ARIMA Modeling of the Consumer Price Index

Vincent La

March 19, 2017

Abstract

Research Questions

- How has the 2008 recession affected the Consumer Price Index? Would the CPI be significantly different had the recession not occurred?
- What are some significant predictors of the Consumer Price Index?
- Specifially, are there external variables which can help us predict the CPI?
- Does a monthly model provide more accurate predictions than a quarterly model?

Main Findings

Monthly vs. Quarterly Data

- Fitting a monthly data was very difficult due to the amount of possible noise that was in the data
- A model based on quarterly data was better both in terms of statistics such as Akaike Information Criterion and in terms of predicting actual data.

Heteroskedasticity and Log Transforms

- Heteroskedasticity is the main issue for fitting an ARIMA model to the CPI
- Some periods have much higher volatility than others
- A log transform stabilized the error variance but gave prediction intervals that were too wide to be useful

In conclusion, using standard seasonal ARIMA models, I was able to identify potential models based on quarterly data which–while not passing formal statistical tests–were able to accurately predict up the path of CPI up to three years ahead. While log transformed models fit the data well, they are not useful for producing actionable information.

RStudio and R Markdown were used in the production of this report.

```
# Load CPI data for all regions/all items
cpi.data <- read.table("CPI - All Urban Consumers/cu.data.1.AllItems.txt", header=TRUE, fill=TRUE)
cpi.all_regions_raw <- subset(cpi.data, cpi.data$series_id == 'CUUR0000AA0')

# Remove annual average (period M13) from data
cpi.all_regions <- subset(cpi.all_regions_raw, cpi.all_regions_raw$period != 'M13')
```

ARIMA Modeling of the Consumer Price Index

What is the Consumer Price Index?

The Consumer Price Index is a monthly time series, dating back to 1946, collected by the Bureau of Labor Statistics. The CPI is created by calculating the price of a typical bundle of goods purchased by American urban consumers. Uses of the CPI include calculating inflation and cost of living.

Motivation and objectives

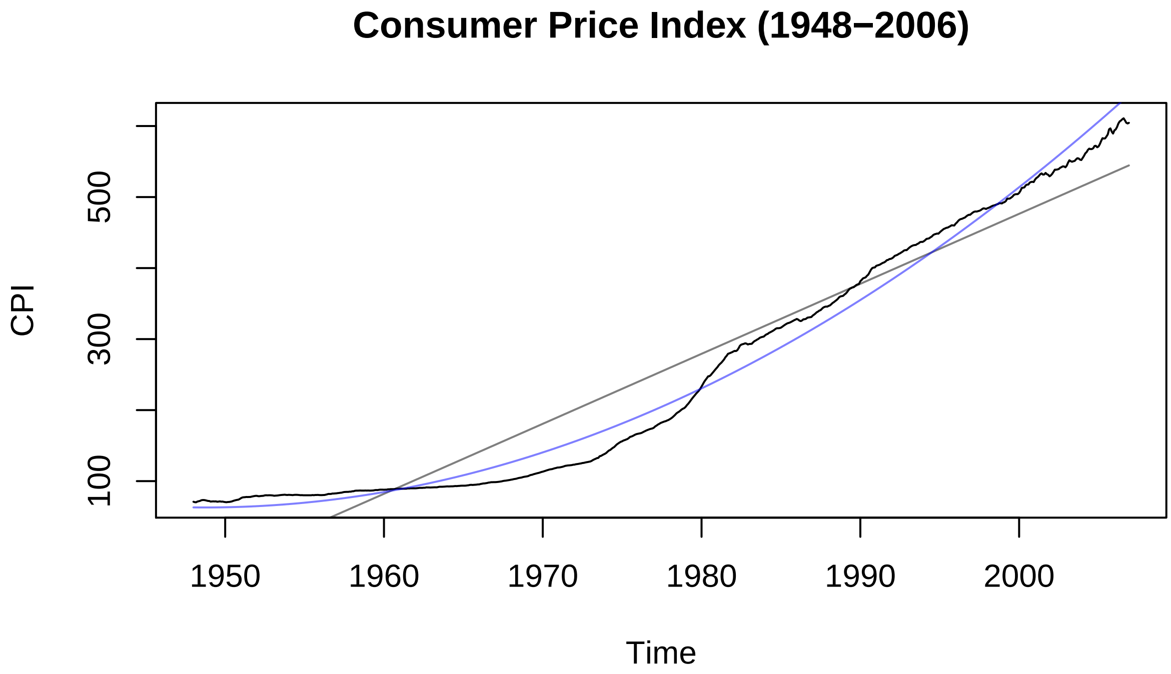
I wanted to analyze the CPI to get an idea of how the cost of living has changed for Americans over time. Specifically, I wanted to see how it has been affected by economic shocks such as the 2008 recession

Time Range

While it would be great if we could create a model for the entirety of the Consumer Price Index, I will restrict the time range of my model for several reasons:

- All of my external variables of interest do not have data that stretches back to 1913
- The period before 1948 (the start of my analysis) contains several economic events which would complicate model building, such as the Great Depression and two World Wars

```r
# Load entire data set
cpi.ts <- ts(data=cpi.all_regions$value, start=c(1913, 1), end=c(2016, 12), freq=12)

# Take subset of CPI
cpi.48_06 <- window(cpi.ts, start=c(1948, 1), end=c(2006, 12), freq=12)
cpi.tslm <- tslm(cpi.48_06 ~ trend)                # Linear Trend?
cpi.tslm2 <- tslm(cpi.48_06 ~ trend + I(trend^2))  # Quadratic Trend?

# Graph CPI
plot(cpi.48_06, main="Consumer Price Index (1948-2006)", ylab="CPI")
lines(cpi.tslm$fitted, col=rgb(0, 0, 0, 0.5))
lines(cpi.tslm2$fitted, col=rgb(0, 0, 1, 0.5))
```



The Consumer Price Index definitely shows positive trend, possibly quadratic or exponential in nature.

Strategy

I will try to find the best monthly model first, and then the best quarterly model. Once I decide whether to use monthly or quarterly data, I will try other methods, e.g. a log transformation, to create better models.

