Eric Randall
Predict 413 - Week 4

In this paper we will be looking at the duration of unemployment in the U.S. between January 1948 to March 2014 from the Federal Reserve Bank of St Louis. The data is measured in weeks and is seasonally adjusted.

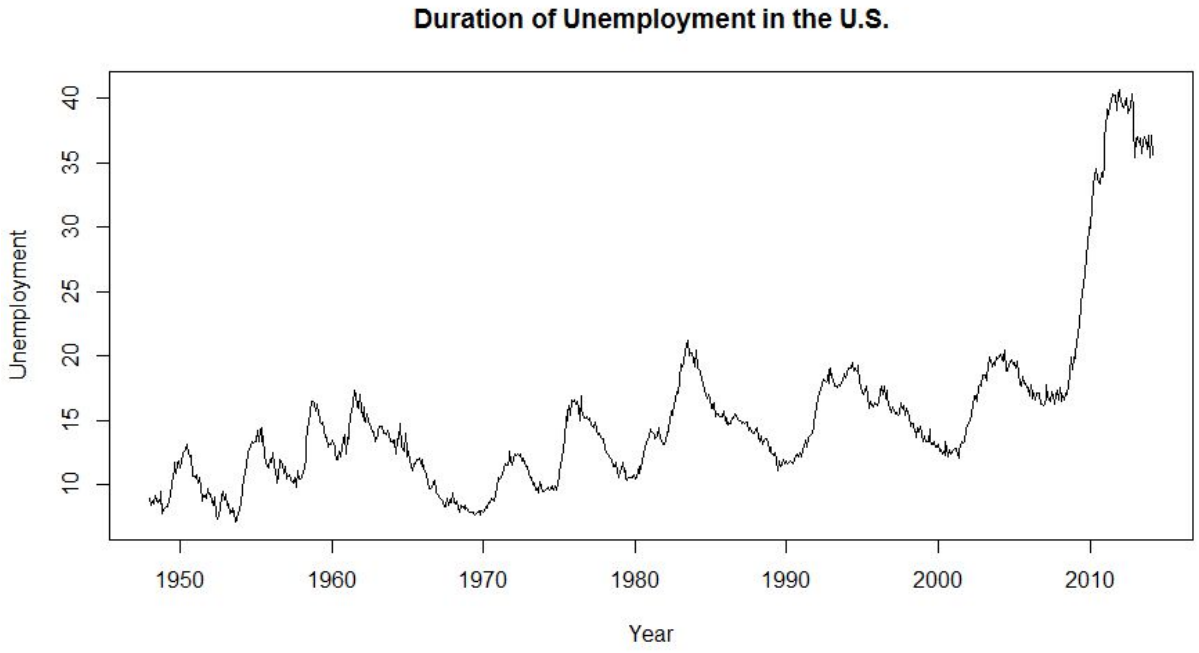

Figure 1: Monthly Duration of Unemployment

In order to determine whether differencing was required, we used a unit root test. First we used an Augmented Dickey-Fuller (ADF) test which returned a p value of 0.5637, over the .05 threshold, and therefore we determined that differencing was required. We also tested using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, which reverses the hypothesis. With a p-value of .01, it confirms the ADF test that we need to difference the data.

```
> adfTest(employ.ts,lags=12,type="c")

Title:
 Augmented Dickey-Fuller Test

Test Results:
  PARAMETER:
    Lag Order: 12
  STATISTIC:
    Dickey-Fuller: -1.3297
  P VALUE:
    0.5637
```

```
> kpss.test(employ.ts)

        KPSS Test for Level Stationarity

data:  employ.ts
KPSS Level = 5.1875, Truncation lag parameter = 6, p-value = 0.01
```

