John Raphael Fox

Homework 4

6.1 – An input and output time series consists of 300 observations. The prewhitened input series is well modeled by an AR(2) model $y_t = 0.5y_{t-1} + 0.2y_{t-2} + \alpha_t$. We have estimated $\hat{\sigma}_\alpha = 0.2$ and $\hat{\sigma}_\beta = 0.4$. The estimated cross-correlation function between the prewhitened input and output time series is shown below.

Lag, j	0	1	2	3	4	5	6	7	8	9	10
$r_{\alpha\beta}(j)$	0.01	0.03	-0.03	-0.25	-0.35	-0.51	-0.30	-0.15	-0.02	0.09	-0.01

a. Find the approximate standard error of the cross-correlation function. Which spikes on the cross-correlation function appear to be significant?

Since it is prewhitened the variance of the cross-correlation function is given by

$$var\left(r_{xy}(j)\right) \approx \frac{1}{N}$$

Which indicates that the standard error is

$$standard\ error = \frac{1}{\sqrt{N}} = 0.058$$

Spikes 3-7 seem to be significant in the cross-correlation function.

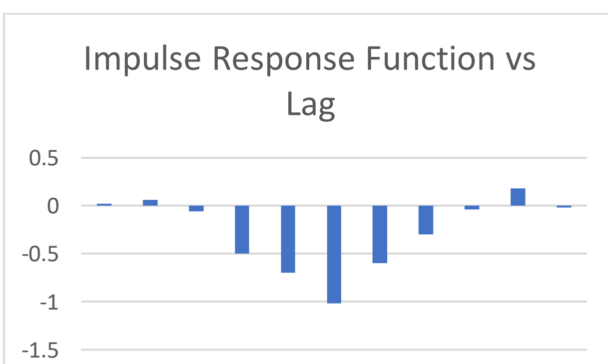
b. Estimate the impulse response function. Tentatively identify the form of the transfer function model.

From the expression

$$\hat{v}_j = r_{\alpha\beta}(j)\frac{\hat{\sigma}_\beta}{\hat{\sigma}_\alpha}$$

We obtain

Lag	0	1	2	3	4	5	6	7	8	9	10
$\hat{v}_j$	0.02	0.06	-0.06	-0.5	-0.7	-1.02	-0.6	-0.3	-0.04	0.18	-0.02



By analyzing the impulse response vs lag plot we can see that a tentative transfer function model could be one with $b = 3$, $s = 2$, and $r = 1$.

6.2 – Find initial estimates of the parameters of the transfer function model for the situation in Exercise 6.1.

For this case we have that

$$v_0 = v_1 = v_2 = 0$$

$$v_3 = \omega_0$$

$$v_4 = \delta_1 \omega_0 - \omega_1$$

$$v_5 = \delta_1^2 \omega_0 - \delta_1 \omega_1 - \omega_2$$

$$v_j = \delta_1 v_{j-1}, \qquad j > 4$$

With this we have that

$$\omega_0 = -0.5$$

$$\delta_1 = 0.5$$

$$\omega_1 = 0.45$$

$$\omega_2 = 0.67$$

6.8 – Consider a transfer function model with $b = 2$, $r = 1$, and $s = 0$. Assume that the noise model is AR(1). Find the forecast in terms of the transfer function and noise model parameters.

The transfer function-noise model is the following

$$y_t = \frac{\omega_0}{1 - \delta_1 B} x_{t-2} + \frac{1}{1 - \phi_1 B} \epsilon_t$$

Rearranging we get

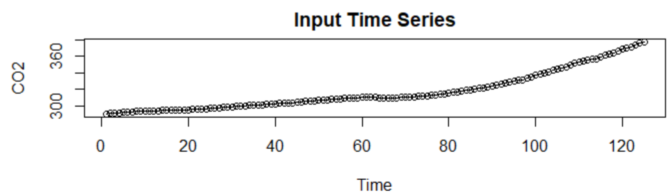
$$y_t = (\phi_1 + \delta_1) y_{t-1} - \delta_1 \phi_1 y_{t-2} + \omega_0 x_{t-2} - \omega_0 \phi_1 x_{t-3} + \epsilon_t - \delta_1 \epsilon_{t-1}$$

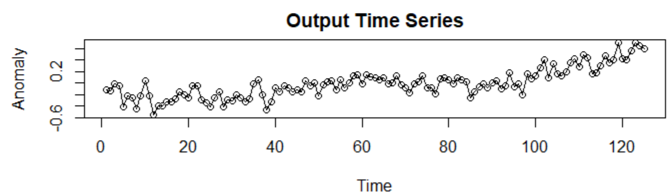
Which means that the τ-step ahead expression is

$$y_{t+\tau} = (\phi_1 + \delta_1) y_{t+\tau-1} - \delta_1 \phi_1 y_{t+\tau-2} + \omega_0 x_{t+\tau-2} - \omega_0 \phi_1 x_{t+\tau-3} + \epsilon_{t+\tau} - \delta_1 \epsilon_{t+\tau-1}$$

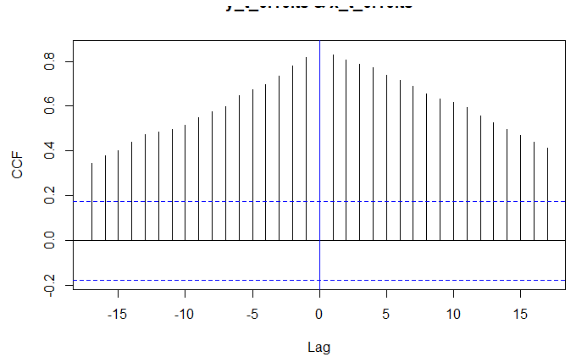
6.16 – Consider the global mean surface air temperature anomaly and global CO_2 concentration data in Table B.6 in Appendix B. Fit an appropriate transfer function model to this data, assuming that CO_2 concentration is the input variable.