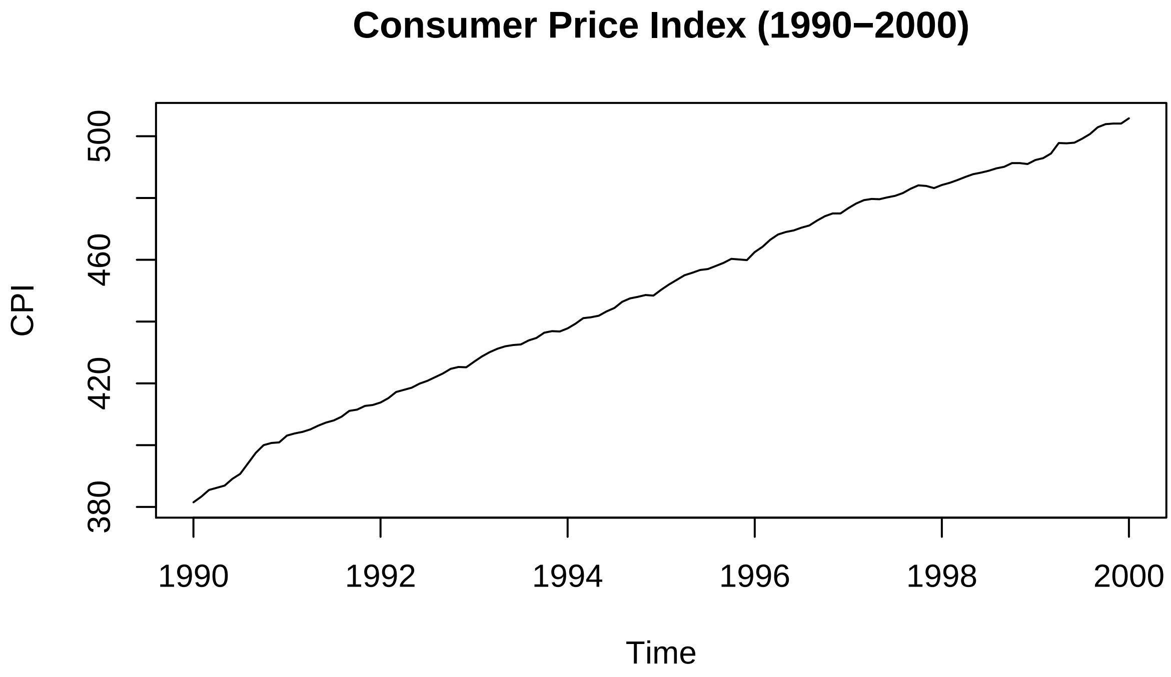
Finding the Best Monthly Model

Seasonality

"Seasonality" is the phenomena of patterns in our data recurring at regular intervals. For example, rainfall levels would be obviously seasonal because precipitation is greater in some parts of the year over others.

The Box-Jenkins method requires that we de-seasonalize the data, so we find patterns in our data without letting seasonal effects confounding our analysis. Since the x-axis covers almost 60 years, we'll zoom in on a decade to see if there is seasonality. Once we do that, there does not appear to be any seasonality.

```
plot(window(cpi.48_06, start=c(1990,1), end=c(2000,1)), main="Consumer Price Index (1990-2000)", ylab="
```



Effect of Differencing

De-Trend

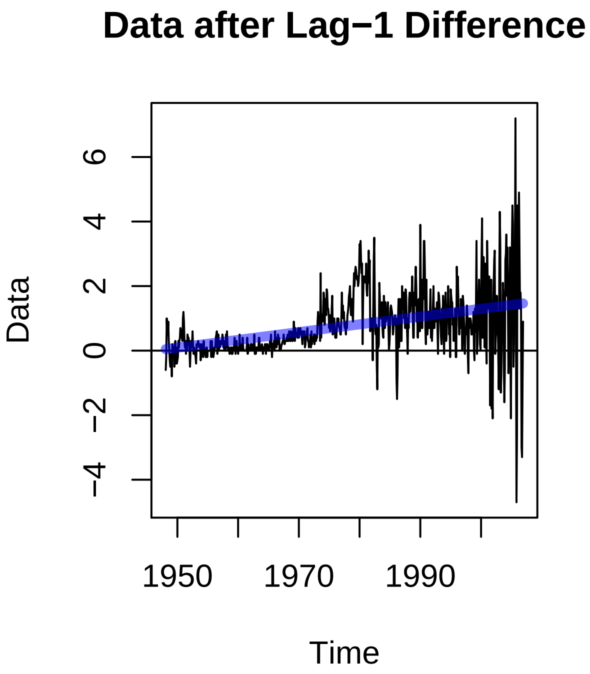
Again, the Box-Jenkins method requires that we also make our data stationary, i.e. not increase over time. This is so we can separate long-term effects from short-term ones.

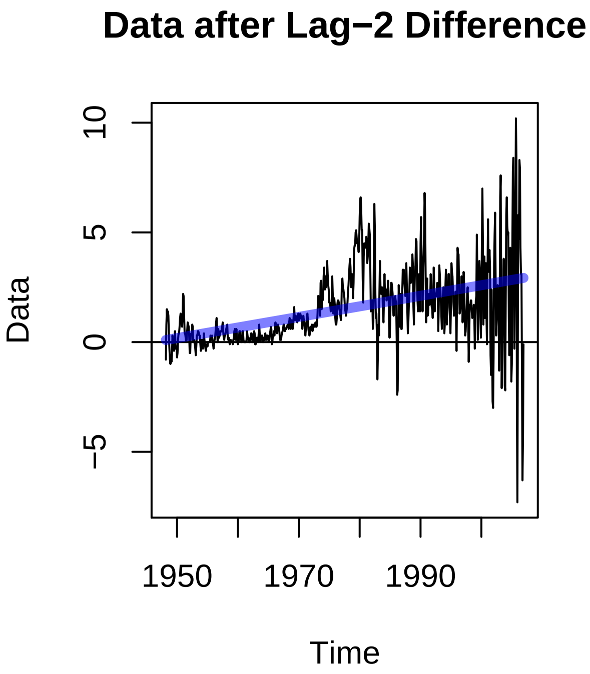
Here, because the CPI roughly appears to follows a quadratic or exponential trend, we'll difference at lag 2, but also at lag 1 to make sure. Surprisingly, a lag-1 difference actually does the most work in reducing the dispersion of our data. This supports the conclusion that it always pays to double check!

```
par(mfrow=c(1,2))
cpi.diff1 <- arma.diff(cpi.48_06, lag=1)
```

```
## Variance of the original data: 30648.85
## Variance of the lag-1 differenced data: 1.065655
```

```
cpi.diff2 <- arma.diff(cpi.48_06, lag=2)
```



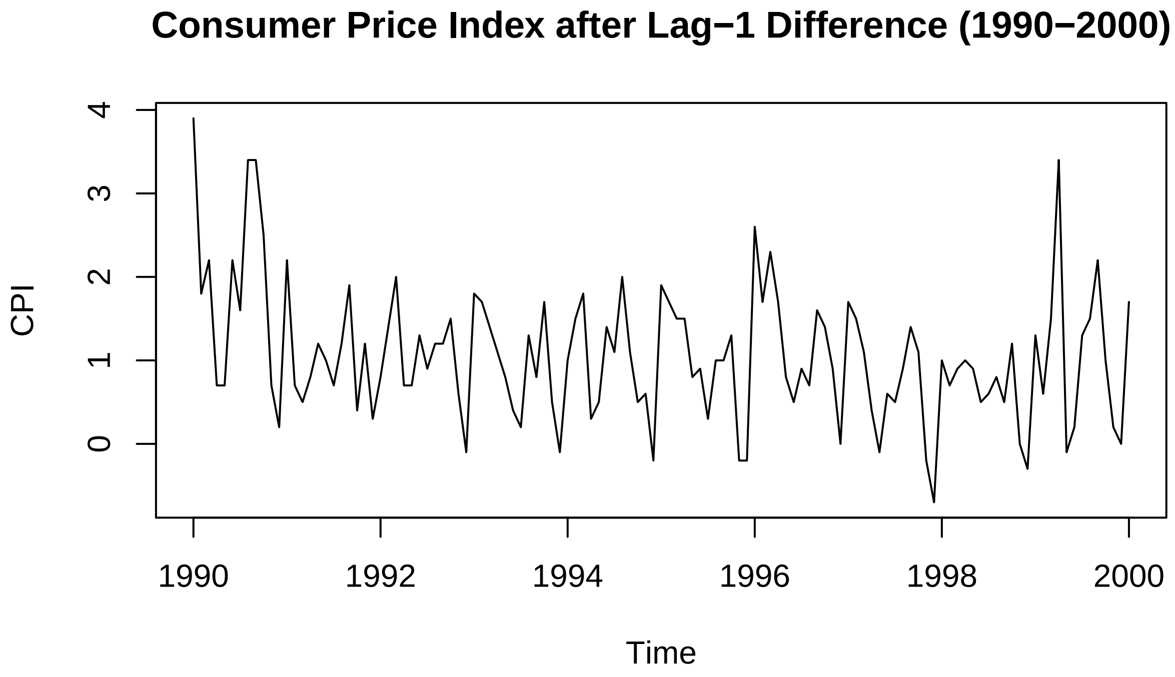


```
## Variance of the original data: 30648.85
## Variance of the lag-2 differenced data: 3.278931
```