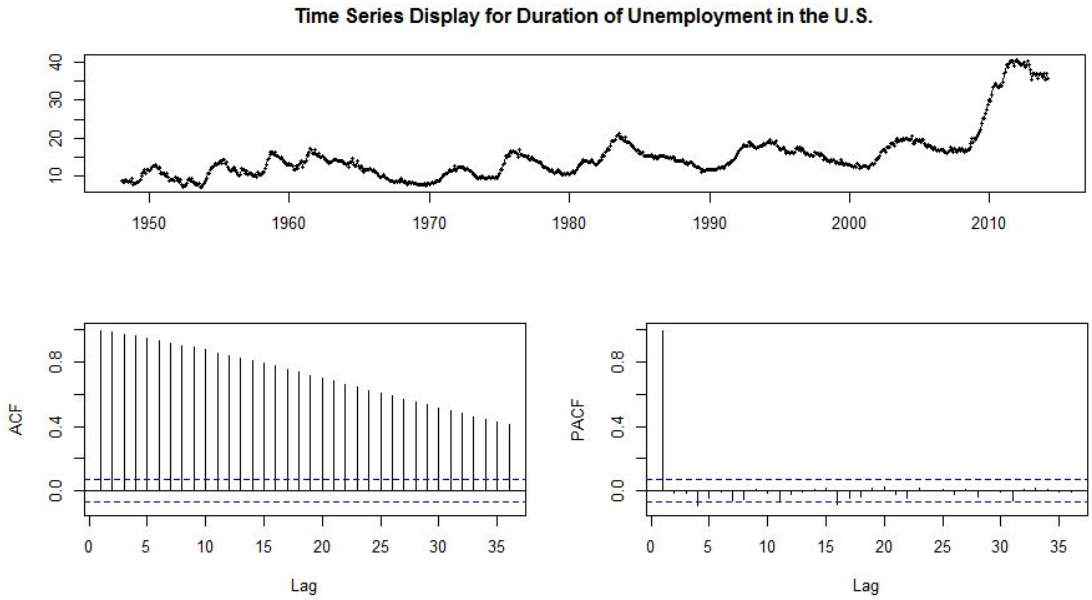

Figure 2: Time Series display

We can confirm our findings from the Augmented Dickey-Fuller test and the KPSS test for Level Stationarity as non-stationary data visually by looking at the decreasing ACF over the number of lags.

Comparing the above chart to the differenced data below we can see that some of the lags are still above the level of significance through the 9th order but that the descending pattern is eliminated.

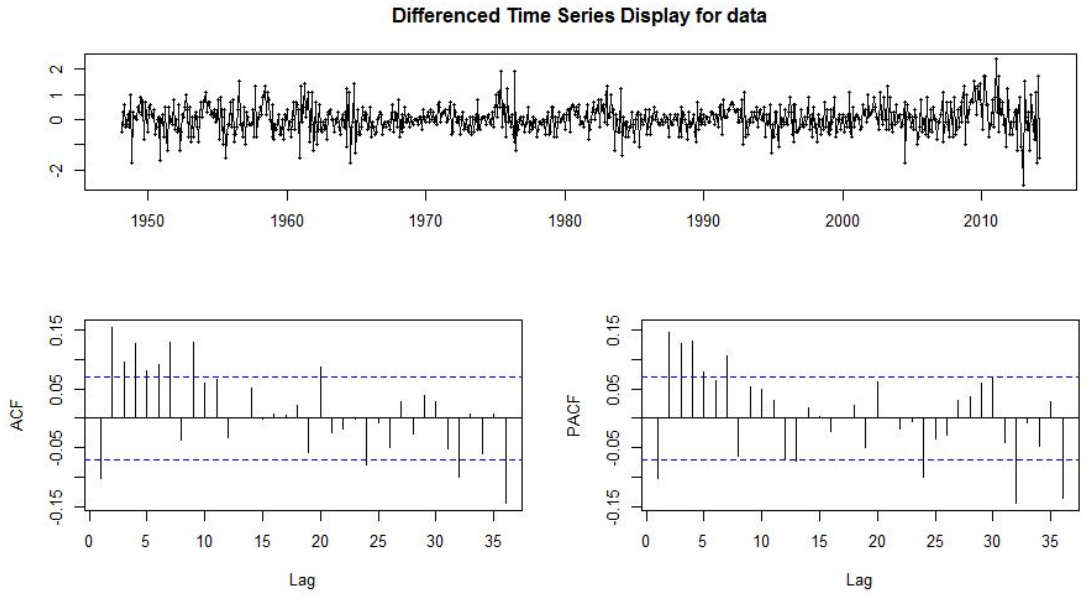

Figure 3: Differenced Time Series display

With a p-value of 0.0992, we can fail to reject the null hypothesis that the mean of the change of the differenced data is 0.

```
> t.test(demploy)

      One Sample t-test

data:  demploy
t = 1.6507, df = 793, p-value = 0.0992
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.006362152  0.073616560
sample estimates:
mean of x
```

