Final Assignment

The purpose of this study is to see if the Federal Reserve's policy affects basic macroeconomic variables such as real GDP and inflation. To examine if monetary policy has a major impact on them, the SVAR framework will be employed. he real chained-2009 GDP (GDPC96), a chain type price indexed at 2009 (gdpctpi), and the ffr(FEDFUNDS) are used in the analysis. GDP and Price were converted to growth rates, and the data were collected on a quarterly basis from 7/1/1954 to 4/1/2017. The first step is to examine each variable's stationarity.

GDPC96

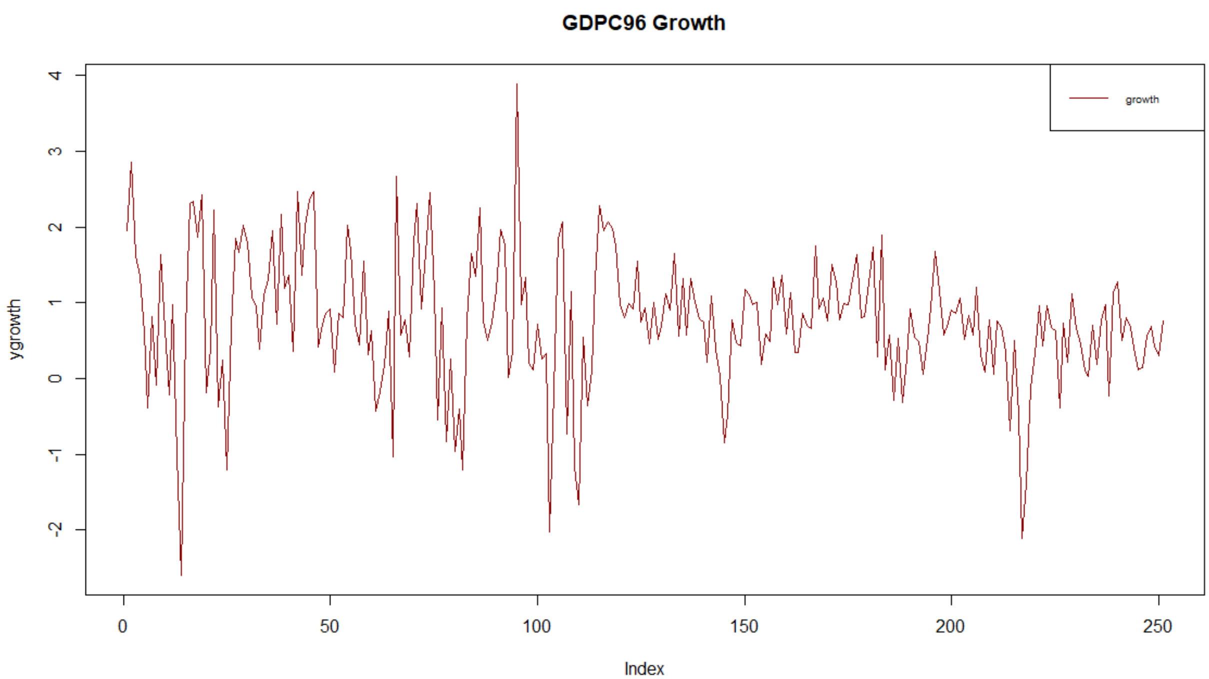

Figure 1.GDPC96 Growth

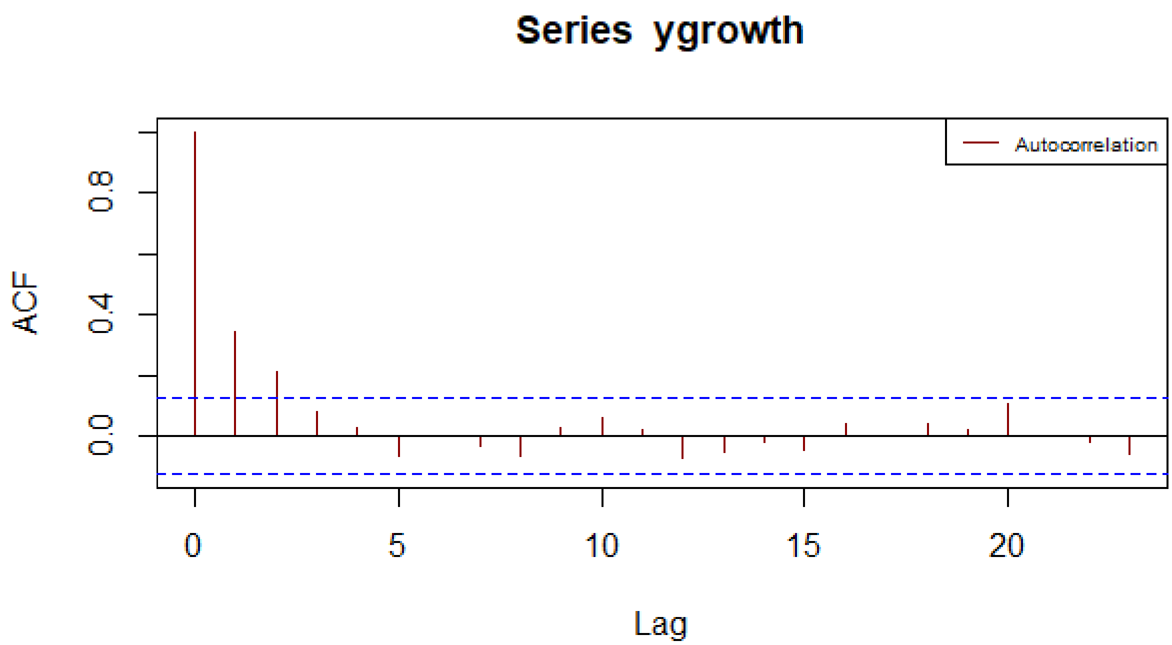

Figure 2.Autocorrelation Function

	1%	**2%**	**5%**
tau3	**-3.98**	**-3.42**	**-3.13**
phi2	6.15	4.71	4.05
phi3	8.34	6.3	5.36