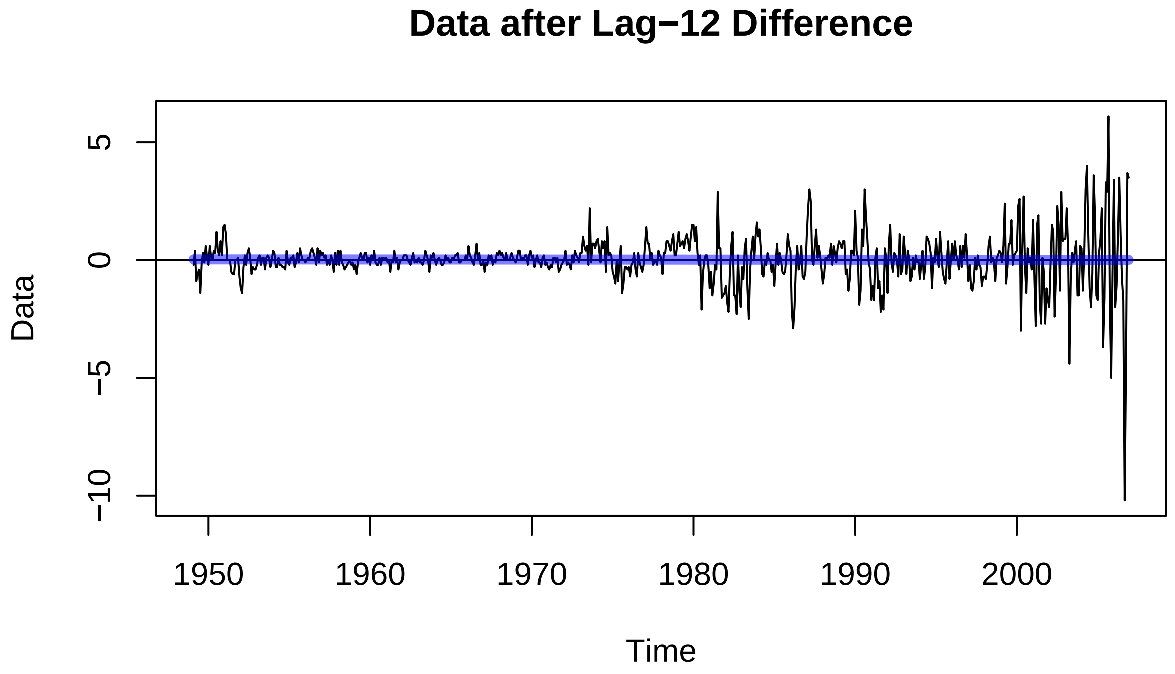
De-Trend and De-Seasonalize

After taking a lag-1 difference in an attempt to de-trend the series, we notice that it becomes seasonal. Furthermore, there is still a remaining trend. Therefore, we should difference at lag 12.

```
plot(window(cpi.diff1, start=c(1990,1), end=c(2000,1)), main="Consumer Price Index after Lag-1 Differen
```



```
cpi.diff1_12 <- arma.diff(cpi.diff1, lag=12)
```



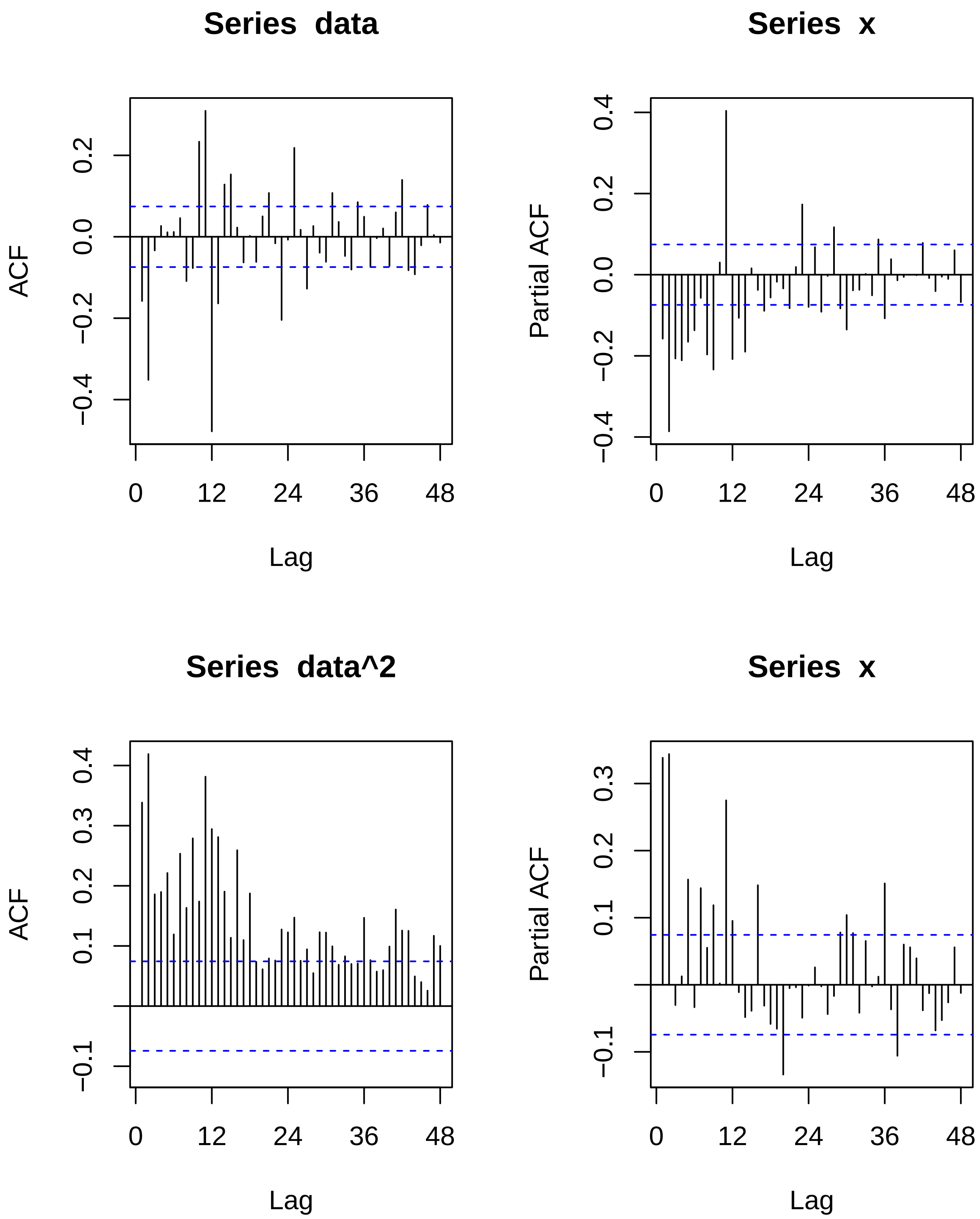
```
## Variance of the original data: 1.065655
## Variance of the lag-12 differenced data: 1.12479
```

While differencing at lag 12 slightly increases the dispersion of our data, it also removes the remaining trend. For the models below, we will apply a first-order difference at lag 1 and a second-order difference at lag 12.