Our first Arima model was built. The model is:

$$y_t = -0.1351y_{t-1}+0.1134y_{t-2}+0.1083y_{t-3}+0.1165y_{t-4}+0.0738y_{t-5}+0.0770y_{t-6}$$
$$+0.0903y_{t-7}-0.0609y_{t-8}+0.0687y_{t-9}+0.0609y_{t-10}+0.0228y_{t-11}-0.0723y_{t-12}$$

```
> model1 <- arima(demploy,order=c(12,0,0),include.mean=F)
> model1

Call:
arima(x = demploy, order = c(12, 0, 0), include.mean = F)

Coefficients:
          ar1     ar2     ar3     ar4     ar5     ar6     ar7      ar8     ar9
ar10    ar11     ar12
      -0.1351  0.1134  0.1083  0.1165  0.0738  0.0770  0.0903  -0.0609  0.0687
0.0609  0.0228  -0.0723
s.e.   0.0355  0.0358  0.0361  0.0363  0.0365  0.0366  0.0365   0.0366  0.0365
0.0366  0.0364   0.0362

sigma^2 estimated as 0.2968:  log likelihood = -644.61,  aic = 1315.22
```

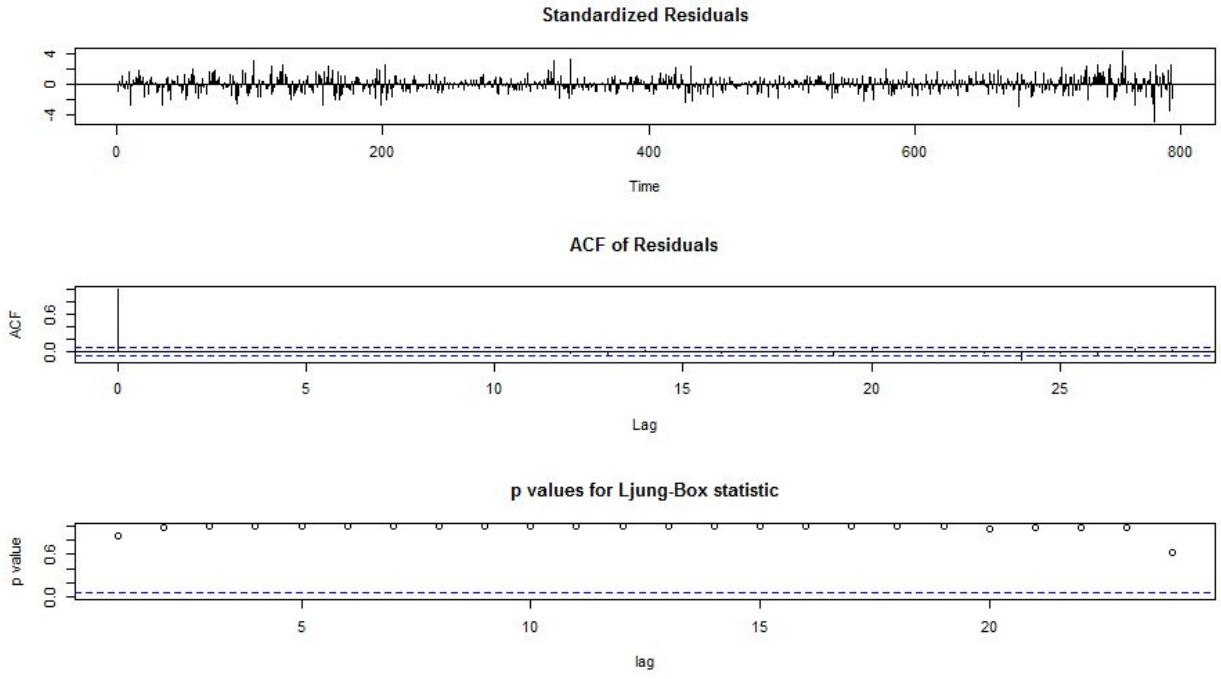

Figure 4: Residual analysis for model1

Using the ACF of Residuals above we see that residuals are uncorrelated, and using the standardized residuals, that the residuals have a zero mean. Therefore the forecasts should meet assumptions and be unbiased.