Plotting both the input and output time series we obtain the following:





The cross-correlation function between these two time-series is:



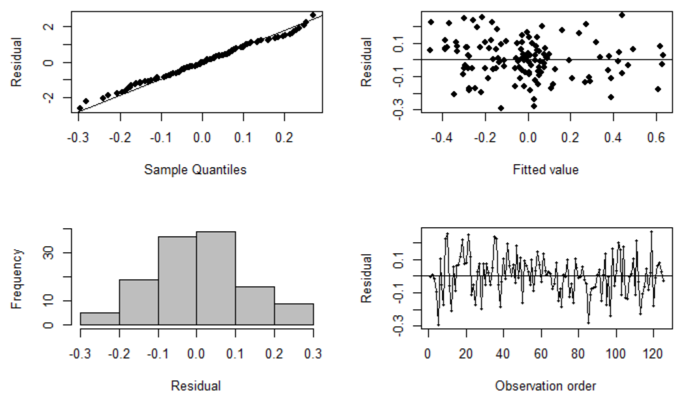
Which indicates that they are cross-correlated. For the pre-whitening we decided to use an ARIMA(1,2,0); and after much work (See attached code), we consider that we have found an appropriate model for this problem. The model is the following:

$$y_t = \frac{0.0156}{1 + 0.0301B} x_t + \frac{1 + 0.9462B + 0.6294B^2 - 0.6294B^3}{1 + 0.6099B + 0.0674B^2 + 0.3197B^3} \epsilon_t$$

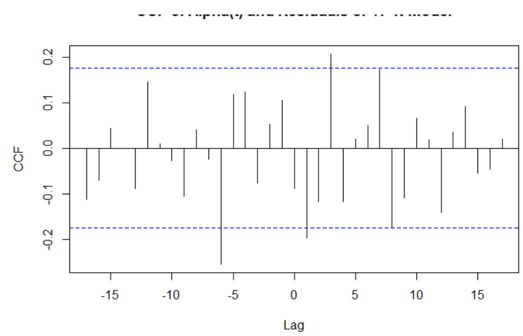
Which corresponds to a model with $b = 0$, $s = 0$, $r = 1$, $q = p = 3$, $d = 2$. The following results were obtained by using R

```
Call:
arimax(x = data2$yt, order = c(3, 2, 3), include.mean = FALSE, xtransf = data.frame(data2$lag0x),
    transfer = list(c(1, 0)))

Coefficients:
          ar1      ar2      ar3      ma1      ma2     ma3  data2.lag0x-AR1  data2.lag0x-MA0
      -0.6099  -0.0674  -0.3197  -0.9462  -0.6831  0.6294          -0.0363           0.0156
s.e.   0.1399   0.1454   0.0962   0.1396   0.1956  0.1258           0.0301           0.0046

sigma^2 estimated as 0.01435:  log likelihood = 81.93,  aic = -147.87
```

To check the adequacy of the model we analyzed the normality and constant variance of the residuals



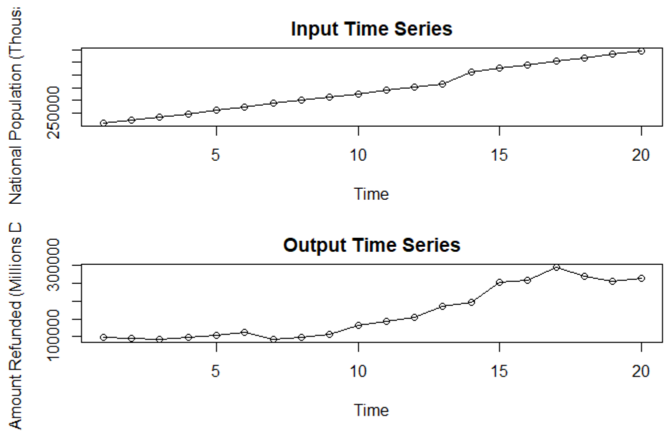
We also checked the cross-correlation between α_t and the residuals of the model



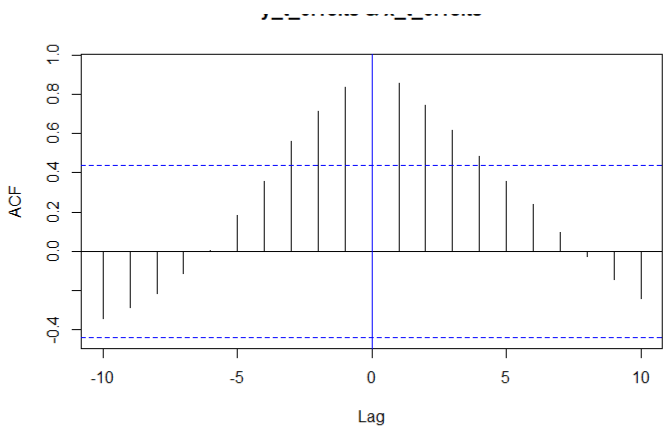
Which results were satisfactory.

