India Index of Industrial Production data Analysis

Kiran Lakhchaura

3/3/2021

In this project we will anlyze India's monthly Index of Industrial Production (IIP) data since Apr. 2012.

Reading the data

```
df <- read.csv(file =
                 'Data/Monthly_indices_of_industrial_production_as_per_use-based_classification.csv',
sep=';')
```

Viewing the data

```
head(df)
```

```
##   MonYear Primary.goods Capital.goods Intermediate.goods
## 1  Apr.12          98.2          89.5              103.6
## 2  May.12         104.0          98.7              105.6
## 3  Jun.12          99.7         101.9              103.5
## 4  Jul.12          98.9          97.0              102.7
## 5  Aug.12          96.5         101.9              103.0
## 6  Sep.12          92.4         102.8              104.6
##   Infrastructure..construction.goods Consumer.durables Consumer.non.durables
## 1                              103.1             102.0                  97.0
## 2                              117.4             105.1                  99.8
## 3                              102.2             104.9                 104.9
## 4                              106.0             102.2                 103.7
## 5                               97.2             101.1                 103.8
## 6                               98.8             107.2                  98.2
```

changing the type of the MonYear column from character to YearMon

```
df$MonYear<- as.yearmon(df$MonYear,'%b.%y')
head(df)
```

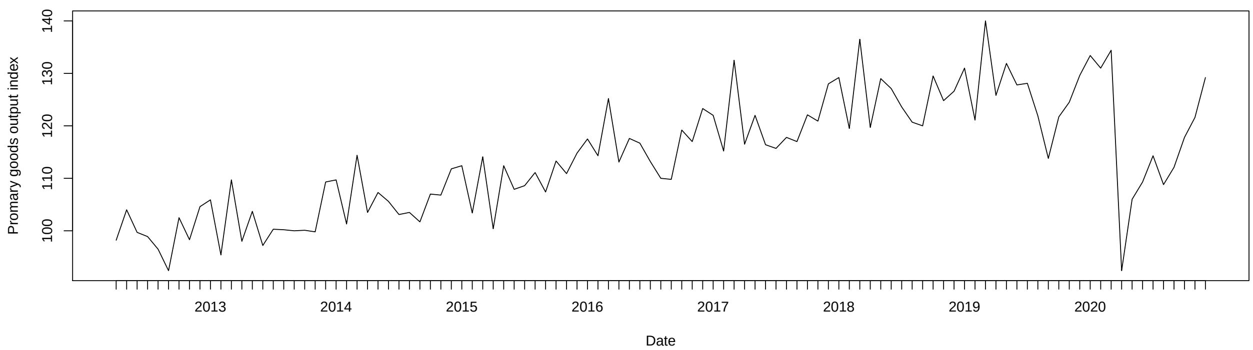
```
##    MonYear Primary.goods Capital.goods Intermediate.goods
## 1 Apr 2012          98.2          89.5              103.6
## 2 May 2012         104.0          98.7              105.6
## 3 Jun 2012          99.7         101.9              103.5
## 4 Jul 2012          98.9          97.0              102.7
```

```
## 5 Aug 2012          96.5          101.9               103.0
## 6 Sep 2012          92.4          102.8               104.6
##   Infrastructure..construction.goods Consumer.durables Consumer.non.durables
## 1                              103.1             102.0                  97.0
## 2                              117.4             105.1                  99.8
## 3                              102.2             104.9                 104.9
## 4                              106.0             102.2                 103.7
## 5                               97.2             101.1                 103.8
## 6                               98.8             107.2                  98.2
```

Plotting the data

The data contains many fields. Here, we will be looking at just the primary goods output index. Let's plot the primary goods column from the dataframe.

```r
plot(df$Primary.goods~df$MonYear,type="l",xlab="Date",ylab="Promary goods output index")
```



The data does show clear trend and seasonality, although there is a clear anomaly seen as a sharp drop around March. 2020. and if we look carefully even around Sep. 2019 there was an anomalous drop. For this reason for simplicity of the analysis we will restrict our analysis till mar. 2019 (i.e. 7 years of data)

```r
df <- df[1:84,]
nrow(df)
```

```
## [1] 84
```