Table 1.Dickey-fuller Test

The series appears to be stationary in Figures 1 and 2, with the exception of the autocorrelation function, which appears in the early lags to be correlated. Because unit root data cannot be used in the analysis, the Dickey-Fuller test should be used and the outcomes are shown in Table 1. The t statistic is **-8.4613**, and no tabulated value exceeds it, as indicated in the table, so we reject the null hypothesis that GDPC96 has a Unit Root.

GDPCTPI

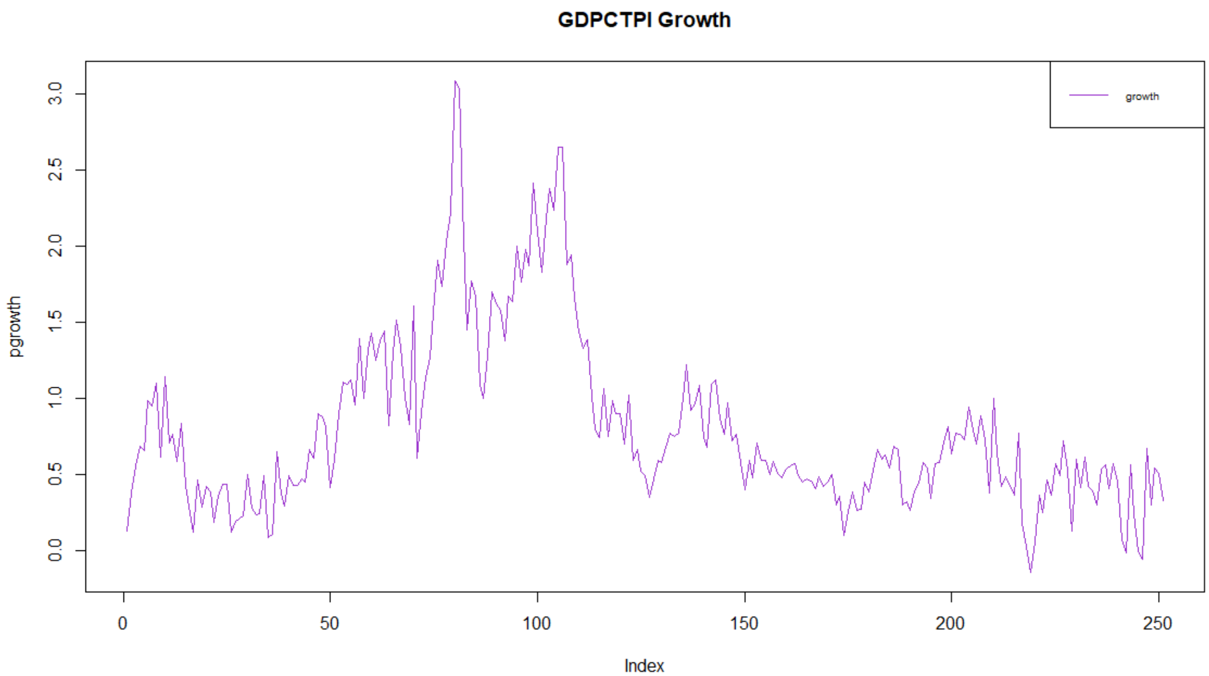

Figure 3. GDPCTPI Growth

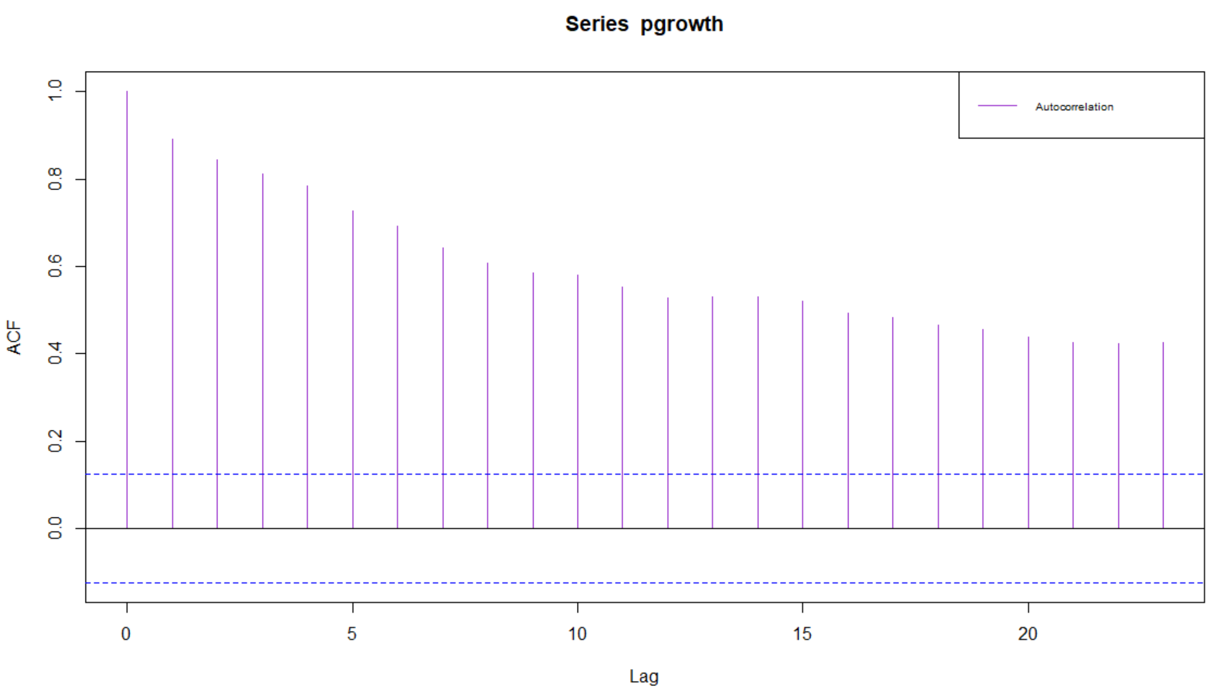

Figure 4.Autocorrelation Function

	t statistic	1%	5%	10%
trend	-3.1315	-3.98	-3.42	-3.13
drift	-2.8438	-3.44	-2.87	-2.57
none	-1.597	-2.58	-1.95	-1.62

Table 2.Dickey-Fuller test

The GDPCTPI series does not appear to be stationary, as shown in figures 3 and 4, as well as in table 2. We can't reject the null hypothesis in this situation, which suggests the series has a Unit Root, and we'll have to difference the data and convert it to stationary.