6.18 – Consider the US Internal Revenue tax refund data in Table B.20. Fit an appropriate transfer function model to these data, assuming that population is the input variable. Does including the population data improve your ability to forecast the tax refund data?

Plotting both time series



The two time-series have the following CCF



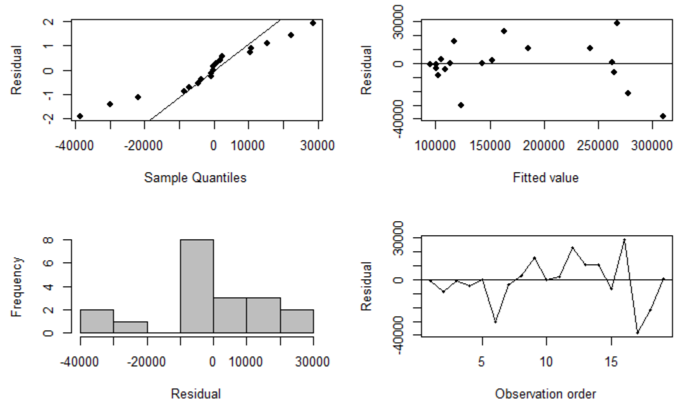
Which indicate that they are cross-correlated. To pre-whiten we selected an AR(1) model. After hard work (see code attachment) we concluded that the best model was the following:

$$y_t = (-1.6294 - 5.2551B)x_t + \frac{1}{1 - 0.1043B}\epsilon_t$$

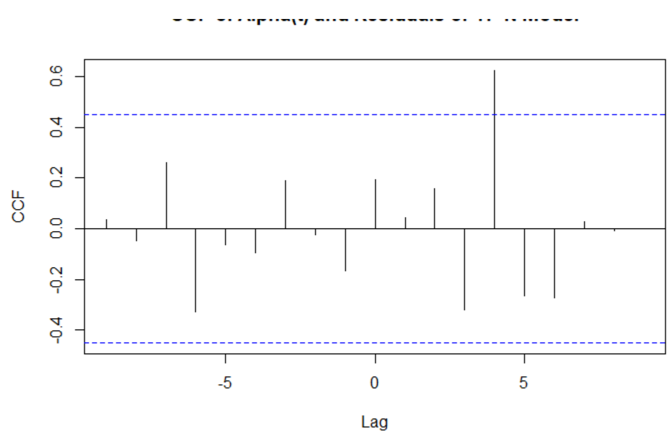
Which corresponds to $b = 0, s = 1, r = 0, p = 1, d = 1, q = 0$. The results by using R gives us

```
Call:
arimax(x = data2$yt, order = c(1, 1, 0), include.mean = FALSE, xtransf =
data.frame(data2$lag1x),
    transfer = list(c(0, 1)))

Coefficients:
         ar1  data2.lag1x-MA0  data2.lag1x-MA1
      0.1043          -1.6294           5.2551
s.e.  0.2377           1.7829           1.7912

sigma^2 estimated as 2.68e+08:  log likelihood = -200.21,  aic = 406.41
```

To check the adequacy of the model we analyzed the normality and constant variance assumptions of the residuals:



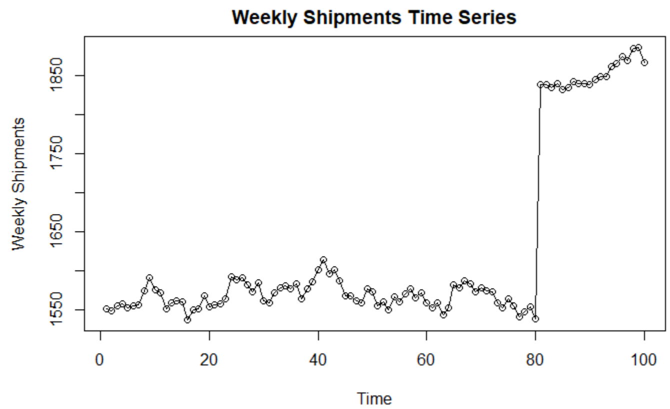
Which was believed to be satisfactory. We also checked the cross-correlation between α_t and the residuals



Which was also believed to be satisfactory. Including the population should improve the ability to forecast do to the fact that both time series are cross-correlated.

6.22 – Table E6.2 provides 100 observations on a time series. These data represent weekly shipment of a product.

a. Plot the data.



b. Note that there is an apparent increase in the level of the time series at about observation 80. Management suspects that this increase in shipments may be due to a strike at a competitor's plant. Build an appropriate intervention model for this data. Do you think that the impact of this intervention is likely to be permanent?

We consider that an appropriate intervention is the following:

$$y_t = 1851.7560S^{80} + \frac{1}{1 - 0.7320B}\epsilon_t$$

The intervention part is a change in the mean of the data, for a certain period of time in the future. That is how ω_0 was found; it is the mean of the data after the event. Since the change is for some time after 80 we used a step function. The second part of the model is just an AR(1) model of the first 80 observations.

We do not consider the impact to be permanent do to the nature of the event. Since it is thought to be a product of a strike, the conflict should end once the management of the company fixed their issues.