Updated plot

```r
plot(df$Primary.goods~df$MonYear,type="l",xlab="Date",ylab="Promary goods output index")
```

Stationarity tests

Let's check for the stationarity of data using ADF test

```
data <- df$Primary.goods
print(adf.test(data))
```

```
## Warning in adf.test(data): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  data
## Dickey-Fuller = -4.958, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

ADF test p-value suggests that the null-hypothesis of unit-root (non-statinarity) should be rejected => data is stationary.

Checking for trend stationarity using KPSS test

```
print(kpss.test(data, null = c("Trend"), lshort = TRUE))
```

```
## Warning in kpss.test(data, null = c("Trend"), lshort = TRUE): p-value greater
## than printed p-value

##
##  KPSS Test for Trend Stationarity
##
## data:  data
## KPSS Trend = 0.060362, Truncation lag parameter = 3, p-value = 0.1
```

KPSS p-value indicates that the null hypothesis of trend stationarity cannot be rejected.

Let's do the PP unit-root test now

```
print(pp.test(data, lshort=TRUE))
```

```
## Warning in pp.test(data, lshort = TRUE): p-value smaller than printed p-value

##
##  Phillips-Perron Unit Root Test
```

```
##
## data:  data
## Dickey-Fuller Z(alpha) = -117.03, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

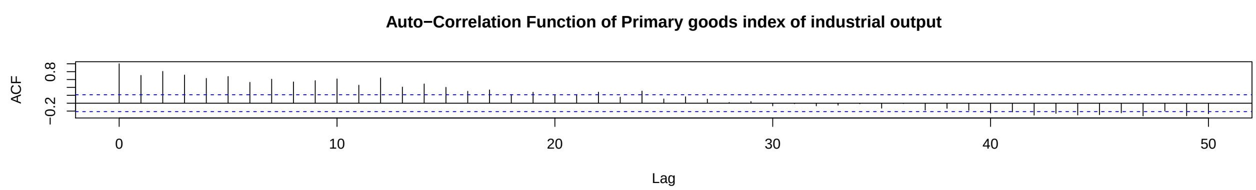
The PP unit root test suggests that the null hypothesis of unit-root (non-stationarity) should be rejected => data is stationary.

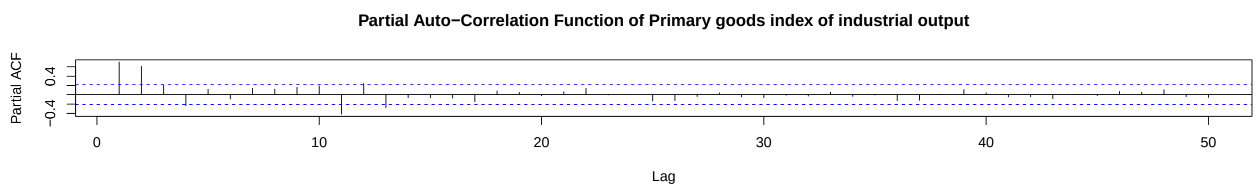
So all three tests indicate that the data is stationary.

Auto-Correlation Functions

Now let's look at the auto-correlations in the ACF and PACF plots

```r
data <- df$Primary.goods
par(mfrow=c(2,1))
acf(data,50,main='Auto-Correlation Function of Primary goods index of industrial output')
pacf(data,50,main='Partial Auto-Correlation Function of Primary goods index of industrial output')
```