Identification

After satisfactorily de-trending and de-seasonalizing the data, we can now begin to identify models. ACF/PACF identification for this time series is made more difficult by the effect of severe economic events, such as recessions.

```r
arma.id(cpi.48_06, d=1, d2=12)
```



The ACF shows two major spikes at the first two lags and then abruptly cuts off, which suggests at least an MA(2) model. Further supporting this is the fact that the PACF tails off in the non-seasonal lags.

Because the ACF doesn't appear to tail off, and the PACF doesn't seem to cutt off at any point, it is hard to identified autoregressive parameters. However, autoregressive models are widely used in econometrics so I would not be surprised if they were neccessary.

Seasonal Effects

There's large ACF spikes at lags 12, 24, 36 or the lags around them. These spikes decay over time, suggesting suggesting seasonal autoregressive components. The same can be said about the PACF, suggesting seasonal moving average components. Because of the PACF spikes at lags 13, 14, and 15 which tail off thereafter, we will try a model with SMA(3) components. Moreover, because of the ACF tailing off at roughly the same lags, we'll also consider SAR(3) components.

ARCH/GARCH Effects

Furthermore, a plot of the squared observations indicates that they are also serially correlated, implying an ARCH/GARCH model may be useful.

Fitting

Because it seems an ARIMA model with 12 MA and 12 AR coefficients seems like overkill, I will also use the auto.arima() function to help identify a model.

```r
model.arima.1 <- arima(x=cpi.48_06, order=c(1, 1, 2), seasonal=list(order=c(3, 1, 3), period=12))
summary(model.arima.1)
```

```
##
## Call:
## arima(x = cpi.48_06, order = c(1, 1, 2), seasonal = list(order = c(3, 1, 3),
##     period = 12))
##
## Coefficients:
##          ar1      ma1      ma2     sar1    sar2     sar3     sma1     sma2
##       0.9684  -0.4377  -0.4157  -0.5230  0.1775  -0.0492  -0.4693  -0.5903
## s.e.  0.0143   0.0390   0.0375   0.5065  0.4269   0.1031   0.5058   0.5917
##         sma3
##       0.3469
## s.e.  0.3309
##
## sigma^2 estimated as 0.5244:  log likelihood = -769.74,  aic = 1559.47
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE      MAPE    MASE
## Training set 0.02139083 0.7175248 0.4409639 0.01648396 0.1876771 0.49541
##                     ACF1
## Training set -0.02964684
```

```r
# Automatic Model
# model.auto.1 <- auto.arima(x=cpi.48_06, d=1, D=1, max.P=3, max.Q=3, max.order=10)
```

In the first model, none of the SAR or SMA coefficients were significant, but this may change once we drop some of them. However, all of the non-seasonal components were significant.

The auto.arima() function takes a very long time to execute, which is not surprising given the ACF/PACF plots.

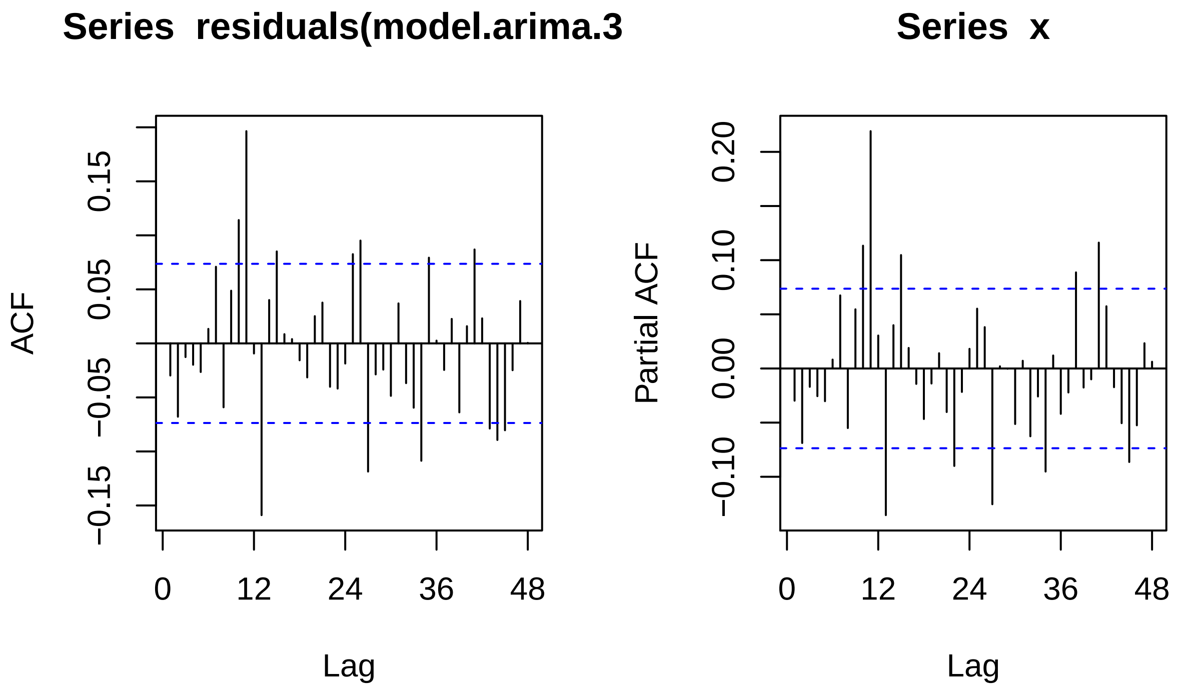
```r
# Omitted for brevity
# model.arima.2 <- arima(x=cpi.48_06, order=c(1, 1, 2), seasonal=list(order=c(2, 1, 2), period=12))
# summary(model.arima.2)
```

```r
model.arima.3 <- arima(x=cpi.48_06, order=c(1, 1, 2), seasonal=list(order=c(1, 1, 2), period=12))
summary(model.arima.3)
```

```
##
## Call:
## arima(x = cpi.48_06, order = c(1, 1, 2), seasonal = list(order = c(1, 1, 2),
##     period = 12))
##
## Coefficients:
##          ar1      ma1      ma2    sar1     sma1    sma2
##       0.9681  -0.4378  -0.4157  0.4091  -1.4047  0.5299
## s.e.  0.0144   0.0391   0.0376  0.1626   0.1472  0.1227
##
## sigma^2 estimated as 0.5246:  log likelihood = -769.87,  aic = 1553.73
##
## Training set error measures:
##                      ME      RMSE      MAE        MPE      MAPE      MASE
## Training set 0.02116564 0.7176698 0.440692 0.01643274 0.1879297 0.4951045
##                     ACF1
## Training set -0.02973879
```

After dropping SMA and SAR coefficients, I found that the ARIMA(1,1,2)x(1,1,2) had all coefficients significant.

```r
par(mfrow=c(1,2))
Acf(residuals(model.arima.3), lag.max=48)
Pacf(residuals(model.arima.3), lag.max=48)
```