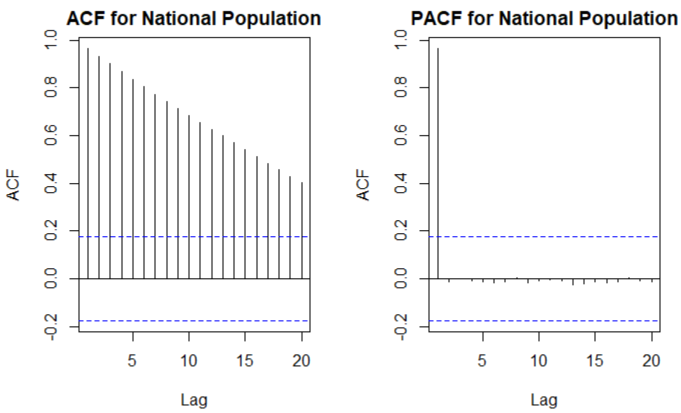
R-Notebook Attached

6.16

~~~~
Problem 6.16
```{r}
df_6.16<-read.csv("B6.csv", header = TRUE)
x_t_6.16.ts<-ts(df_6.16[,3])
y_t_6.16.ts<-ts(df_6.16[,2])
par(mfrow=c(2,1), oma=c(0,0,0,0))
plot(x_t_6.16.ts, type="o", xlab="Time", ylab="CO2", main="Input Time Series")
plot(y_t_6.16.ts, type="o", xlab="Time", ylab = "Anomaly", main = "Output Time Series")
```
~~~~

~~~~
Plotting the cross correlation between input and output
```{r}
ccf(y_t_6.16.ts, x_t_6.16.ts, ylab="CCF")
abline(v=0, col="blue")
```
~~~~

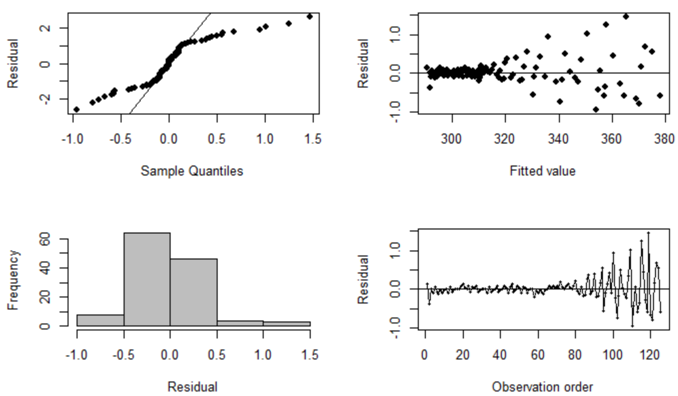
~~~~
The two series do seem to be extremely cross correlated. Now, let's plot the ACF and PACF of the input
time series
```{r}
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(x_t_6.16.ts, type = "correlation", main="ACF for National Population")
acf(x_t_6.16.ts, type = "partial", main="PACF for National Population")
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(diff(diff(x_t_6.16.ts)), type = "correlation", main="ACF for National Population")
acf(diff(diff(x_t_6.16.ts)), type = "partial", main="PACF for National Population")
```
~~~~



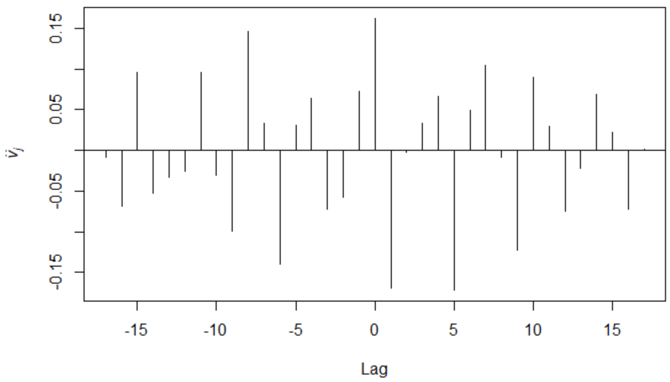

~~~~
This indicated that an AR(1) model seems to be adequate for the input time series
```{r}

x_t_6.16.ts.AR1<-arima(x_t_6.16.ts, order = c(1,2,0))
x_t_6.16.ts.AR1
#auto.arima(x_t_6.16.ts)
#x_t_6.16.ts.AR32<-arima(x_t_6.16.ts, order = c(3,2,0))
```
~~~~

For the prewhitening we will use the ARIMA(1,2,0) model. Checking model adequecy

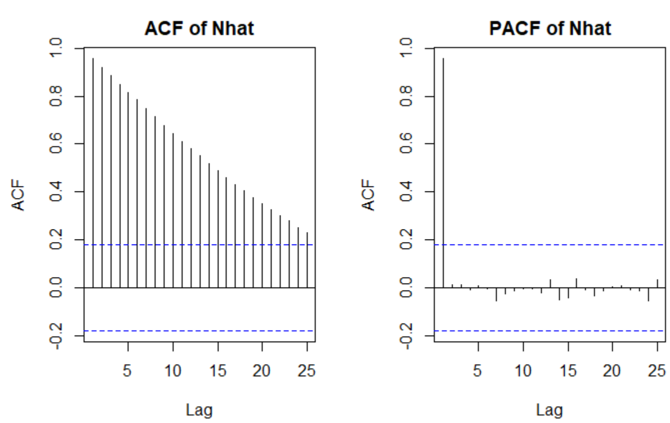
~~~~
```{r}
res.x_t_6.16.ts<-as.vector(residuals(x_t_6.16.ts.AR1))
fit.x_t_6.16.ts<-as.vector(fitted(x_t_6.16.ts.AR1))

par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.x_t_6.16.ts, lag.max = 25, type="correlation", main="ACF of the Residuals for x(t)")
acf(res.x_t_6.16.ts, lag.max = 25, type="partial", main="PACF of the Residuals for x(t)")

par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.x_t_6.16.ts, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.x_t_6.16.ts, datax=TRUE)
plot(fit.x_t_6.16.ts, res.x_t_6.16.ts, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.x_t_6.16.ts, col="gray", xlab="Residual", main="")
plot(res.x_t_6.16.ts, type="l", xlab="Observation order", ylab="Residual")
points(res.x_t_6.16.ts, pch=16, cex=0.5)
abline(h=0)
```
~~~~