Next we are fitting a seasonal ARIMA(2,0,1)(1,0,1)$_{12}$ model.

```
Call:
arima(x = demploy, order = c(2, 0, 1), seasonal = list(order = c(1, 0, 1), period =
12),
    include.mean = FALSE)

Coefficients:
         ar1     ar2      ma1    sar1     sma1
      0.6538  0.2637  -0.8022  0.5662  -0.7429
s.e.  0.0478  0.0360   0.0382  0.0755   0.0585

sigma^2 estimated as 0.2926:  log likelihood = -639.43,  aic = 1290.85
```

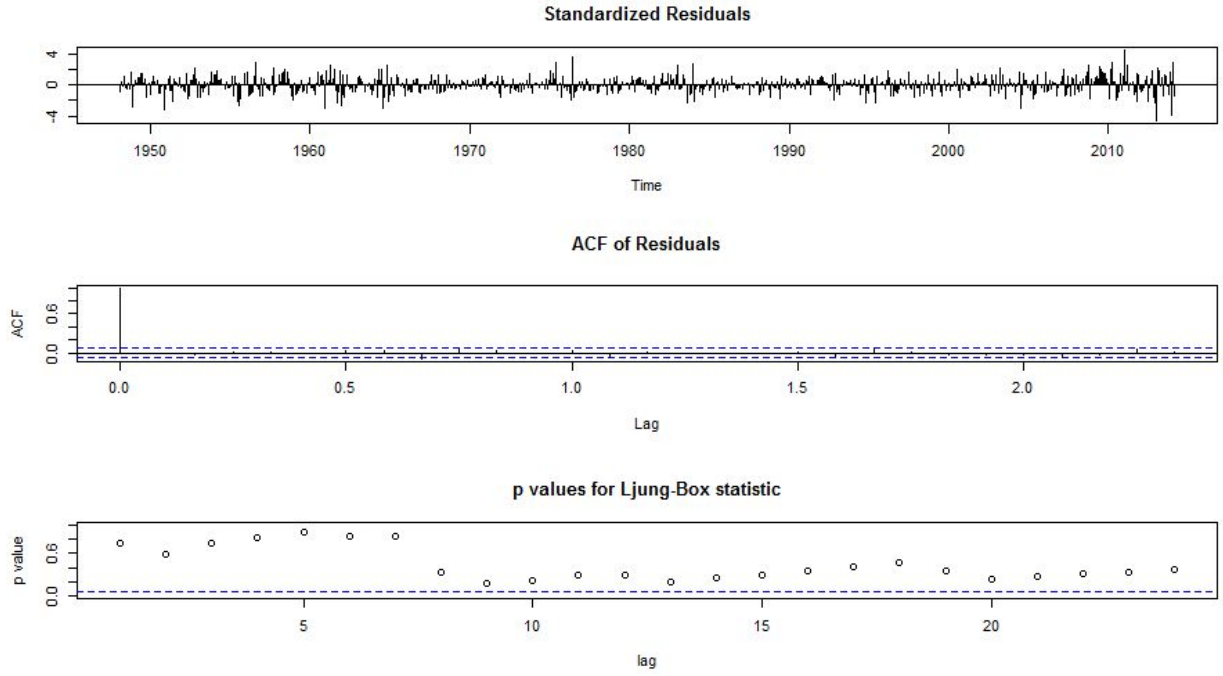

Figure 5: Residual analysis for model2

As we can see the residuals are stationary and acting as white noise as confirmed visually on the ACF and the p values for the Ljung-Box statistic confirm this.

Comparing between our two models, the second model has the lower AIC value. Furthermore, the RMSE is lower between the two:

```
> accuracy(model1)
                     ME      RMSE       MAE MPE MAPE      MASE         ACF1
Training set 0.01766675 0.544767 0.4140733 NaN  Inf 0.6530631 -0.006053442
> accuracy(model2)
                     ME      RMSE      MAE MPE MAPE      MASE        ACF1
Training set 0.02208027 0.5409351 0.405764 NaN  Inf 0.6399579 -0.01164798
```

Backtesting between the two models we can further confirm that the second model is superior with the lower RMSE of out-of-sample forecast figure of 0.9440497.

```
> backtest(model1,demploy,750,1,inc.mean=F)
[1] "RMSE of out-of-sample forecasts"
[1] 0.9719466
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.7655835

> backtest(model2,demploy,750,1,inc.mean=F)
[1] "RMSE of out-of-sample forecasts"
[1] 0.9440497
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.7391323
```

In this portion of the paper we are considering weekly crude oil prices from West Texas Intermediate (WTI) in Cushing, Oklahoma from January 3, 1986 to April 2, 2014.