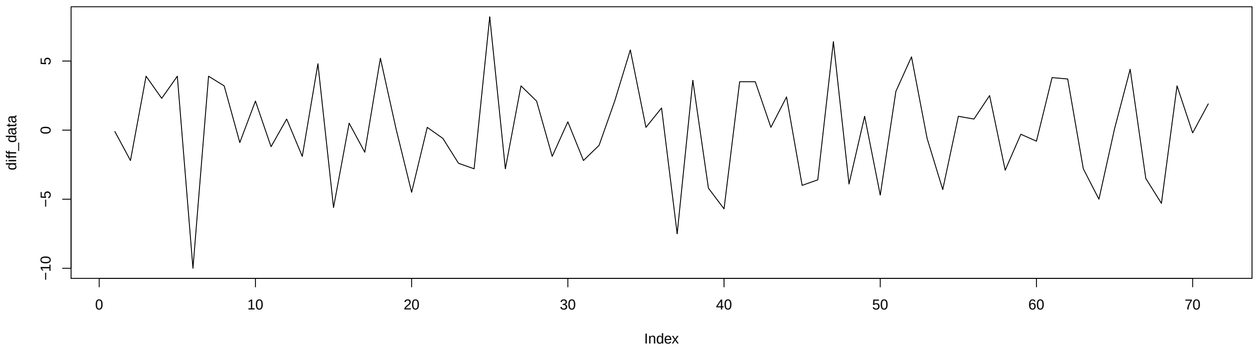
There are significant correlations in the ACF plot upto lag=25 and PACF plot also shows significant correlations upto lag=13.

Guessing the right orders for (S)ARIMA model fitting

1. Differencing orders (d, D)

Since we do see clear seasonality and trend in the data, we should look at the differenced data using both seasonal as well as non-seasonal differencing -> *diff(diff(data),12)*

```r
diff_data <- diff(diff(data),12)
plot(diff_data,type="l")
```



The data looks almost stationary (except for the drop in the last part) with no clear trend, seasonality or change in variation. Now let's test the differenced data with the ADF test.

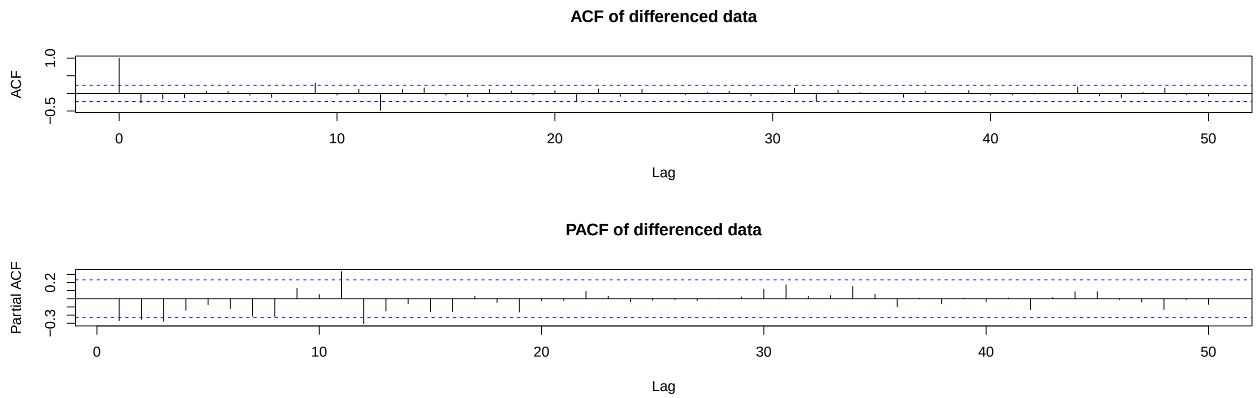
```
print(adf.test(diff_data))
```

```
## Warning in adf.test(diff_data): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_data
## Dickey-Fuller = -5.4282, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

Now let's look at the ACF and PACF plots for the differenced data

```
par(mfrow=c(2,1))
acf(diff_data,50,main='ACF of differenced data')
pacf(diff_data,50,main='PACF of differenced data');
```



We see that the correlations have reduced significantly. In the ACF plot there is significant correlation only at lag=12 which might be due to seasonality and in PACF plot also al the correlations are really small. => d=1, D=1

2. Finding the best orders for the auro-regressive (AR; p, P) and Moving Average (MA; q, Q) terms

ACF and PACF for differenced data

Trying for different values of p,q,P,Q and note down AIC, SSE and p-value (for Ljun-box-test). We want high p-values and small AIC and SSE using parsimony principle (simpler the better) while searching