Let's prewhiten

~~~~
```{r}
T<-length(x_t_6.16.ts)
alpha_t<-x_t_6.16.ts[4:T]-1.7337*x_t_6.16.ts[3:(T-1)]+0.4674*x_t_6.16.ts[2:(T-2)]+0.2663*x_t_6.16.ts[1:
(T-3)]
beta_t<-y_t_6.16.ts[4:T]-1.7337*y_t_6.16.ts[3:(T-1)]+0.4674*y_t_6.16.ts[2:(T-2)]+0.2663*y_t_6.16.ts[1:(
T-3)]
ralbe<-ccf(beta_t, alpha_t, main = "CCF of alpha(t) and beta(t)", ylab="CCF")
#abline(v=0, col="blue")
vhat<-sqrt(var(beta_t)/var(alpha_t))*ralbe$acf
nl<-length(vhat)
plot(seq(-(nl-1)/2, (nl-1)/2, 1), vhat, type="h", xlab="Lag",
ylab=expression(italic(hat(v)[italic(j)])))
#abline(v=0, col="blue")
abline(h=0)
~~~~




````
```{r}
v0<-vhat[18]
v1<-vhat[19]
v2<-vhat[20]
v3<-vhat[21]
v4<-vhat[22]
v5<-vhat[23]
```
````

Creation of N_hat and analyzing N_hat for the selectio of its model

````
```{r}
N_hat<-array(0, dim = c(1,T))
for(i in 6:T){
  N_hat[i]<-y_t_6.16.ts[i]-v0*x_t_6.16.ts[i]-v1*x_t_6.16.ts[i-1]-v2*x_t_6.16.ts[i-2]-v3*x_t_6.16.ts[i-3
]-v4*x_t_6.16.ts[i-4]-v5*x_t_6.16.ts[i-5]
}
N_hat<-N_hat[6:T]
plot(N_hat, type="o", pch=16, cex=.5, xlab="Time", ylab=expression(italic(hat(N)[italic(t)])))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(N_hat, lag.max=25, type="correlation", main="ACF of Nhat")
acf(N_hat, lag.max=25, type="partial", main="PACF of Nhat")
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(diff(N_hat), lag.max=25, type="correlation", main="ACF of Nhat")
acf(diff(N_hat), lag.max=25, type="partial", main="PACF of Nhat")

auto.arima(N_hat)
arima(N_hat, order = c(0,1,0))
par(mfrow=c(1,2), oma=c(0,0,0,0))
plot(diff(N_hat))
plot(diff(diff(N_hat)))
```
````



By using the auto.arima function we can see that N_hat is best modeled with an ARIMA(3,2,3). Now let's create the final model

```{r}
lag0.x<-lag(x_t_6.16.ts, 0)
dat0<-cbind.data.frame(x_t_6.16.ts, lag0.x, y_t_6.16.ts)
dimnames(dat0)[[2]]<-c("xt","lag0x", "yt")
data2<-na.omit(as.data.frame(dat0))

tf1.6.16<-arimax(data2$yt, order=c(1,2,0), xtransf = data.frame(data2$lag0x), transfer = list(c(1,0)), include.mean = FALSE)
tf1.6.16
tf2.6.16<-arimax(data2$yt, order=c(2,2,0), xtransf = data.frame(data2$lag0x), transfer = list(c(1,0)), include.mean = FALSE)
tf2.6.16
tf3.6.16<-arimax(data2$yt, order=c(3,2,0), xtransf = data.frame(data2$lag0x), transfer = list(c(1,0)), include.mean = FALSE)
tf3.6.16
tf4.6.16<-arimax(data2$yt, order=c(3,2,3), xtransf = data.frame(data2$lag0x), transfer = list(c(1,0)), include.mean = FALSE)
tf4.6.16
```

Model adequacy

```{r}
res.tf.6.16<-na.omit(as.vector(residuals(tf4.6.16)))
fit.tf.6.16<-na.omit(as.vector(fitted(tf4.6.16)))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.tf.6.16, lag.max = 25, type="correlation", main="ACF of the Residuals for TF-N Model")
acf(res.tf.6.16, lag.max = 25, type="partial", main="PACF of the Residuals for TF-N Model")
fit.tf.6.16
par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.tf.6.16, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.tf.6.16, datax=TRUE)
plot(fit.tf.6.16, res.tf.6.16, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.tf.6.16, col="gray", xlab="Residual", main="")
plot(res.tf.6.16, type="l", xlab="Observation order", ylab="Residual")
points(res.tf.6.16, pch=16, cex=0.5)
abline(h=0)
```