FEDDFUNDS

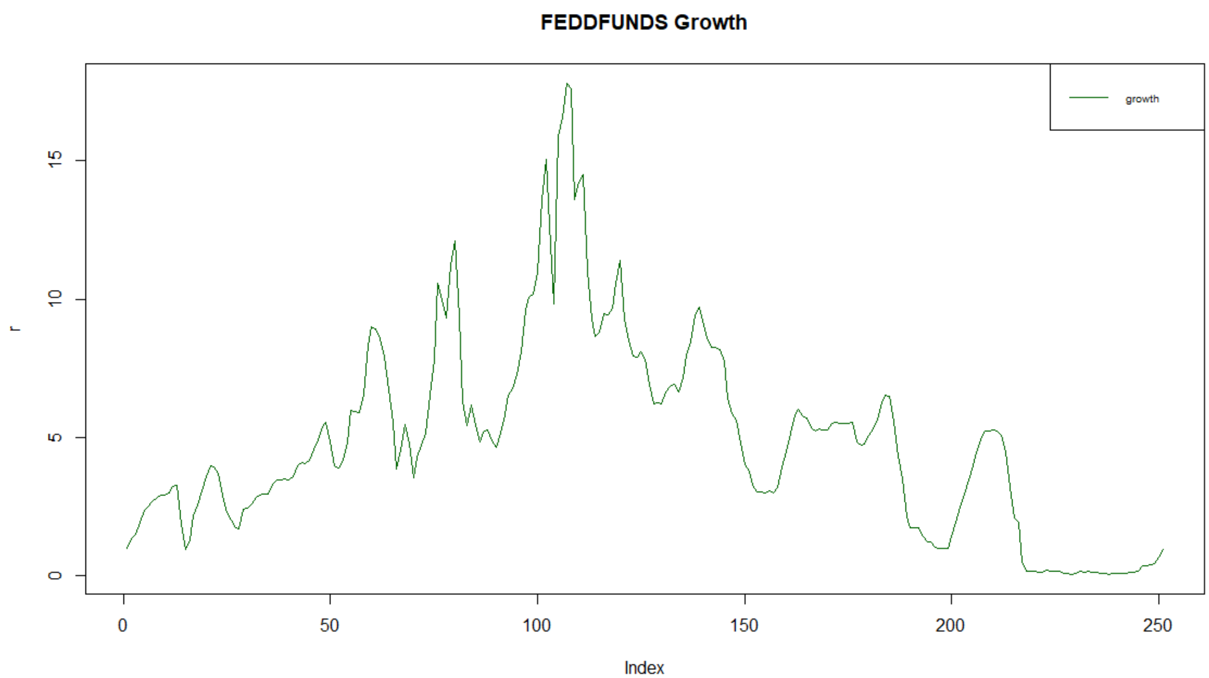

Figure 5. FEDDFUNDS Growth

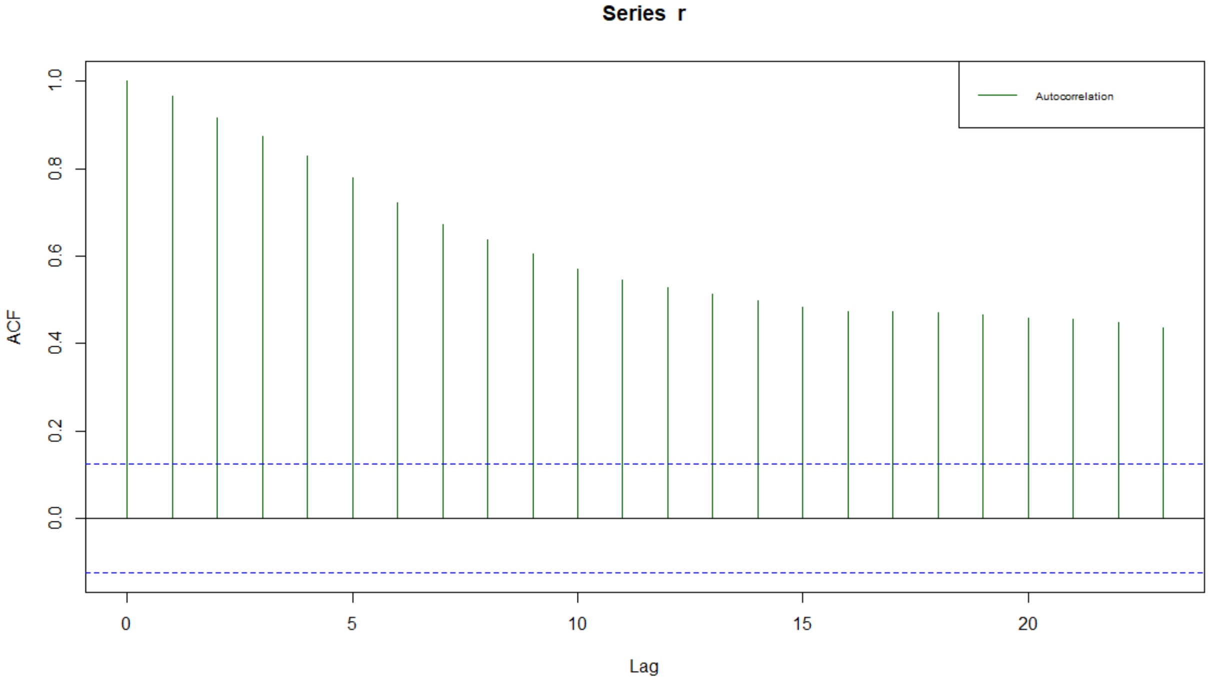

Figure 6.Autocorrelation Function

	t statistic	1%	5%	10%
trend	-2.8021	-3.98	-3.42	-3.13
drift	-2.4915	-3.44	-2.87	-2.57
none	-1.4666	-2.58	-1.95	-1.62

Table 3.Dickey-Fuller Test

The FEDDFUNDS series does not appear to be stationary, as shown in figures 3 and 4, as well as in table 2. We can't reject the null hypothesis in this situation, which suggests the series has a Unit Root, and we'll have to difference the data and convert it to stationary.

In the next step, we'll use the difference in the series using Unit Root to convert the series to stationary for GDPCPTI and FEDDFUNDS variable.