Then, I attempted to use the residuals to modify the model. Again, the residuals suggested seasonal effects. But as we have seen above, adding more seasonal components leads to insigificant coefficients.

```r
# Best Model
model.arima.4 <- arima(x=cpi.48_06, order=c(1, 1, 3), seasonal=list(order=c(1, 1, 2), period=12))
summary(model.arima.4)
```

```
##
## Call:
## arima(x = cpi.48_06, order = c(1, 1, 3), seasonal = list(order = c(1, 1, 2),
##     period = 12))
##
## Coefficients:
##          ar1      ma1      ma2     ma3    sar1     sma1    sma2
##       0.9648  -0.4777  -0.4487  0.0875  0.4000  -1.3961  0.5297
## s.e.  0.0149   0.0421   0.0391  0.0421  0.1568   0.1417  0.1177
##
## sigma^2 estimated as 0.5216:  log likelihood = -767.71,  aic = 1551.42
##
## Training set error measures:
##                      ME      RMSE     MAE         MPE      MAPE      MASE
## Training set 0.01965701 0.7156096 0.44108 0.01571544 0.1887354 0.4955403
##                     ACF1
## Training set 0.005104216
```

```r
# Some coefficients insigificant
model.arima.5 <- arima(x=cpi.48_06, order=c(2, 1, 3), seasonal=list(order=c(1, 1, 2), period=12))
summary(model.arima.5)
```

```
##
## Call:
## arima(x = cpi.48_06, order = c(2, 1, 3), seasonal = list(order = c(1, 1, 2),
##     period = 12))
##
## Coefficients:
##          ar1      ar2      ma1      ma2     ma3    sar1     sma1    sma2
##       1.7885  -0.8072  -1.3371  -0.0611  0.4682  0.2831  -1.3136  0.4707
## s.e.  0.0461   0.0468   0.0519   0.0690  0.0378  0.1648   0.1499  0.1255
##
## sigma^2 estimated as 0.5044:  log likelihood = -756.15,  aic = 1530.29
##
## Training set error measures:
##                      ME      RMSE       MAE       MPE      MAPE      MASE
## Training set 0.01966136 0.7036966 0.4331476 0.0159433 0.1869477 0.4866286
##                     ACF1
## Training set 0.008758324
```

We can add an extra moving average term and still have all coefficients be significant. However, adding another autoregressive term causes some coefficients to be insigificant. Therefore, it seems we have reached a stopping point.

Best Fitting Model

The results of the previous section are summarized here.

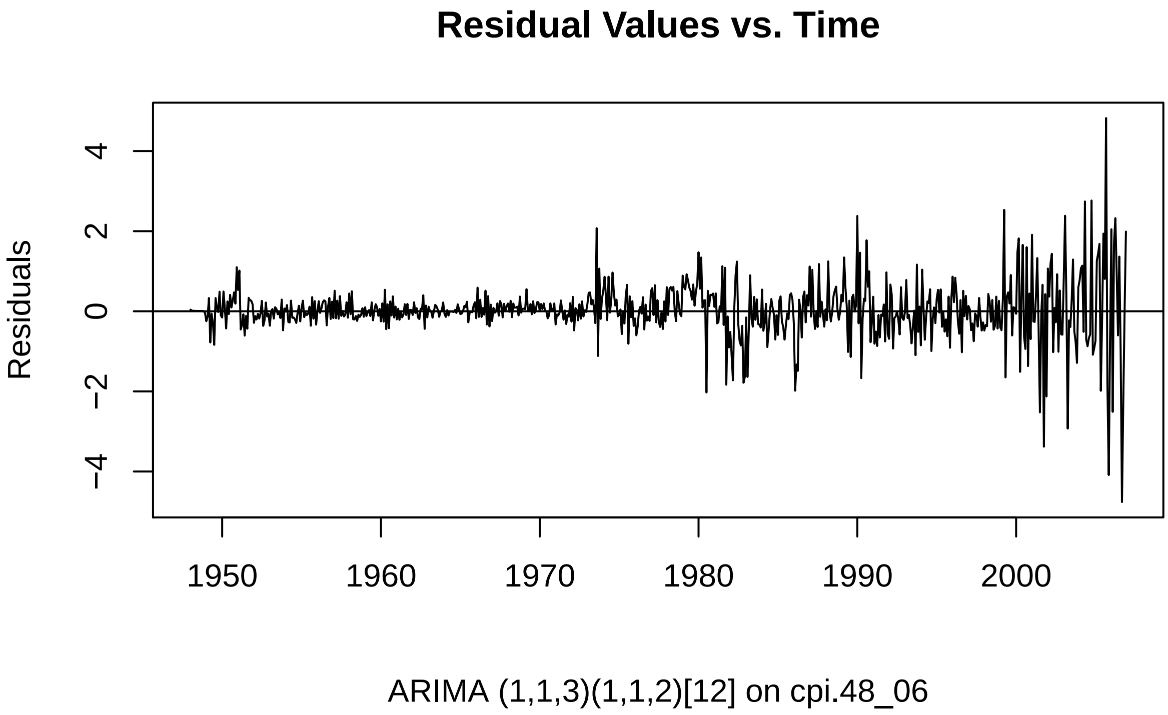
```r
fit.compare(model.arima.1, model.arima.3, model.arima.4)
```

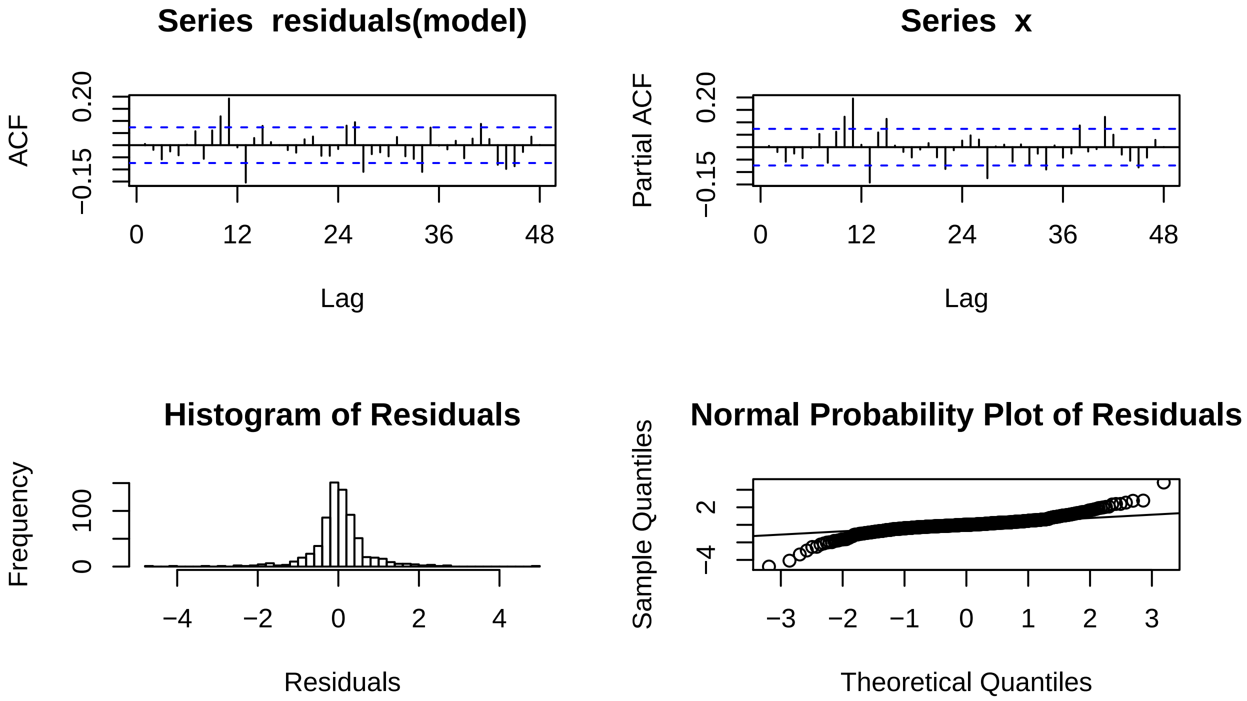
```
##             Model Order Log.Lik     AIC    AICc sigma2 Convergence.
## 1 model.arima.1     9 -769.74 1559.47 1559.73   0.52          Yes
## 2 model.arima.3     6 -769.87 1553.73 1553.85   0.52          Yes
## 3 model.arima.4     7 -767.71 1551.42 1551.58   0.52          Yes
```

model.arima.4 has the lowest AICc and variance, as well as the highest log-likelihood.