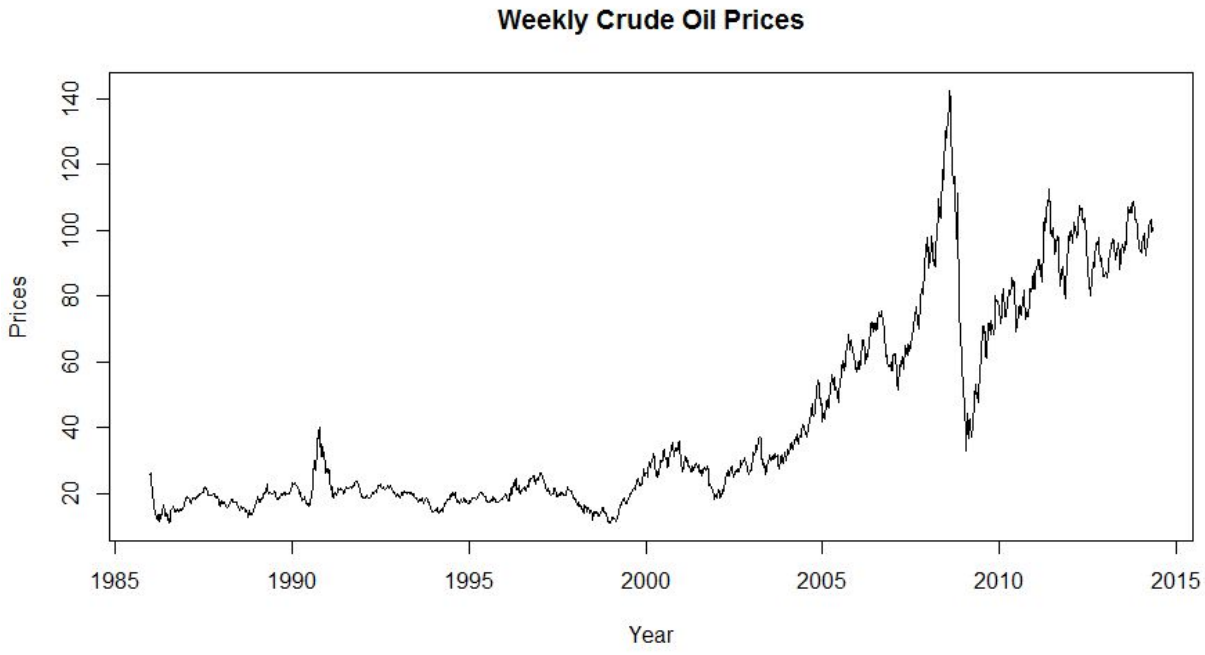

Figure 6: Weekly Crude Oil Prices 1986-2014

As we can see there is a strong increase starting in 2002 and increasing until the end of the dataset amid a weaker cyclical pattern. Seasonal patterns exist but are minute compared to the overall trend.

We will be looking at the differenced logs of the Weekly Crude Oil prices data.