GDPCTPI

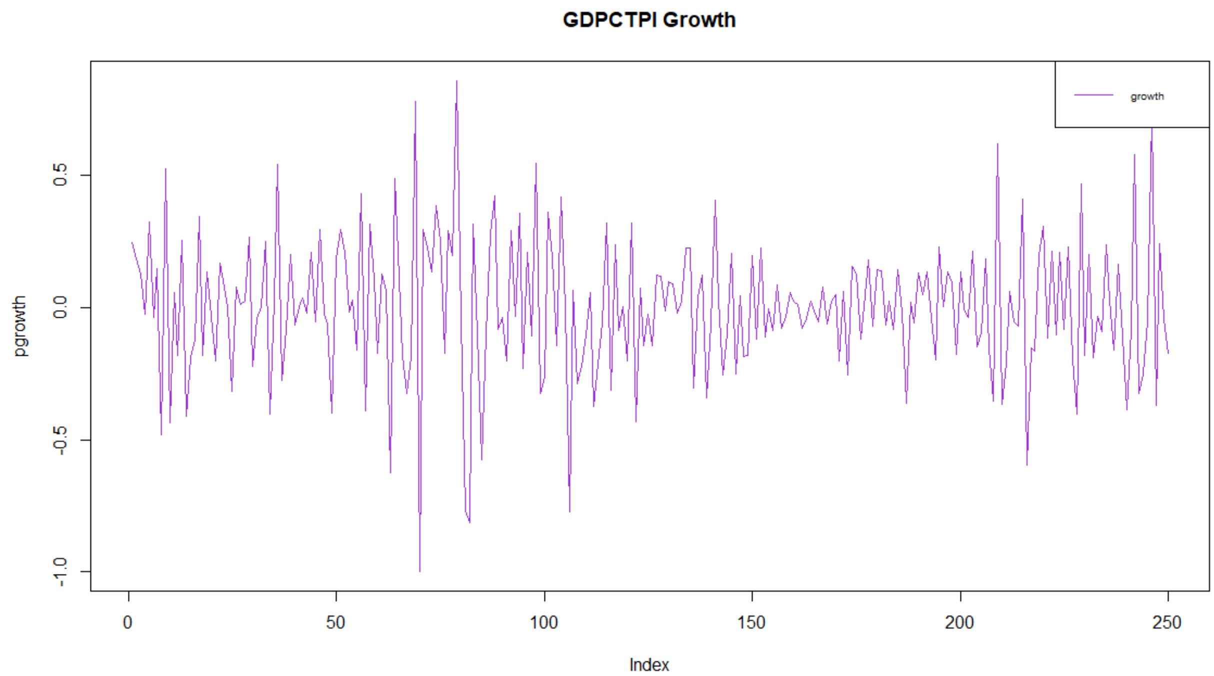

Figure 7. GDPCTPI Growth

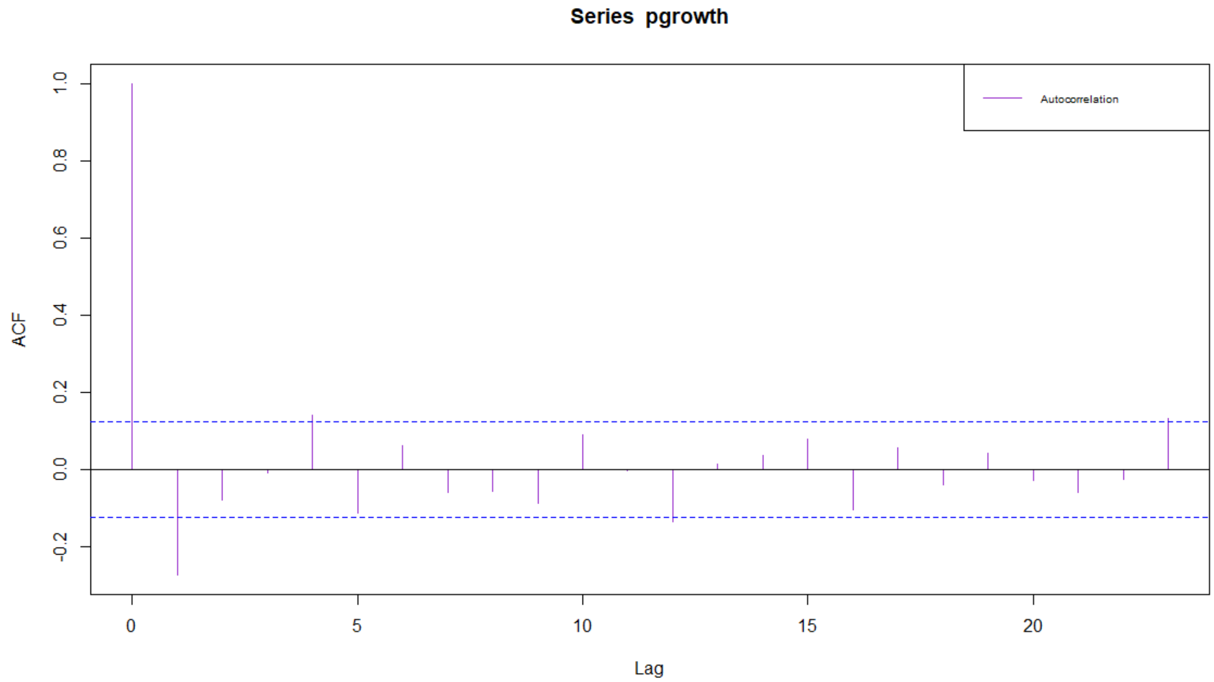

Figure 8.Autocorrelation Function

	1%	2%	5%
tau3	**-3.99**	**-3.43**	**-3.13**
phi2	6.22	4.75	4.07
phi3	8.43	6.49	5.47

Table 4.Dickey-Fuller Test

There appear to be differences in Figures 7 and 8 after the alteration. Especially in the ACF plot, where there were formerly spikes, it appears that they no longer exist. Finally it appears that Dickey-prices Fuller's have altered so we reject the null hypothesis that GDPCPTI has a Unit Root.