```
d=1; DD=1; per=12

for(p in 1:4){
  for(q in 1:4){
    for(i in 1:4){
      for(j in 1:4){
        if(p+d+q+i+DD+j<=10){

          model<-arima(x=data, order = c((p-1),d,(q-1)), seasonal = list(order=c((i-1),DD,(j-1)), perio

          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))

          sse<-sum(model$residuals^2)

          cat(p-1,d,q-1,i-1,DD,j-1,per, 'AIC=', model$aic, ' SSE=',sse,' p-VALUE=', pval$p.value,'\n')

        }
      }
    }
  }
}
```

```
## 0 1 0 0 1 0 12 AIC= 383.8146  SSE= 900.2041  p-VALUE= 0.05805043
## 0 1 0 0 1 1 12 AIC= 358.4238  SSE= 516.7009  p-VALUE= 0.04520712
## 0 1 0 0 1 2 12 AIC= 359.6614  SSE= 539.7359  p-VALUE= 0.04012424
## 0 1 0 0 1 3 12 AIC= 361.1897  SSE= 504.3696  p-VALUE= 0.03845047
## 0 1 0 1 1 0 12 AIC= 361.6413  SSE= 597.3847  p-VALUE= 0.04185224
## 0 1 0 1 1 1 12 AIC= 359.4027  SSE= 537.7315  p-VALUE= 0.03835106

## Warning in arima(x = data, order = c((p - 1), d, (q - 1)), seasonal = list(order
## = c((i - : possible convergence problem: optim gave code = 1

## 0 1 0 1 1 2 12 AIC= 358.0764  SSE= 459.7463  p-VALUE= 0.03916853

## Warning in arima(x = data, order = c((p - 1), d, (q - 1)), seasonal = list(order
## = c((i - : possible convergence problem: optim gave code = 1

## 0 1 0 1 1 3 12 AIC= 359.7837  SSE= 425.6699  p-VALUE= 0.04440311
## 0 1 0 2 1 0 12 AIC= 361.2438  SSE= 568.5331  p-VALUE= 0.03891582
## 0 1 0 2 1 1 12 AIC= 360.8968  SSE= 460.1147  p-VALUE= 0.03818458
## 0 1 0 2 1 2 12 AIC= 359.9201  SSE= 442.7144  p-VALUE= 0.04206669
## 0 1 0 3 1 0 12 AIC= 359.3255  SSE= 506.6389  p-VALUE= 0.03679709

## Warning in arima(x = data, order = c((p - 1), d, (q - 1)), seasonal = list(order
## = c((i - : possible convergence problem: optim gave code = 1

## 0 1 0 3 1 1 12 AIC= 357.0338  SSE= 359.4335  p-VALUE= 0.03669721
## 0 1 1 0 1 0 12 AIC= 371.9595  SSE= 733.5683  p-VALUE= 0.1767264
## 0 1 1 0 1 1 12 AIC= 343.2273  SSE= 388.4652  p-VALUE= 0.4063213
## 0 1 1 0 1 2 12 AIC= 343.8299  SSE= 412.1013  p-VALUE= 0.5328783
## 0 1 1 0 1 3 12 AIC= 344.769  SSE= 341.8211  p-VALUE= 0.4466607
## 0 1 1 1 1 0 12 AIC= 345.748  SSE= 453.9052  p-VALUE= 0.4084277
## 0 1 1 1 1 1 12 AIC= 343.1925  SSE= 407.4021  p-VALUE= 0.5434732

