACF Plots", outer = T)
## ----------------------------------------------------------------------
par(mfrow = c(2, 2), bg = "whitesmoke", oma = c(0, 0, 3,
    0))
```

```r
pacf(unemp)
pacf(dunemp)
pacf(dx)
title("PACF Plots", outer = T)
## --------------------------------------- we work with
## dx for the analysis test the p-value
tseries::adf.test(dx)
## --------------------------------------- eacf test
TSA::eacf(dx)  # 2 choices AR(4,0) ARMA(2,2)
# Start with ARMA(4,0) ---------
# fit the model
arma.fit3 = sarima(dx, p = 4, d = 0, q = 0, details = T)
# tsdiag(arma.fit$fit) diagnostic plots
resids3 = residuals(arma.fit3$fit)  # calculate residuals
# plot the residuals withe the forecast package
# resids1%>%forecast::ggtsdisplay(main='TS-ACF-PACF for
# residuals', col=sample(colors(),1),theme=theme_gray())
# forecast::gghistogram(resids1,add.normal = T,add.kde =
# T) # plot of residuals with gghistogram plot the
# histtogram of the residuals with basic
hist(resids3, 30, col = sample(colors(), 1), border = "red",
    panel.first = grid(25, 25), freq = F, main = "Plot of the residuals for ARMA(4,0)")
lines(density(resids3), lwd = 2)
rug(resids3, col = "red4")
## ---------------------------------------------
kable(unlist(TSA::runs(resids3)))
# Start with ARMA(0,3) ---------
# fit the model
arma.fit4 = sarima(dx, p = 2, d = 0, q = 2, details = T)
# tsdiag(arma.fit$fit) diagnostic plots
resids4 = residuals(arma.fit4$fit)  # calculate residuals
# plot the residuals withe the forecast package
# resids1%>%forecast::ggtsdisplay(main='TS-ACF-PACF for
# residuals', col=sample(colors(),1),theme=theme_gray())
# forecast::gghistogram(resids1,add.normal = T,add.kde =
# T) # plot of residuals with gghistogram plot the
# histtogram of the residuals with basic
hist(resids4, 30, col = sample(colors(), 1), border = "red",
    panel.first = grid(25, 25), freq = F, main = "Plot of the residuals for ARMA(0,3)")
lines(density(resids4), lwd = 2)
rug(resids4, col = "red4")
## ----------------------------------
kable(unlist(TSA::runs(resids4)))
## ------------------------------------- ckeck the AIC
## and BIC for each model
AIC_BIC_mat1 = cbind(c(arma.fit3$AIC, arma.fit4$AIC), c(arma.fit3$BIC,
    arma.fit4$BIC))  # ; AIC_mat
colnames(AIC_BIC_mat1) = c("AIC", "BIC")
rownames(AIC_BIC_mat1) = c("ARMA(4,0)", "ARMA(2,2)")
kable(AIC_BIC_mat1)
# we choose the ARMA(1,1) and make predictions
par(mfrow = c(1, 1), bg = "whitesmoke")
S2 <- sarima.for(unemp, n.ahead = 20, p = 1, d = 1, q = 1,
```

```
    P = 4, D = 1, Q = 0, S = 12)
# the same results but using predict
# fore2=predict(arima(oil,order=c(1,0,1),include.mean =
# F),n.ahead = 20)
# cols4=sample(colors(),2)
# ts.plot(oil,fore2$pred,col=cols4,
# lwd=3,main='ARMA(1,1) with predictions') U2 =
# fore2$pred+fore2$se; L2 = fore2$pred-fore2$se xx2 =
# c(time(U2), rev(time(U2))); yy2 = c(L2, rev(U2))
# polygon(xx2, yy2, border = 8, col = gray(.6, alpha =
# .2)) points(fore2$pred, pch=19, col=2,cex=0.3)
# legend('topleft',legend=c('Oil TS','Predicted Oil 20
# ahead'), col=cols1,lty=1,lwd=2)
## ----ref.label=knitr::all_labels(), echo = T, eval =
## F-------------------
```