FEDDFUNDS

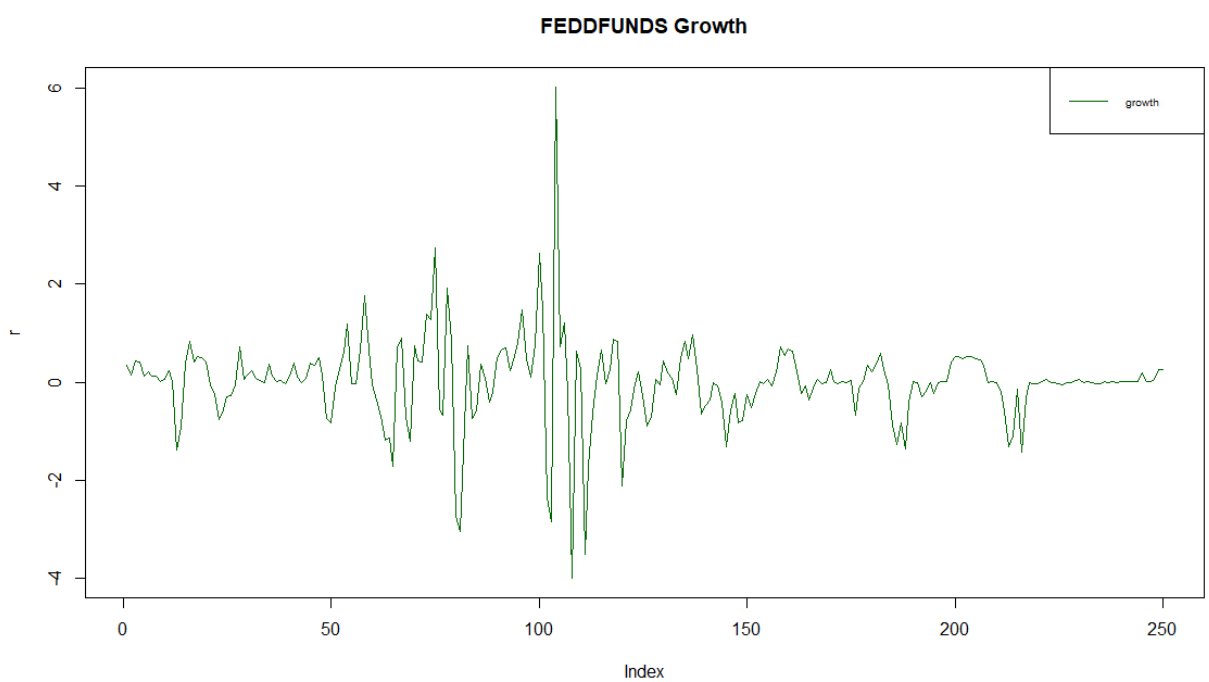

Figure 9. FEDDFUNDS Growth

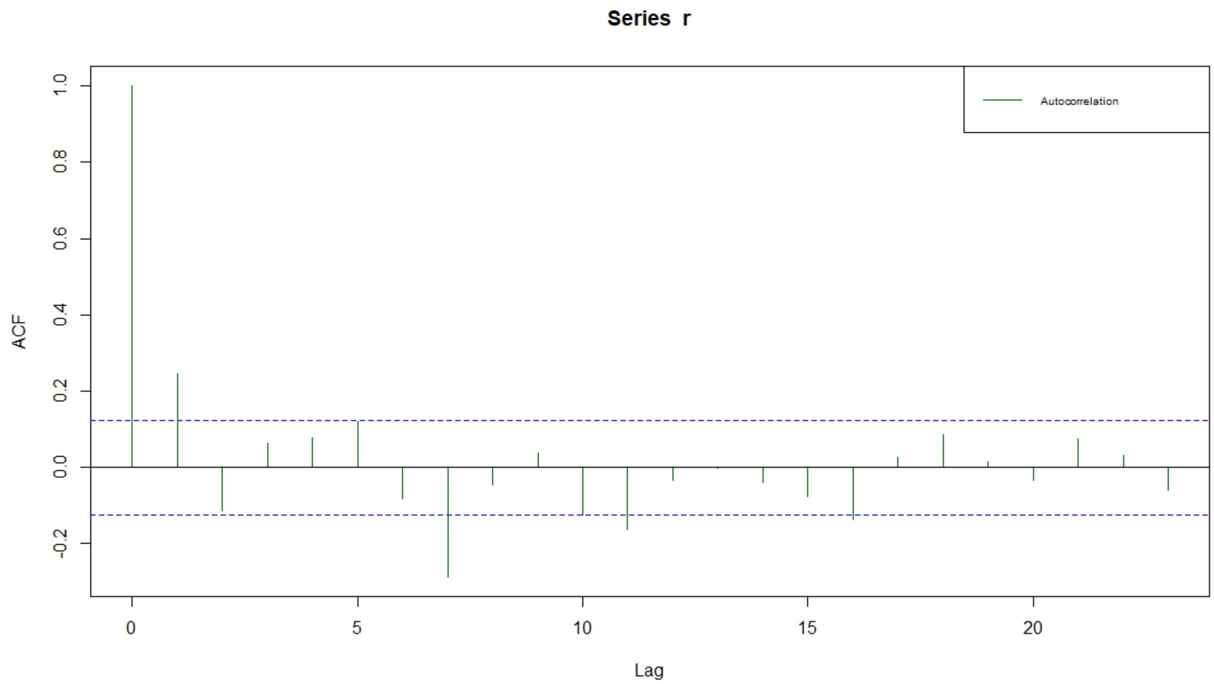

Figure 10.Autocorrelation Function

	1%	2%	5%
tau3	-3.99	-3.43	-3.13
phi2	6.22	4.75	4.07
phi3	8.43	6.49	5.47

Table 5.Dickey-Fuller Test

And again there appear to be differences in Figures 9 and 10 after the alteration. Especially in the ACF plot, where there were formerly spikes, it appears that they no longer exist. Finally it appears that Dickey-prices Fuller's have altered so we reject the null hypothesis that FEDDFUNDS has a Unit Root.

VAR Analysis

Next, we'll determine the maximum number of lags that can be checked over (pmax) using the following information criteria: to Akaike (AIC),Shwarz-Bayesian (SBIC) και Hannan-Quinn (HQIC). Table 6 illustrates the results, indicating that two and ten lags are the appropriate number.

	Lags
AIC	10
SBIC	2
HQIC	2

Table 6.Number of Lags

Then, in order to calculate the *Impulse Response Function* (IRF), the horizon is set to 10 periods ahead. After a shock in a specific moment, an Impulse Response Function represents the evolution of the variable of interest over a specified time horizon. A shock is an uncorrelated and economically significant exogenous force, such as monetary shocks. In this study, we'll look at how FEDFUNDS errors affect the other two variables.

0.774	0.000	0.000
-0.005	0.241	0.000
0.153	0.125	0.761

Table 7.Cholesky Matrix

For the shocks, a triangle identification technique is utilized. We're looking for shocks on structural errors to find the impact multiplier matrix and reform the reduced form errors into structural errors. To identify the matrix and thus the shocks, it has been employed the Cholesky decomposition. Table 7 reveals that there is a lower triangular table, indicating that interest rate shocks have no synchronous influence on the other two variables, GDPC96 and GDPCTPI.