## Warning in arima(x = data, order = c((p - 1), d, (q - 1)), seasonal = list(order
```

```
## = c((i - : possible convergence problem: optim gave code = 1
```

```
## 0 1 1 1 1 2 12 AIC= 340.6104  SSE= 329.6658  p-VALUE= 0.4508023
## 0 1 1 2 1 0 12 AIC= 345.4537  SSE= 434.0246  p-VALUE= 0.5339025
## 0 1 1 2 1 1 12 AIC= 344.1827  SSE= 344.866  p-VALUE= 0.4644096
## 0 1 1 3 1 0 12 AIC= 341.5364  SSE= 363.1433  p-VALUE= 0.5605429
## 0 1 2 0 1 0 12 AIC= 367.2751  SSE= 660.269  p-VALUE= 0.9422096
## 0 1 2 0 1 1 12 AIC= 341.6945  SSE= 374.7903  p-VALUE= 0.9722501
## 0 1 2 0 1 2 12 AIC= 342.7717  SSE= 394.9682  p-VALUE= 0.989831
## 0 1 2 1 1 0 12 AIC= 343.5432  SSE= 428.4052  p-VALUE= 0.9949983
## 0 1 2 1 1 1 12 AIC= 342.1622  SSE= 392.6753  p-VALUE= 0.9912465
## 0 1 2 2 1 0 12 AIC= 344.0225  SSE= 415.1881  p-VALUE= 0.9982206
## 0 1 3 0 1 0 12 AIC= 368.2174  SSE= 647.4333  p-VALUE= 0.9916728
## 0 1 3 0 1 1 12 AIC= 343.6943  SSE= 374.8876  p-VALUE= 0.9717524
## 0 1 3 1 1 0 12 AIC= 345.5336  SSE= 428.2203  p-VALUE= 0.9954375
## 1 1 0 0 1 0 12 AIC= 380.5143  SSE= 834.5908  p-VALUE= 0.04488388
## 1 1 0 0 1 1 12 AIC= 353.7114  SSE= 471.2212  p-VALUE= 0.03873284
## 1 1 0 0 1 2 12 AIC= 354.3441  SSE= 486.725  p-VALUE= 0.04545119
## 1 1 0 0 1 3 12 AIC= 356.0521  SSE= 470.9225  p-VALUE= 0.03880666
## 1 1 0 1 1 0 12 AIC= 356.7573  SSE= 538.1585  p-VALUE= 0.05500837
## 1 1 0 1 1 1 12 AIC= 354.0186  SSE= 482.6763  p-VALUE= 0.04393713
## 1 1 0 1 1 2 12 AIC= 351.9688  SSE= 399.6485  p-VALUE= 0.07522442
## 1 1 0 2 1 0 12 AIC= 356.0277  SSE= 509.7137  p-VALUE= 0.04680801
## 1 1 0 2 1 1 12 AIC= 355.6542  SSE= 432.7777  p-VALUE= 0.03978545
## 1 1 0 3 1 0 12 AIC= 352.7378  SSE= 436.4416  p-VALUE= 0.08017155
## 1 1 1 0 1 0 12 AIC= 366.605  SSE= 649.7251  p-VALUE= 0.9267308
## 1 1 1 0 1 1 12 AIC= 342.1253  SSE= 380.534  p-VALUE= 0.8858535
## 1 1 1 0 1 2 12 AIC= 343.1822  SSE= 397.9261  p-VALUE= 0.9354759
## 1 1 1 1 1 0 12 AIC= 344.0643  SSE= 431.6433  p-VALUE= 0.9562596
## 1 1 1 1 1 1 12 AIC= 342.5986  SSE= 395.2815  p-VALUE= 0.9403914
## 1 1 1 2 1 0 12 AIC= 344.5086  SSE= 418.1467  p-VALUE= 0.9647027
## 1 1 2 0 1 0 12 AIC= 368.4408  SSE= 649.393  p-VALUE= 0.9720196
## 1 1 2 0 1 1 12 AIC= 343.6944  SSE= 374.8967  p-VALUE= 0.9719035
## 1 1 2 1 1 0 12 AIC= 345.5334  SSE= 428.201  p-VALUE= 0.9954742
## 1 1 3 0 1 0 12 AIC= 370.1355  SSE= 647.4613  p-VALUE= 0.9996218
## 2 1 0 0 1 0 12 AIC= 377.8896  SSE= 780.5549  p-VALUE= 0.1511292
## 2 1 0 0 1 1 12 AIC= 349.3836  SSE= 418.1762  p-VALUE= 0.2160548
## 2 1 0 0 1 2 12 AIC= 349.7334  SSE= 439.0554  p-VALUE= 0.2961194
## 2 1 0 1 1 0 12 AIC= 351.9825  SSE= 485.7472  p-VALUE= 0.3173922
## 2 1 0 1 1 1 12 AIC= 349.2  SSE= 434.1367  p-VALUE= 0.298677
## 2 1 0 2 1 0 12 AIC= 351.1152  SSE= 457.6742  p-VALUE= 0.3376532
## 2 1 1 0 1 0 12 AIC= 368.4128  SSE= 649.4652  p-VALUE= 0.9794639
## 2 1 1 0 1 1 12 AIC= 343.6014  SSE= 372.9964  p-VALUE= 0.987403
## 2 1 1 1 1 0 12 AIC= 345.5529  SSE= 428.827  p-VALUE= 0.9930274
## 2 1 2 0 1 0 12 AIC= 368.9252  SSE= 622.5581  p-VALUE= 0.8846976
## 3 1 0 0 1 0 12 AIC= 374.2341  SSE= 718.3746  p-VALUE= 0.5015338
## 3 1 0 0 1 1 12 AIC= 346.8763  SSE= 377.7765  p-VALUE= 0.5129917
## 3 1 0 1 1 0 12 AIC= 350.9779  SSE= 468.7125  p-VALUE= 0.3979103
## 3 1 1 0 1 0 12 AIC= 370.3832  SSE= 649.5847  p-VALUE= 0.986412
```

2. Using auto.arima()