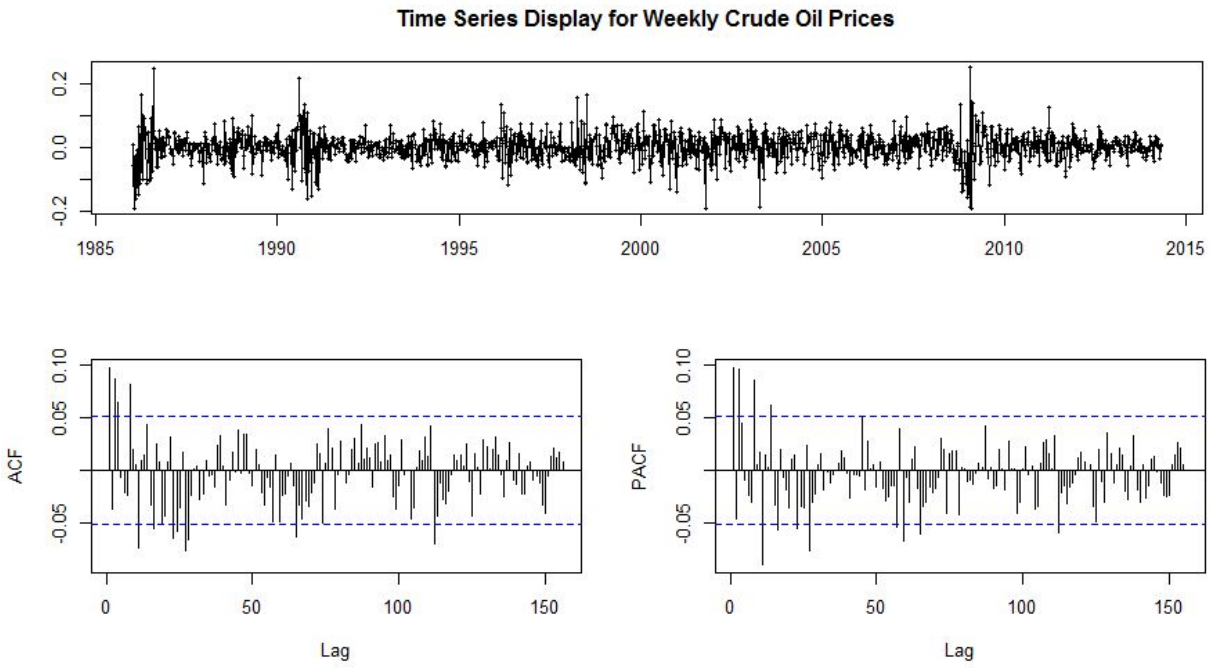

Figure 7: Diagnostics for the log differenced data

Using a Ljung-Box test we see that get we get a very small p-value of 1.742e-06, so we cannot confirm that the data is behaving as white noise.

```
> Box.test(dlcoil,lag=10,type='Ljung')

        Box-Ljung test

data:  dlcoil
X-squared = 45.5333, df = 10, p-value = 1.742e-06
```

Next we will make an ARIMA(11,0,0) model.

```
> coil_model2 <- arima(dlcoil,order=c(11,0,0))
> coil_model2

Call:
arima(x = dlcoil, order = c(11, 0, 0))

Coefficients:
         ar1      ar2     ar3     ar4      ar5      ar6      ar7     ar8      ar9
ar10     ar11  intercept
      0.1055  -0.0513  0.1054  0.0421  -0.0192  -0.0160  -0.0381  0.0980  -0.0041
0.0303  -0.0942     0.0009
s.e.  0.0259   0.0261  0.0263  0.0263   0.0264   0.0264   0.0265  0.0265   0.0266
0.0266   0.0264     0.0013

sigma^2 estimated as 0.001827:  log likelihood = 2553.47,  aic = -5080.93
```

The model is:

$$y_t = 0.0009 + 0.1055y_{t-1}-0.0513y_{t-2}+0.1054y_{t-3}+0.0421y_{t-4}-0.0192y_{t-5}-0.0160y_{t-6} -0.0381y_{t-7}+0.0980y_{t-8}-0.0041y_{t-9}+0.0303y_{t-10}-0.0942y_{t-11}$$