Model Diagnostics

```r
arma.diag(model.arima.4)
```





The residuals appear to be somewhat Normal, but with a heavy-tail distribution. Is this expected because of unexpected economic events. Furthermore, as mentioned above, there is still serial correlation at seasonal lags.

The worrisome problem with this model however, is shown by the first plot. The model is clearly heteroskedastic with respect to time, with the error getting worse as we approach the present.

```
arma.test(model.arima.4)
```

```
## [[1]]
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.86106, p-value < 2.2e-16
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  residuals(model)
## A = 25.256, p-value < 2.2e-16
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  residuals(model)
## X-squared = 88.08, df = 19.608, p-value = 1.178e-10
##
##
## [[4]]
##
##  Box-Ljung test
##
## data:  residuals(model)^2
## X-squared = 634.76, df = 26.608, p-value < 2.2e-16
```

Unsurprisingly, this model passed none of the formal statistical tests. Nevertheless, it is still the best monthly model.

```
model.arima <- model.arima.4
```

Finding the Best Quarterly Model

Perhaps one way we could improve our models is to use quarterly data instead of monthly data. By smoothing out monthly disturbances, we could perhaps get a better view of the bigger picture.

Because quarterly CPI data is not available, I will create it by setting the value of each quarter to the average of the months contained.

```
# Full dataset
cpi_qrt.ts <- ts(quarterize.avg(cpi.ts), start=c(1913, 1), end=c(2016, 4), freq=4)

# Training dataset
cpi_qrt.48_05 <- window(cpi_qrt.ts, start=c(1948, 1), end=c(2005, 2), freq=4)
model_qrt.lm <- tslm(cpi_qrt.48_05 ~ trend)
```

Effect of Differencing

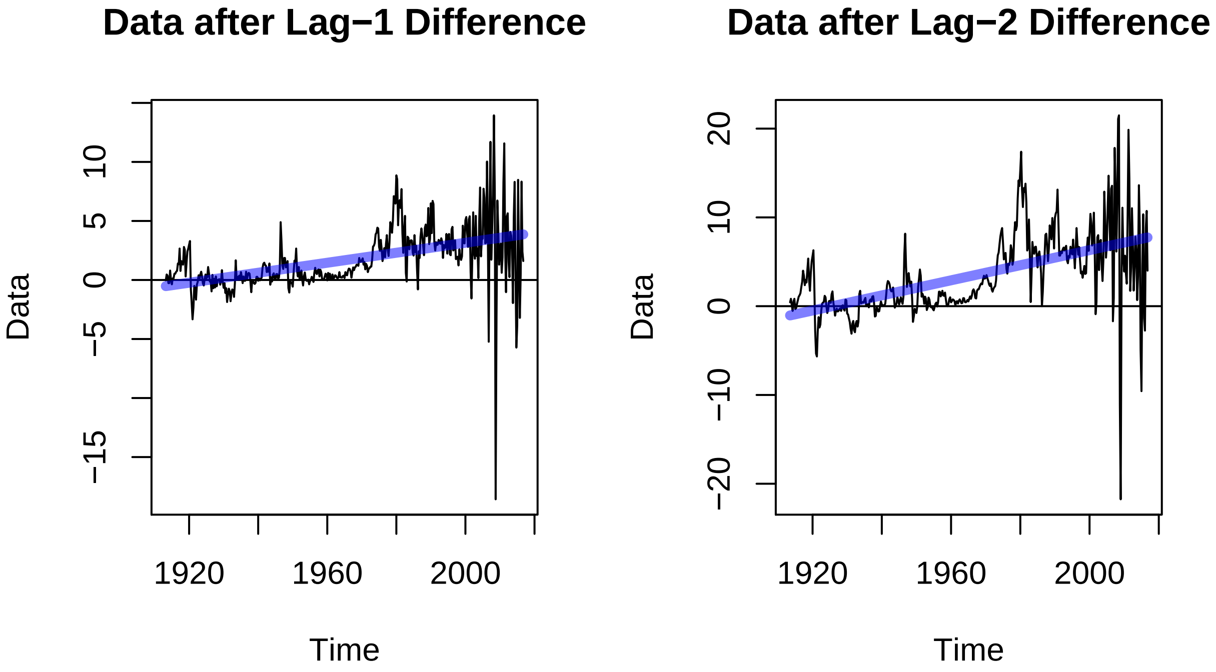
De-Trend

Unsurprising, the same analysis for the montly data holds when it comes to differencing. A lag-1 difference provides the best results.

```
par(mfrow=c(1,2))
cpi_qrt.diff1 <- arma.diff(cpi_qrt.ts, 1)
```

```
## Variance of the original data: 48363.5
## Variance of the lag-1 differenced data: 6.96963
```

```r
cpi_qrt.diff2 <- arma.diff(cpi_qrt.ts, 2)
```