```{r}
T<-length(res.tf.6.16)
T
Ta<-length(alpha_t)+3
Ta
res.tf.6.16
alpha_t
ccf(res.tf.6.16, alpha_t[(Ta-T+1):Ta], main = "CCF of Alpha(t) and Residuals of TF-N Model", ylab = "CCF",  na.action = na.pass)
```

6.18

Here we present the R notebook for the solutions of Homework 4

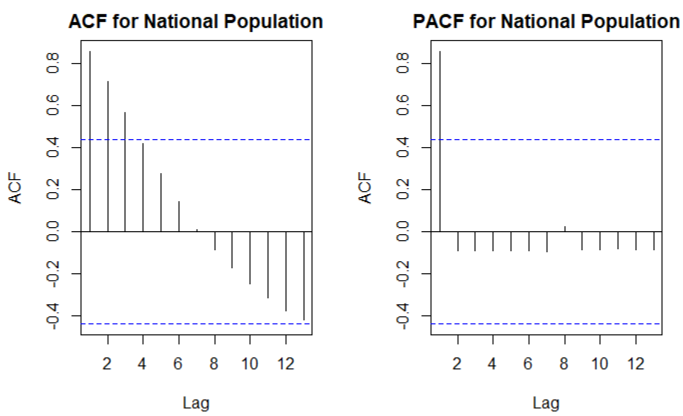
```{r}
df_6.18<-read.csv("B20.csv", header = TRUE)
x_t_6.18<-df_6.18[,3]
y_t_6.18<-df_6.18[,2]
x_t_6.18.ts<-ts(x_t_6.18)
y_t_6.18.ts<-ts(y_t_6.18)
par(mfrow=c(2,1), oma=c(0,0,0,0))
plot(x_t_6.18.ts, type="o", xlab="Time", ylab="National Population (Thousands)", main="Input Time Series")
plot(y_t_6.18.ts, type="o", xlab="Time", ylab = "Amount Refunded (Millions Dollars)", main = "Output Time Series")
```

Plotting the Cross corrilation function CCF

```{r}
ccf(y_t_6.18.ts, x_t_6.18.ts)
abline(v=0, col="blue")
```

The input and output time series seem to be cross correlated. Now let's plot the input time series ACF and PACF

```{r}
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(x_t_6.18.ts, type = "correlation", main="ACF for National Population")
acf(x_t_6.18.ts, type = "partial", main="PACF for National Population")
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(diff(x_t_6.18.ts), type = "correlation", main="ACF for National Population")
acf(diff(x_t_6.18.ts), type = "partial", main="PACF for National Population")
```



By looking at the first set of ACF and PACF it looks like an AR(1) model would work, but the time series still seems non-stationary. So what we did was first difference the time series and plotted the ACF and PACF, this looks a lot better and it indicated that an IMA(1,1) model would be most appropriate. Hense, we try an AR(1) and an IMA(1)

```{r}
x_t_6.18.ts.AR1<-arima(x_t_6.18.ts, order = c(1,0,0))
arima(x_t_6.18.ts, order = c(1,1,0))
#x_t_6.18.ts.IMA<-arima(x_t_6.18.ts, order = c(0,1,1))
#x_t_6.18.ts.IMA
x_t_6.18.ts.AR1
```

```
Call:
arima(x = x_t_6.18.ts, order = c(1, 1, 0))

Coefficients:
         ar1
      0.7600
s.e.  0.1365

sigma^2 estimated as 4508442:  log likelihood = -172.94,  aic = 347.89

Call:
arima(x = x_t_6.18.ts, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.9923  270589.91
s.e.  0.0108   26310.85

sigma^2 estimated as 11318627:  log likelihood = -192.89,  aic = 389.78
```

We select the ARIMA(1,0,0). Checking model adequacy