```r
y <- msts(data, seasonal.periods=c(12))
auto.arima( y, d = 1, D = 1,  max.p = 4,  max.q = 4,  max.P = 4,  max.Q = 4, max.order = 10,  start.p =
```

```
## Series: y
## ARIMA(0,1,2)(0,1,1)[12]
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.5497  -0.2369  -0.8304
## s.e.   0.1185   0.1215   0.2576
##
## sigma^2 estimated as 5.512:  log likelihood=-166.85
## AIC=341.69   AICc=342.3   BIC=350.75
```

Best-model

The models with the minimum values of Akaike Information Criterion (AIC) seem to be very similar from the two methods and correponds to an order p,d,q,P,D,Q of 1,1,1,0,1,1 with a seaonal period of 12 (AIC~342) which also has a large enough Ljung-Box test p-value (~0.88).

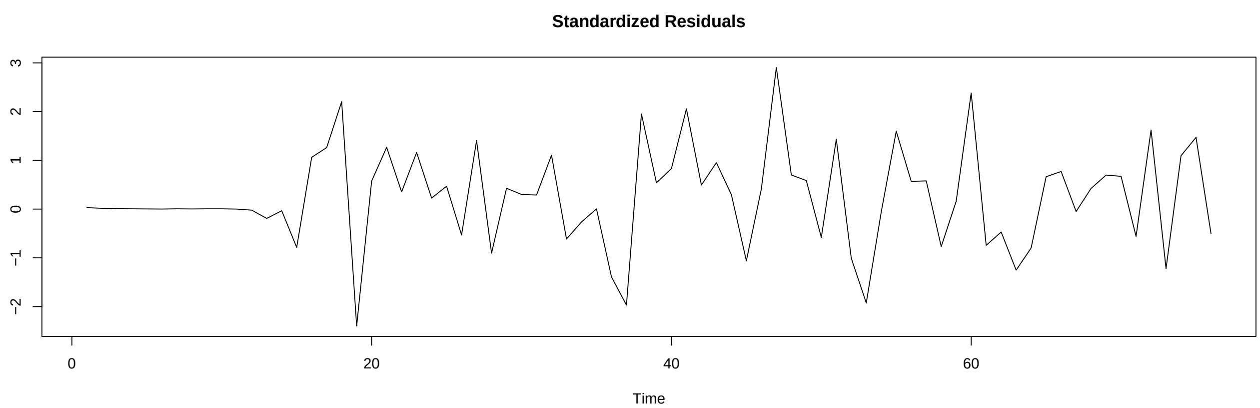
Fitting the best-model on the data

Train-test split