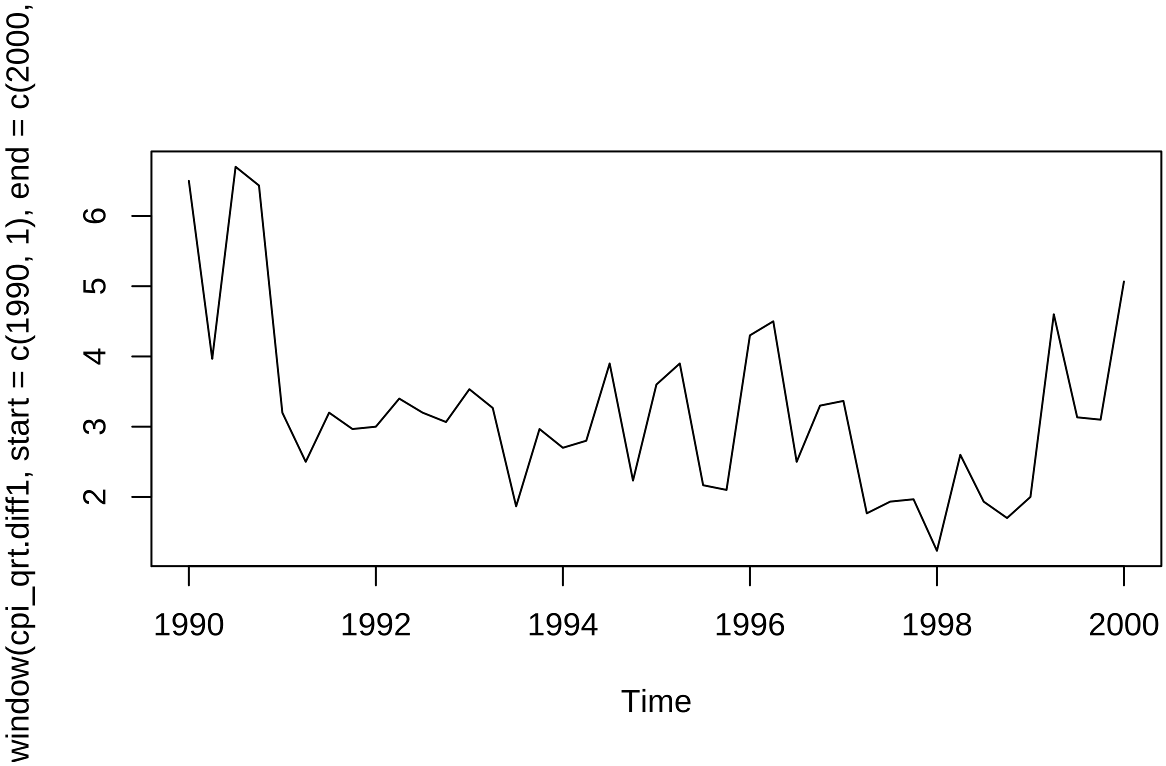

Data after Lag−1 Difference

Data after Lag−2 Difference

```
## Variance of the original data: 48363.5
## Variance of the lag-2 differenced data: 20.93565
```

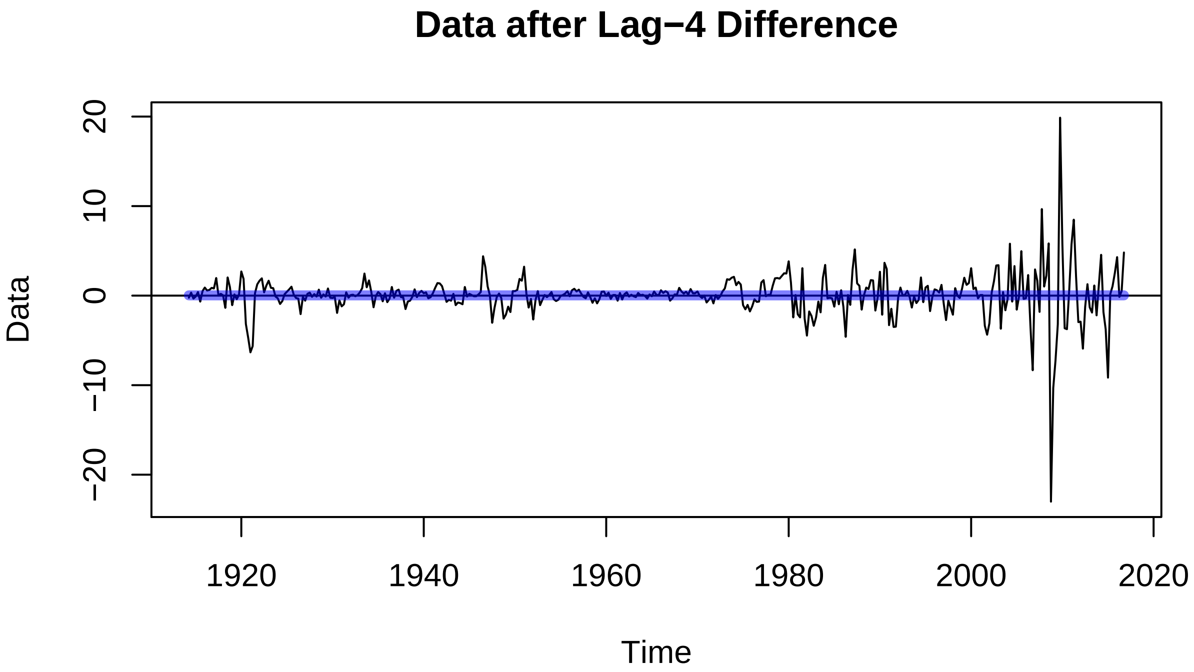
De-Seasonalize

```r
plot(window(cpi_qrt.diff1, start=c(1990, 1), end=c(2000, 1)))
```



After differencing at lag 1, we notice that there might be some seasonality especially after zooming in on 1990-2000–a relatively stable period for the US economy.

```
# Quarterly data --> use a lag 4 difference to de-seasonalize
cpi_qrt.diff1_4 <- arma.diff(cpi_qrt.diff1, 4)
```

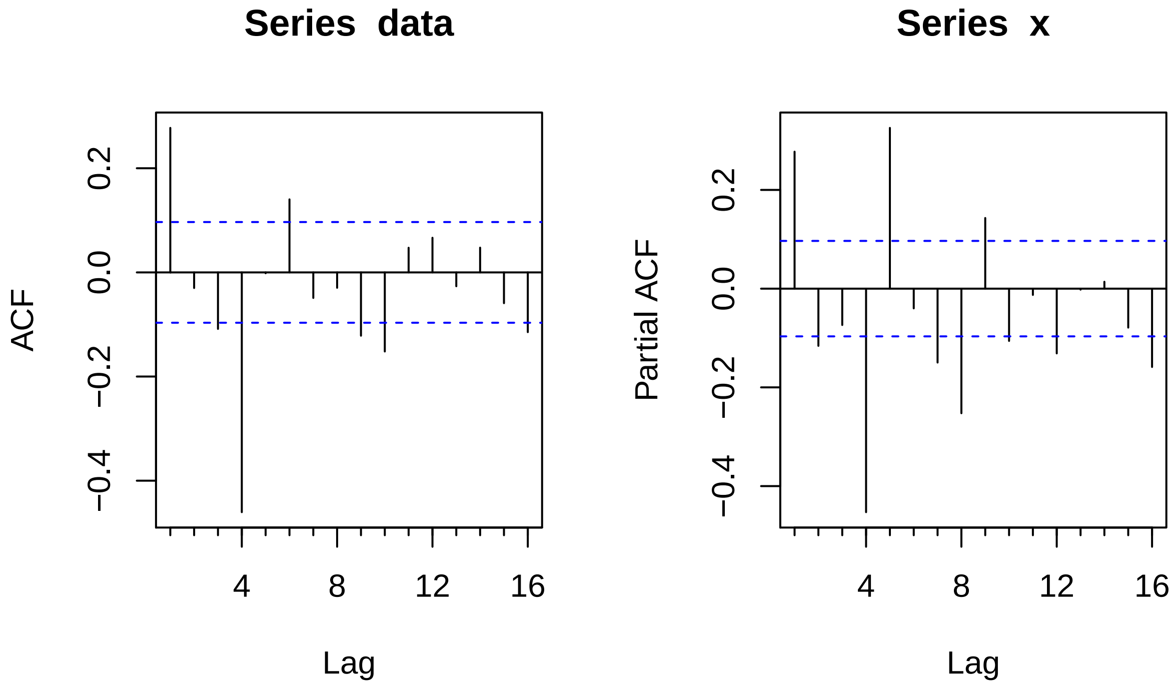


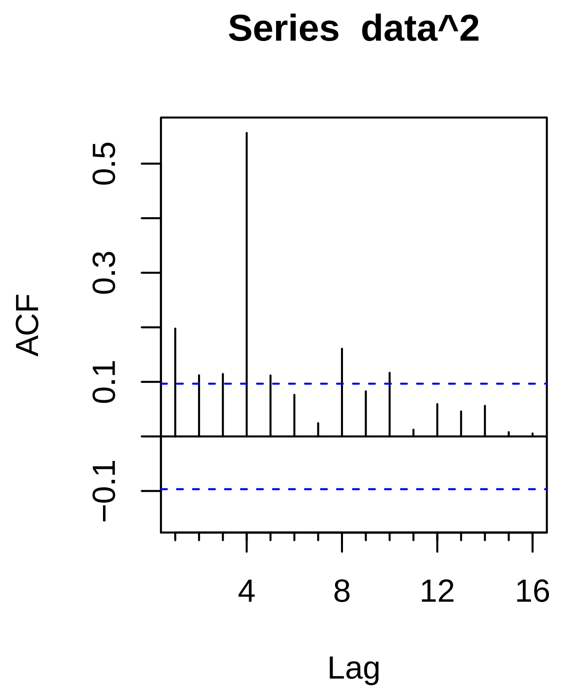
```
## Variance of the original data: 6.96963
## Variance of the lag-4 differenced data: 6.250456
```