```
```{r}
res.x_t_6.18.ts<-as.vector(residuals(x_t_6.18.ts.AR1))
fit.x_t_6.18.ts<-as.vector(fitted(x_t_6.18.ts.AR1))

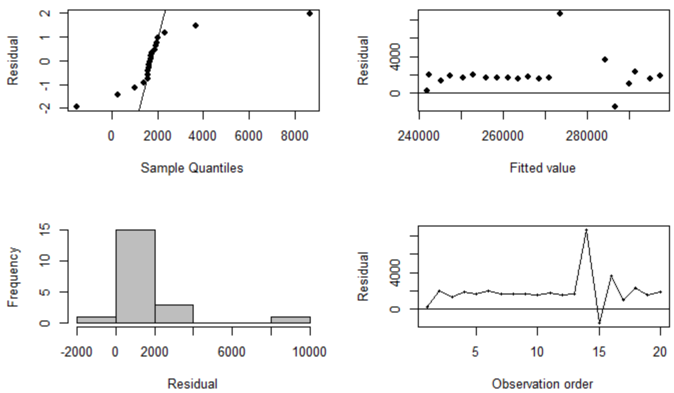
par(mfrow=c(1,2), oma=c(0,0,0,0))
```

```
acf(res.x_t_6.18.ts, lag.max = 25, type="correlation", main="ACF of the Residuals for x(t)")
acf(res.x_t_6.18.ts, lag.max = 25, type="partial", main="PACF of the Residuals for x(t)")

par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.x_t_6.18.ts, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.x_t_6.18.ts, datax=TRUE)
plot(fit.x_t_6.18.ts, res.x_t_6.18.ts, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.x_t_6.18.ts, col="gray", xlab="Residual", main="")
plot(res.x_t_6.18.ts, type="l", xlab="Observation order", ylab="Residual")
points(res.x_t_6.18.ts, pch=16, cex=0.5)
abline(h=0)
res.x_t_6.18.ts<-as.vector(residuals(x_t_6.18.ts.IMA))
fit.x_t_6.18.ts<-as.vector(fitted(x_t_6.18.ts.IMA))

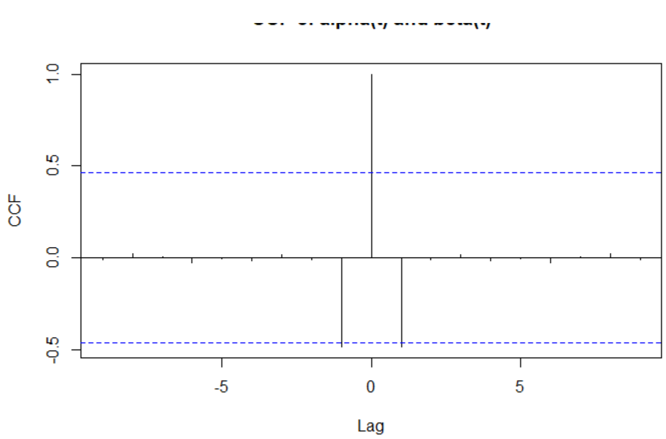
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.x_t_6.18.ts, lag.max = 25, type="correlation", main="ACF of the Residuals for x(t)")
acf(res.x_t_6.18.ts, lag.max = 25, type="partial", main="PACF of the Residuals for x(t)")

par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.x_t_6.18.ts, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.x_t_6.18.ts, datax=TRUE)
plot(fit.x_t_6.18.ts, res.x_t_6.18.ts, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.x_t_6.18.ts, col="gray", xlab="Residual", main="")
plot(res.x_t_6.18.ts, type="l", xlab="Observation order", ylab="Residual")
points(res.x_t_6.18.ts, pch=16, cex=0.5)
abline(h=0)
```
```



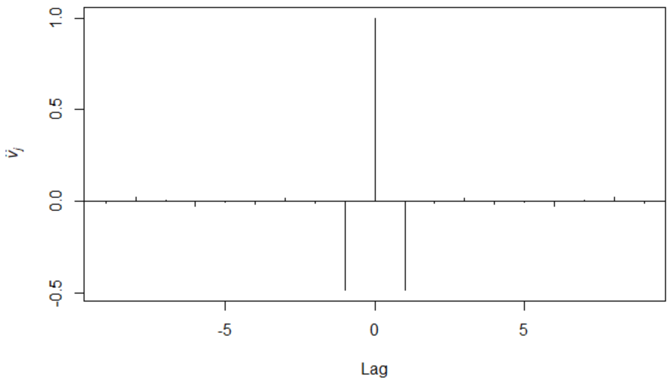

It seems pretty good, but it does have a couple of outliers. As can be seen from the time series plot as well.

```{r}
T<-length(x_t_6.18.ts)
#alpha_t<-diff(x_t_6.18.ts)[2:T]-0.76*diff(x_t_6.18.ts)[1:(T-1)]
alpha_t<-x_t_6.18.ts[3:T]-1.76*x_t_6.18.ts[2:(T-1)]+0.76*x_t_6.18.ts[1:(T-2)]
beta_t<-x_t_6.18.ts[3:T]-1.76*x_t_6.18.ts[2:(T-1)]+0.76*x_t_6.18.ts[1:(T-2)]
#beta_t<-diff(y_t_6.18.ts)[2:T]-0.76*diff(y_t_6.18.ts)[1:(T-1)]
ralbe<-ccf(beta_t, alpha_t, main = "CCF of alpha(t) and beta(t)", ylab="CCF")
#abline(v=0, col="blue")
```



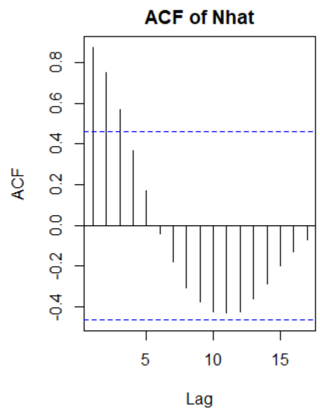
Obtaining the estimates of v_t

```{r}
v_hat<-sqrt(var(beta_t)/var(alpha_t))*ralbe$acf
n1<-length(v_hat)
plot(seq(-(n1-1)/2, (n1-1)/2,1), v_hat, type = 'h', xlab = 'Lag', ylab =
expression(italic(hat(v)[italic(j)])))
#abline(v=0, col='blue')
abline(h=0)
v0<-v_hat[10]
v1<-v_hat[11]
```



Now to model the noise

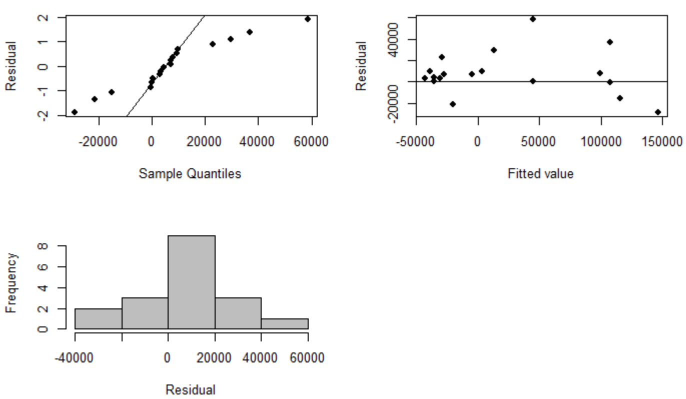
```{r}
N_hat<-array(0, dim = c(1,T))
for(i in 2:T){
  N_hat[i]<-y_t_6.18.ts[i]-v0*x_t_6.18.ts[i]-v1*x_t_6.18.ts[i-1]
}
N_hat<-N_hat[3:T]
N_hat
plot(N_hat, type="o", pch=16, cex=.5, xlab="Time", ylab=expression(italic(hat(N)[italic(t)])))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(N_hat, lag.max=25, type="correlation", main="ACF of Nhat")
acf(N_hat, lag.max=25, type="partial", main="PACF of Nhat")
```



It seems that N_hat can be modeled as an AR(1) or AR(2)