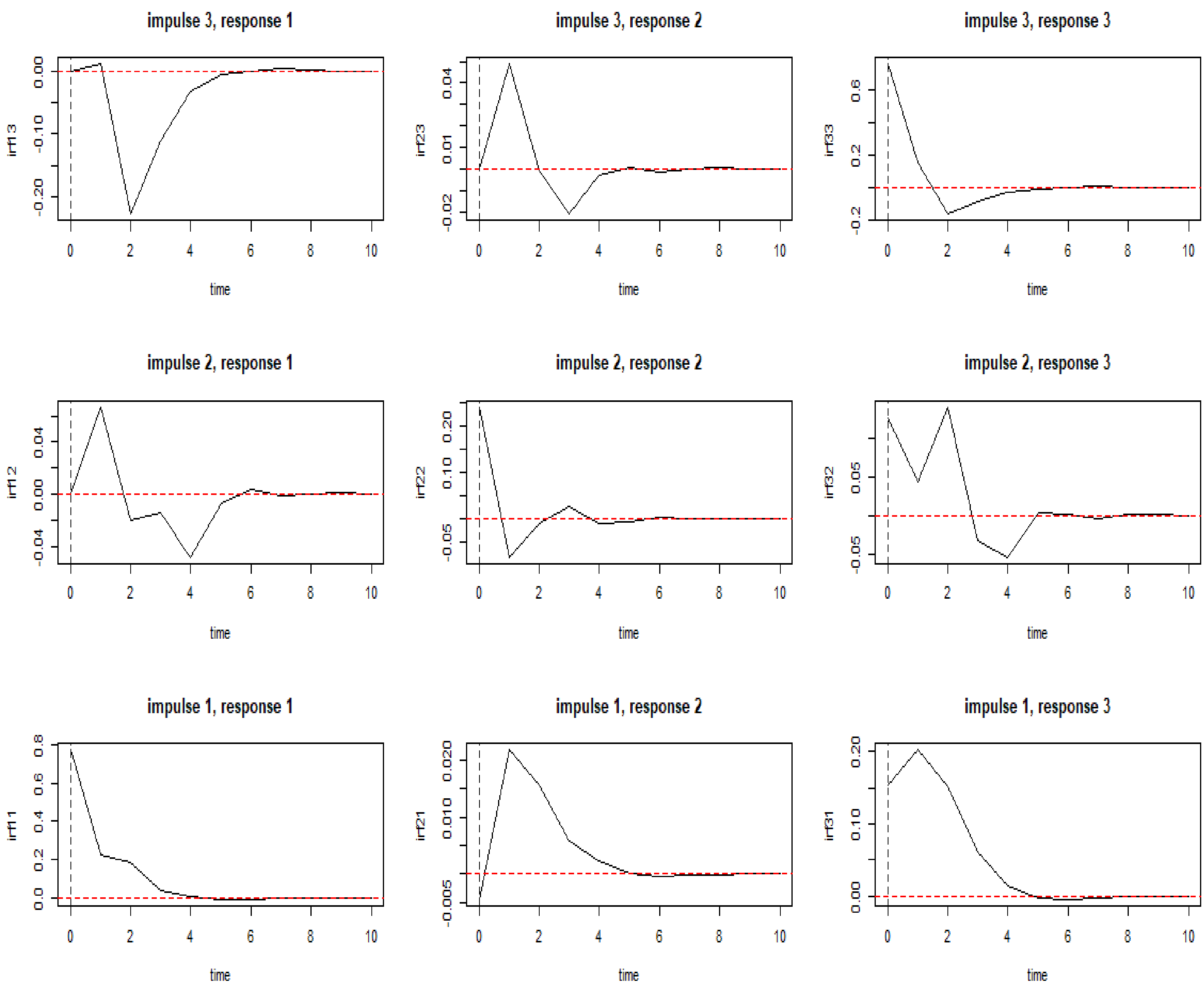

Figure11.IRF

Figure 11 shows the impact of an unanticipated monetary policy on the market's goods and pricing. Raising interest rates reduces consumption by discouraging investment and making saving more appealing.The first variable (GDPC96) is shown to react to a shock on the first, second, and third structural errors in the first row. Respectively the other lines.