```r
N = length(data)
n = round(0.9*N)
train = data[1:n]
test  = data[(n+1):N]
```

Training the model with the train set

```r
model<-arima(x=train, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
standard_residuals<- model$residuals/sd(model$residuals)
plot(standard_residuals,ylab='',main='Standardized Residuals')
```



We see that the residuals look almost stationary which we can also confirm with the ADF test

```r
print(adf.test(standard_residuals))
```

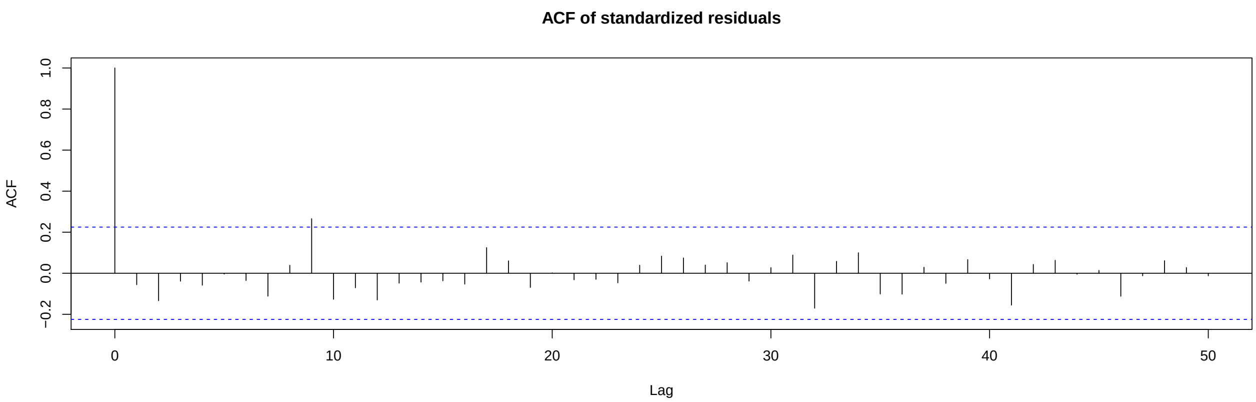
```
## Warning in adf.test(standard_residuals): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  standard_residuals
## Dickey-Fuller = -4.3255, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

The residuals seem to be almost stationary.

Let's check for correlations in the residual using the ACF plot

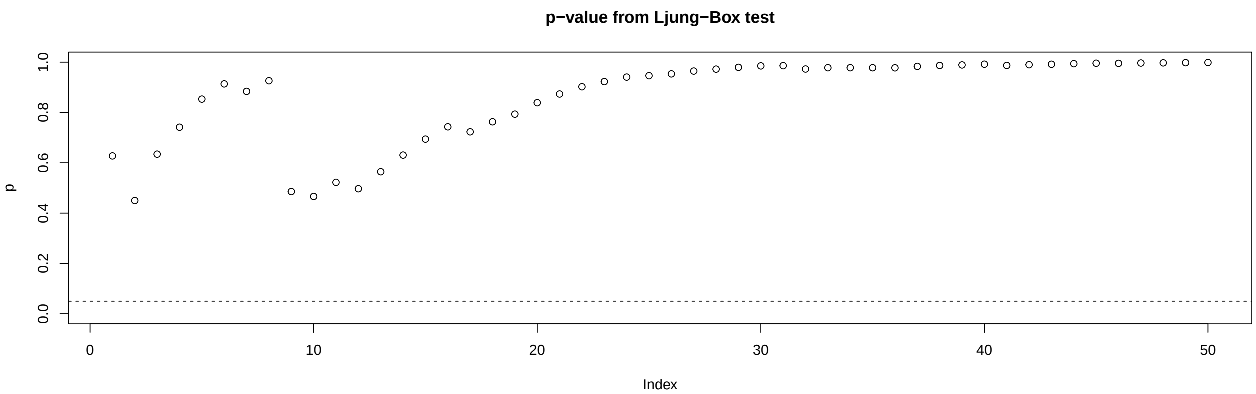
```r
acf(standard_residuals,50,main='ACF of standardized residuals');
```



There is almost no significant correlation in the residuals.