~~~
```{r}
N_hat.ar1<-arima(N_hat, order = c(1,1,0), include.mean = FALSE)
auto.arima(N_hat)
res.N_hat.ar1<-as.vector(residuals(N_hat.ar1))
fit.N_hat.ar1<-as.vector(fitted(N_hat.ar1))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.N_hat.ar1, lag.max = 25, type="correlation", main="ACF of the Residuals for N_hat")
acf(res.N_hat.ar1, lag.max = 25, type="partial", main="PACF of the Residuals for N_hat")

par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.N_hat.ar1, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.N_hat.ar1, datax=TRUE)
plot(fit.N_hat.ar1, res.N_hat.ar1, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.N_hat.ar1, col="gray", xlab="Residual", main="")
plot(res.N_hat.ar1, type="l", xlab="Observation order", ylab="Residual")
points(res.N_hat.ar1, pch=16, cex=0.5)
abline(h=0)
~~~



Using the arimax funtion in TSA

~~~
```{r}
lag1.x_t_6.18.ts<-lag(x_t_6.18.ts,0)
dat1<-cbind(x_t_6.18.ts, lag1.x_t_6.18.ts, y_t_6.18.ts)
dimnames(dat1)[[2]]<-c("xt","lag1x","yt")
data2<-na.omit(as.data.frame(dat1))
data2
#Input arguments
#order: determines the model for the error component, i.e. the order of the
#ARIMA model for y(t) if there were no x(t)

#xtrans:x(t)
#transfer: the orders (r and s) of the trasfer function
tf.6.18<-arimax(data2$yt, order=c(1,1,0), xtransf = data.frame(data2$lag1x), transfer=list(c(0,1)),
include.mean = FALSE)
tf.6.18
#ARIMAX.6.18.1<-arimax(y_t_6.18.ts, order = c(1,0,0), transfer = list(c(0,1)))
```
~~~

Model adquecy check
```{r}
res.tf.6.18<-na.omit(as.vector(residuals(tf.6.18)))
fit.tf.6.18<-na.omit(as.vector(fitted(tf.6.18)))
par(mfrow=c(1,2), oma=c(0,0,0,0))
acf(res.tf.6.18, lag.max = 25, type="correlation", main="ACF of the Residuals for TF-N Model")
acf(res.tf.6.18, lag.max = 25, type="partial", main="PACF of the Residuals for TF-N Model")
fit.tf.6.18
par(mfrow=c(2,2), oma=c(0,0,0,0))
qqnorm(res.tf.6.18, datax=TRUE, pch=16, xlab="Residual", main="")
qqline(res.tf.6.18, datax=TRUE)
plot(fit.tf.6.18, res.tf.6.18, pch=16, xlab="Fitted value", ylab="Residual")
abline(h=0)
hist(res.tf.6.18, col="gray", xlab="Residual", main="")
plot(res.tf.6.18, type="l", xlab="Observation order", ylab="Residual")
points(res.tf.6.18, pch=16, cex=0.5)
abline(h=0)
```

```{r}
T<-length(res.tf.6.18)
T
Ta<-length(alpha_t)+1
Ta
res.tf.6.18
alpha_t
ccf(res.tf.6.18, alpha_t[(Ta-T+1):Ta], main = "CCF of Alpha(t) and Residuals of TF-N Model", ylab =
"CCF",  na.action = na.pass)
```

Problem 6.22

Problem 6.22
```{r}
df_6.22<-read.csv("E6_2.csv", header = FALSE)
df_6.22.ts<-ts(df_6.22)
plot.ts(df_6.22.ts, type="o", ylab = "Weekly Shipments", main = "Weekly Shipments Time Series")
```

We will fit an ARIMA model for the first 80 observetions
```{r}
first_80_observations<-df_6.22.ts[1:80]
plot.ts(first_80_observations, type = "o")
auto.arima(first_80_observations)
```

Fitting an ARIMA model for the last 20 observations. So we can calculate the mean and with this fit an intevention model
```{r}
last_20_observations<-df_6.22.ts[81:100]
plot.ts(last_20_observations)
auto.arima(last_20_observations)
arima(last_20_observations, order = c(1,0,0))
mean(last_20_observations)

```

```{r}
acf(last_20_observations, type = "correlation")
acf(last_20_observations, type = "partial")
arima(last_20_observations, order = c(1,1,0), include.mean = TRUE)
arima(last_20_observations, order = c(1,0,0))
```