CODE:

```
y<-data$GDPC96
p<-data$GDPCTPI
r<-data$FEDFUNDS
ygrowth <- rep(1,252)
pgrowth<- rep(1,252)
for (i in 2:252){
  ygrowth[i]<-(y[i]-y[i-1])/y[i-1]*100
  pgrowth[i]<-(p[i]-p[i-1])/p[i-1]*100
}
ygrowth<-ygrowth[2:252]
pgrowth<-pgrowth[2:252]
r<-r[2:252]
plot(ygrowth, type="l",col="darkred",main="GDPC96 Growth")
legend("topright",lty=1,col=c("darkred"),
       legend=c("growth"),cex = 0.6)
acf(ygrowth,col="darkred") # Generate the corellogram
legend("topright",lty=1,col=c("darkred"),
       legend=c("Autocorrelation"),cex = 0.6)
df_ygrowth_trend<-ur.df(ygrowth,type = "trend",selectlags = "AIC")
summary(df_ygrowth_trend)
plot(pgrowth, type="l",col="darkorchid3",main="GDPCTPI Growth")
legend("topright",lty=1,col=c("darkorchid3"),
       legend=c("growth"),cex = 0.6)
acf(pgrowth,col="darkorchid3")
legend("topright",lty=1,col=c("darkorchid3"),
       legend=c("Autocorrelation"),cex = 0.6)
df_pgrowth_trend<-ur.df(pgrowth,type="trend",selectlags = "AIC")
```

```
summary(df_pgrowth_trend)
df_pgrowth_drift<-ur.df(pgrowth,type = "drift", selectlags = "AIC")
summary(df_pgrowth_drift)
df_pgrowth_none<-ur.df(pgrowth, type = "none", selectlags = "AIC")
summary(df_pgrowth_none)
plot(r, type="l",col="darkgreen",main="FEDDFUNDS Growth")
legend("topright",lty=1,col=c("darkgreen"),
    legend=c("growth"),cex = 0.6)
acf(r,col="darkgreen")
legend("topright",lty=1,col=c("darkgreen"),
    legend=c("Autocorrelation"),cex = 0.6)
df_r_trend<-ur.df(r,type = "trend", selectlags = "AIC")
summary(df_r_trend)
df_r_drift<-ur.df(r,type = "drift", selectlags = "AIC")
summary(df_r_drift)
df_r_none<-ur.df(r,type = "none", selectlags = "AIC")
summary(df_r_none)
ygrowth<-ygrowth[2:251]
pgrowth<-tail(pgrowth,250) - head(pgrowth,250)
r<-tail(r,250) - head(r,250)
plot(pgrowth, type="l",col="darkorchid3",main="GDPCTPI Growth")
legend("topright",lty=1,col=c("darkorchid3"),
    legend=c("growth"),cex = 0.6)
acf(ygrowth,col="darkorchid3") # Generate the corellogram
legend("topright",lty=1,col=c("darkred"),
    legend=c("Autocorrelation"),cex = 0.6)
acf(pgrowth,col="darkorchid3") # Generate the corellogram
legend("topright",lty=1,col=c("darkorchid3"),
```

```
      legend=c("Autocorrelation"),cex = 0.6)


df_pgrowth_trend<-ur.df(pgrowth, type = "trend", selectlags = "AIC")
summary(df_pgrowth_trend)
plot(r, type="l",col="darkgreen",main="FEDDFUNDS Growth")
legend("topright",lty=1,col=c("darkgreen"),
      legend=c("growth"),cex = 0.6)
acf(r,col="darkgreen")
legend("topright",lty=1,col=c("darkgreen"),
      legend=c("Autocorrelation"),cex = 0.6)
df_r_trend<-ur.df(r,type = "trend", selectlags = "AIC")
summary(df_r_trend)
data<-cbind(ygrowth,pgrowth,r) # This is the data set that we will work with.
                              # Observe that the first variable is ygrowht.
                              # the second is pgrowth and the third is r.


colnames(data)<-NULL #erase column names
n<- dim(data)[1];n # number of observasions in my data set
k<- dim(data)[2];k # number of variables in my data set
optim_lag<-lag_select(12)
optim_lag # Both the SBIC and the HQIC suggest the we should set the number of
  p<-2
hor<-10 # The time horizon for the irf's
results<-ols_est(data,p,n)