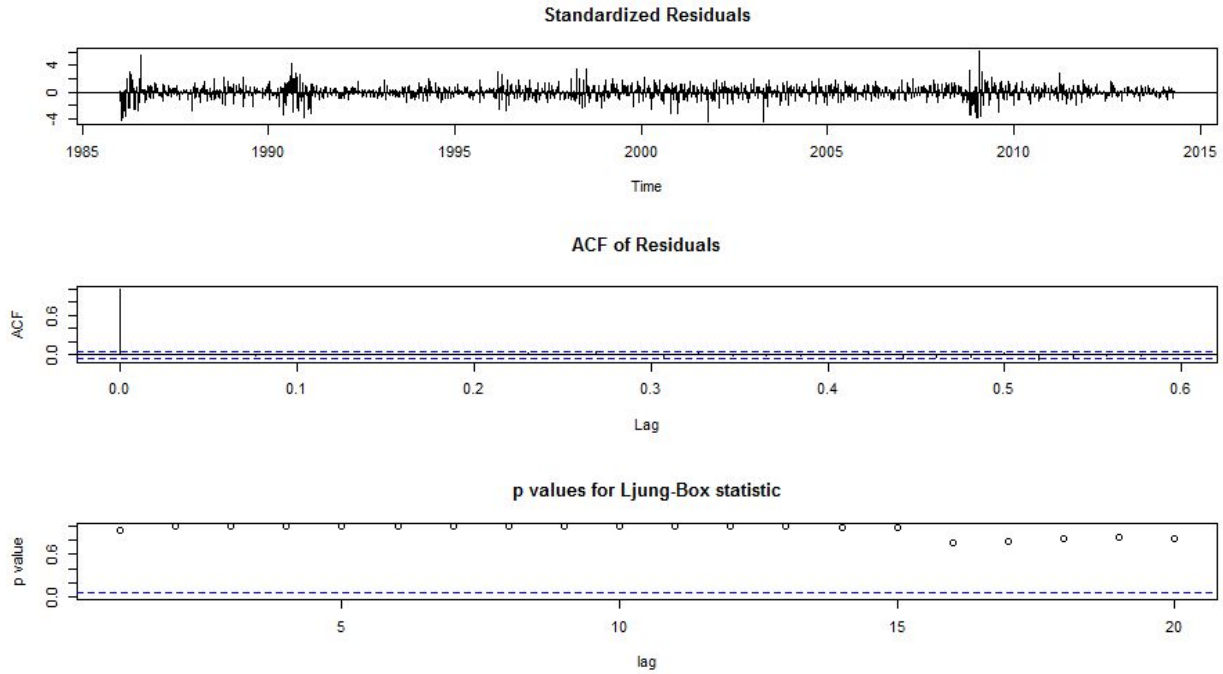

Figure 8: Diagnostics for the ARIMA(11,0,0) model

As we can see the residuals are stationary and acting as white noise as confirmed visually on the ACF and the p values for the Ljung-Box statistic confirm this. This meets all the assumptions given by the model.

```
> Box.test(residuals(coil_model2),type='Ljung')

        Box-Ljung test

data:  residuals(coil_model2)
X-squared = 0.0041, df = 1, p-value = 0.9488
```

Next we are fitting an ARIMA(3,0,2) model.

```
> coil_model3 <- arima(rtn,order=c(3,0,2),include.mean=FALSE)
> coil_model3

Call:
arima(x = rtn, order = c(3, 0, 2), include.mean = FALSE)

Coefficients:
         ar1      ar2     ar3      ma1     ma2
      0.5664  -0.8548  0.1689  -0.4680  0.7753
s.e.  0.0934   0.0681  0.0270   0.0931  0.0680

sigma^2 estimated as 0.001845:  log likelihood = 2546.4,  aic = -5080.8
```

The model is:

$$y_t = .5664y_{t-1}-0.8548y_{t-2}+0.1689y_{t-3}-0.4680e_{t-1}+0.7753e_{t-2}$$

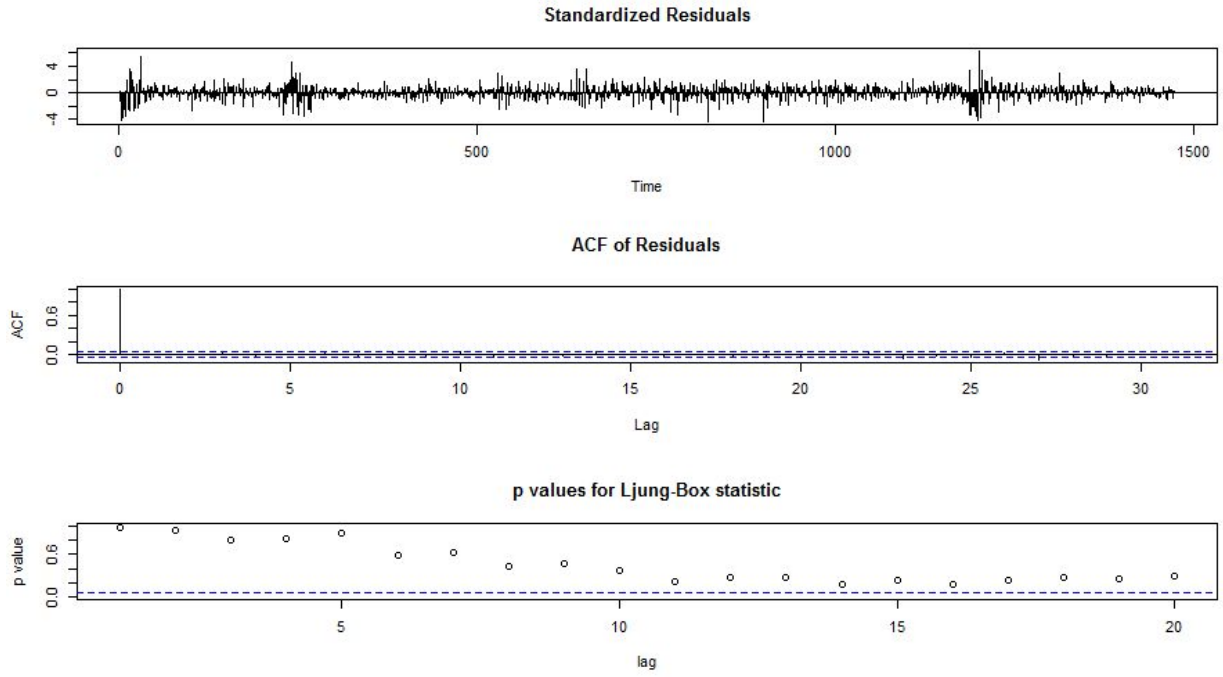

Figure 8: Diagnostics for the ARIMA(3,0,2) model

And checking with a Ljung-box we see that the residuals are acting as white noise.

```
> Box.test(residuals(coil_model3),type='Ljung')

        Box-Ljung test

data:  residuals(coil_model3)
X-squared = 5e-04, df = 1, p-value = 0.9824
```

Comparing between the two we prefer the ARIMA(11,0,0) model due to the slightly lower RMSE.

```
> accuracy(coil_model2)
                      ME       RMSE        MAE MPE MAPE     MASE        ACF1
Training set 7.779306e-06 0.04274386 0.03096273 NaN  Inf 0.742659 0.001670274
> accuracy(coil_model3)
                      ME       RMSE        MAE MPE MAPE      MASE         ACF1
Training set 0.0007899324 0.04295061 0.03105575 NaN  Inf 0.7448904 0.0005752461
```

Code:

```
library(fBasics)
library(sjPlot)
library(fpp)
library(fUnitRoots)
source("backtest.R")
setwd("C:/Users/00811289/Desktop/413/week4")

# Section 1

employ <- read.table("m-unempmean.txt",header=T)
head(employ)

# Plot EDA
employ.ts <- ts(employ$Value, start=c(1948,1),end=c(2014,3), frequency=12)
plot(employ.ts, main="Duration of Unemployment in the U.S.", xlab="Year",
ylab="Unemployment")
monthplot(employ.ts)
plot(stl(employ.ts, s.window="periodic"))

# Unit Root Test
adf.test(employ.ts, alternative = "stationary")
adfTest(employ.ts,lags=12,type="c")
kpss.test(employ.ts)
tsdisplay(employ.ts,main="Time Series Display for Duration of Unemployment in the
U.S.")

# Difference data
demploy <- diff(employ.ts)

# Test stationarity
tsdisplay(demploy,main="Differenced Time Series Display for data")
t.test(demploy)

# Create/Diagnose AR model
model1 <- arima(demploy,order=c(12,0,0),include.mean=FALSE)
model1
tsdiag(model1,gof=24)

# Create a seasonal model
model2 <-
arima(demploy,order=c(2,0,1),seasonal=list(order=c(1,0,1),period=12),include.mean=FAL
SE)
model2
tsdiag(model2,gof=24)

accuracy(model1)
```

```
accuracy(model2)

backtest(model1,demploy,750,1,inc.mean=F)
backtest(model2,demploy,750,1,inc.mean=F)

# Section 2
coil <- read.table("w-coilwtico.txt",header=T)
head(coil)
tail(coil)
plot(coil$Value)

# Plot EDA
coil.ts <- ts(coil$Value, start=c(1986,1), frequency=52)
plot(coil.ts, main="Weekly Crude Oil Prices", xlab="Year", ylab="Prices")
plot(stl(coil.ts, s.window="periodic"))
tsdisplay(coil.ts,main="Time Series Display for Weekly Crude Oil Prices")

# Convert to first difference of log oil prices
dlcoil <- diff(log(coil.ts))
plot(dlcoil,type='l')
tsdisplay(dlcoil,main="Time Series Display for Weekly Crude Oil Prices")

# Check serial correlation
Box.test(dlcoil,lag=10,type='Ljung')

# Check order and create ARIMA model
coil_model1 <- ar(dlcoil, method="mle")
coil_model1$order #11

coil_model2 <- arima(dlcoil,order=c(11,0,0))
coil_model2

tsdiag(coil_model2, gof=20)
Box.test(residuals(coil_model2),type='Ljung')

# Create ARIMA(3,0,2) model
coil_model3 <- arima(rtn,order=c(3,0,2),include.mean=FALSE)
coil_model3
tsdiag(coil_model3,gof=20)
Box.test(residuals(coil_model3),type='Ljung')

# Compare fit between the models
accuracy(coil_model2)
accuracy(coil_model3)
```