Now, we will perform a Ljung-Box test on the residuals. The null hypotheis for the test is:
H0: The dataset points are independently distributed (not correlated).
where a p-value of greater than 0.05 will be insifficient to reject the null hypothesis.

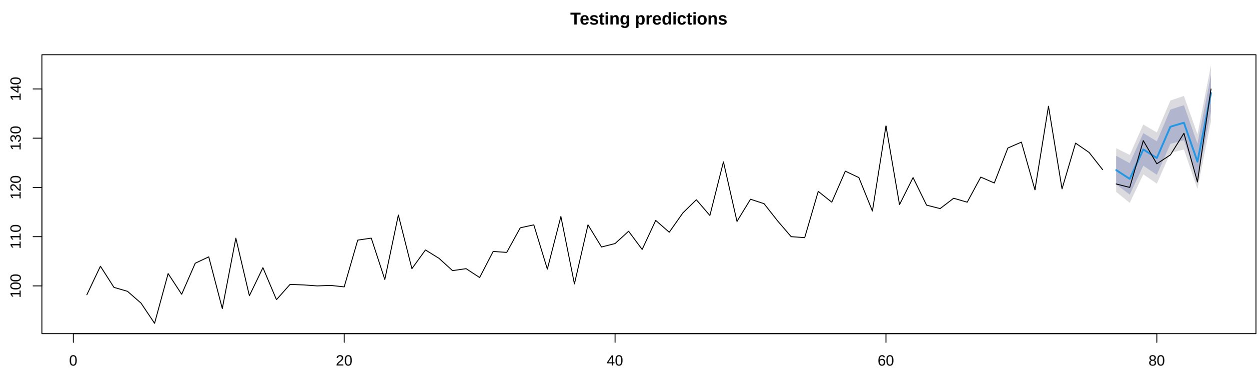
```r
for (lag in seq(1:50)){
  pval<-Box.test(model$residuals, lag=lag)
  p[lag]=pval$p.value
}
plot(p,ylim = (0.0:1), main='p-value from Ljung-Box test')
abline(h=0.05,lty=2)
```



Any value above the dashed line (at y=0.05) is significant. We see that the p-values of the Ljung-Box test at all the lags are significant and therefore the hypothesis that the residuals are not correlated cannot be rejected.

Testing the predictions on the test set

```
model<-arima(x=train, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
pred_len=length(test)
plot(forecast(model, h=pred_len),main='Testing predictions')
train_x = seq(length(train)+1,length(train)+length(test))
lines(train_x,test)
```



Here the black lines in the first part (left) shows the training data and those in the second part shows the test data which alos has blue lines overlaid on it showing the predictions from our model which seem to match the test data pretty well. The small shaded region on the blue lines shows the confidence interval (difficult to resolve here but it actually consists of two different dark and light shaded regions showing the 80% and 95% confidence regions).

Evaluating predictions

```
df2 <- forecast(model,h=pred_len)
df2 <- data.frame(df2)
print(paste0('Root Mean Squared Error in predictions =', round(sqrt(mean((test-df2[,1])**2))/mean(test)
```

```
## [1] "Root Mean Squared Error in predictions =2.34%"
```

Forecasting using the best-model

```
model<-arima(x=data, order = c(1,1,1), seasonal = list(order=c(0,1,1), period=per))
par(mfrow=c(1,1))
h=12 # forecasting for the 12 months after the end of the dataset
plot(forecast(model,h), main='Forecasts for next 12 months');
```