bhat<-results$bhat;bhat
sigmahat<-results$sigmahat;sigmahat
resida<-results$resids
H<-t(chol(sigmahat));H # indentification of matrix H using Cholesky decomposition.
```

```
bhatt<-bhat[2:(k*p+1),]
companion<-cbind(diag(k*(p-1)),matrix(0,nrow =(k*(p-1)),ncol = k))
companion<-rbind(t(bhatt),companion);companion
irf<-irf_fun(companion,k,p,H,hor)
irf11<-irf[1,]
irf21<-irf[2,]
irf31<-irf[3,]
irf12<-irf[4,]
irf22<-irf[5,]
irf32<-irf[6,]
irf13<-irf[7,]
irf23<-irf[8,]
irf33<-irf[9,]
par(mfrow= c(3,3))
plot(0:hor,irf13,type="l",col="black",main="impulse 3, response 1", xlab="time")
abline(h=0, col="red", lty="dashed")
abline(v=0,col="black" , lty="dashed")
plot(0:hor,irf23,type="l",col="black",main="impulse 3, response 2", xlab="time")
abline(h=0, col="red", lty="dashed")
abline(v=0,col="black" , lty="dashed")
plot(0:hor,irf33,type="l",col="black",main="impulse 3, response 3", xlab="time")
abline(h=0, col="red", lty="dashed")
abline(v=0,col="black" , lty="dashed")
plot(0:hor,irf12,type="l",col="black",main="impulse 2, response 1", xlab="time")
abline(h=0, col="red", lty="dashed")
abline(v=0,col="black" , lty="dashed")
plot(0:hor,irf22,type="l",col="black",main="impulse 2, response 2", xlab="time")
abline(h=0, col="red", lty="dashed")
```

```
abline(v=0,col="black" , lty="dashed")

plot(0:hor,irf32,type="l",col="black",main="impulse 2, response 3", xlab="time")

abline(h=0, col="red", lty="dashed")

abline(v=0,col="black" , lty="dashed")

plot(0:hor,irf11,type="l",col="black",main="impulse 1, response 1", xlab="time")

abline(h=0, col="red", lty="dashed")

abline(v=0,col="black" , lty="dashed")

plot(0:hor,irf21,type="l",col="black",main="impulse 1, response 2", xlab="time")

abline(h=0, col="red", lty="dashed")

abline(v=0,col="black" , lty="dashed")

plot(0:hor,irf31,type="l",col="black",main="impulse 1, response 3", xlab="time")

abline(h=0, col="red", lty="dashed")

abline(v=0,col="black" , lty="dashed")
```

Functions

Lag selection function:

```
lag_select<- function(pmax){

 mat<-matrix(NA,nrow = 3,ncol = pmax)

 for (p in 1:pmax){

    sigmahat<-ols_est(data,p,n)$sigmahat

    kk<-(k^2)*p +k  #total number of parameters in all equations

   AIC<-log(det(sigmahat)) + 2*kk/n

   SBIC<-log(det(sigmahat)) + kk*log(n)/n

   HQIC<- log(det(sigmahat)) + 2*kk*log(log(n))/n

   IC<-rbind(AIC,SBIC,HQIC)

   mat[,p]<-IC


 }
```

```
  score<-matrix(apply(mat,1,which.min))
  colnames(score)<- "number of lags"
  rownames(score)<- c("AIC","SBIC","HQIC")
   return(score)

}
```

OLS:

```
ols_est<-function(data,p,n){

  Y<-data[(p+1):n,]
  X<-matrix(1,nrow = n-p,ncol = 1)  #add intercepts to the model
  for(i in 1:p){
   X<-cbind(X,data[(p+1-i):(n-i),])
  }

  bhat<-solve(t(X) %*% X) %*% t(X) %*% Y
  resids<-Y-X%*%bhat
  sigmahat<-var(resids)
  output<- list(bhat=bhat, resids=resids, sigmahat=sigmahat)
  return(output)

}
```

IRF :

```
irf_fun<- function(companion,k,p,H,hor){

  J<-cbind(diag(k),matrix(0,nrow = k,ncol = (k*(p-1))))
```

```
  irf<-matrix(H,nrow=k^2,ncol = 1)

  for (j in 1:hor){

    A1<-J%*%(companion%^%j)%*%t(J)%*%H
    A2<-matrix(A1,nrow = k^2,ncol = 1)
    irf<-cbind(irf,A2)

  }
  return(irf)

}
```