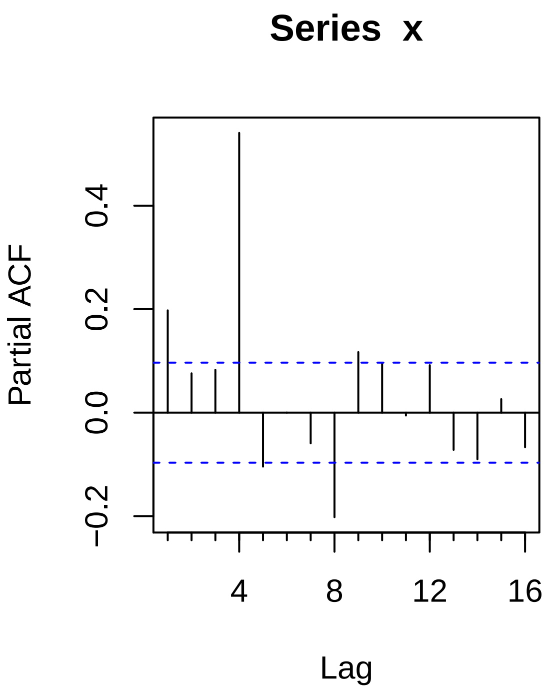
In addition to reducing the variance of our data, the lag-4 difference also removed the remaining trend.

Identification

```
arma.id(cpi_qrt.diff1_4, lag.max=16)
```







There is a large ACF spike in the first quarter, and then tails off after. The same for the PACF. Therefore, I will consider an ARMA(1, 1) model for the non-seasonal lags.

Seasonal Effects

There appear to be significant ACF spikes at lags 9 and 10. The ACF and PACF seem to share the same behavior for seasonal lags, so we will use SMA and SAR components of equal order (2).

ARCH/GARCH Effects

Because of the signifcant ACF/PACF spikes for the squared series, an ARCH/GARCH model might be appropriate.

```r
# First try
model_qrt.arima.1 <- arima(x=cpi_qrt.48_05, order=c(1, 1, 1), seasonal=list(order=c(2, 1, 2), period=4)
summary(model_qrt.arima.1)
```

```
##
## Call:
## arima(x = cpi_qrt.48_05, order = c(1, 1, 1), seasonal = list(order = c(2, 1,
##     2), period = 4))
##
## Coefficients:
##          ar1      ma1    sar1     sar2     sma1    sma2
##       0.8654  -0.4268  0.5684  -0.1588  -1.4081  0.5711
## s.e.  0.0527   0.0904  0.2848   0.0904   0.2805  0.2329
##
## sigma^2 estimated as 1.398:  log likelihood = -358.97,  aic = 731.94
##
```

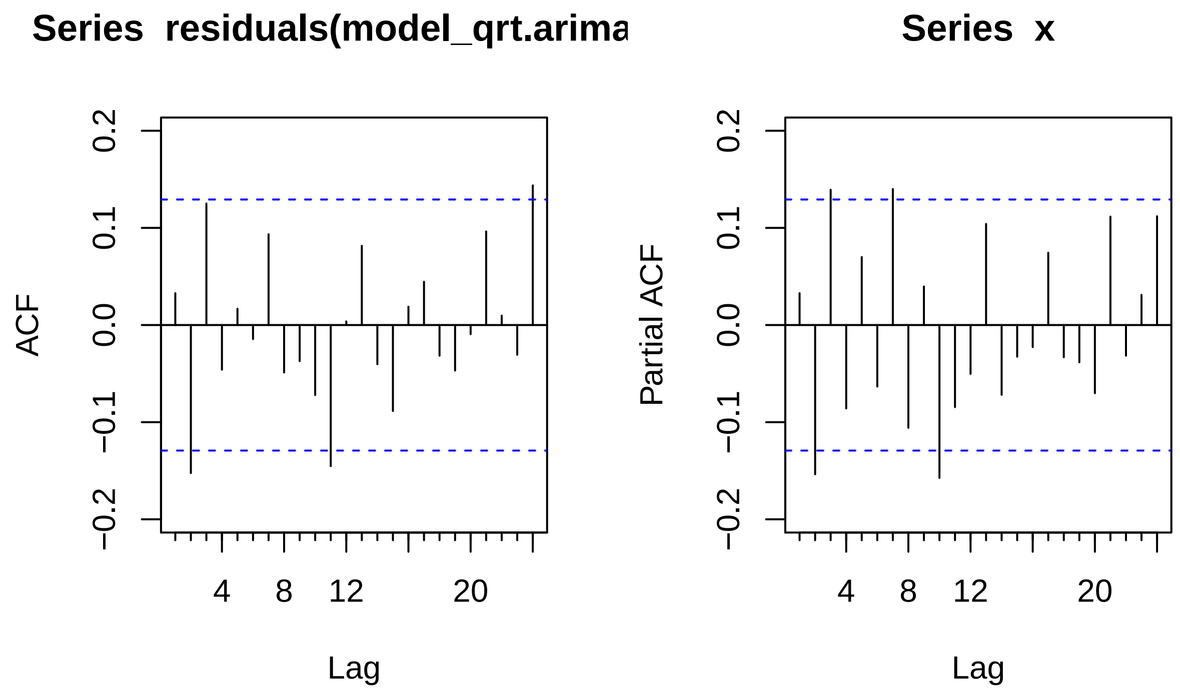
```
## Training set error measures:
##                      ME     RMSE       MAE       MPE      MAPE      MASE
## Training set 0.05632807 1.169316 0.7951193 0.0426859 0.3617569 0.3466254
##                    ACF1
## Training set 0.03384725
```

The second-order SAR coefficient is not significant, so we will drop it.

```
model_qrt.arima.2 <- arima(x=cpi_qrt.48_05, order=c(1, 1, 1), seasonal=list(order=c(1, 1, 2), period=4)
summary(model_qrt.arima.2)
```

```
##
## Call:
## arima(x = cpi_qrt.48_05, order = c(1, 1, 1), seasonal = list(order = c(1, 1,
##     2), period = 4))
##
## Coefficients:
##          ar1      ma1     sar1    sma1     sma2
##       0.8690  -0.4458  -0.9371  0.1691  -0.8209
## s.e.  0.0601   0.0963   0.0278  0.0223   0.0318
##
## sigma^2 estimated as 1.398:  log likelihood = -360.4,  aic = 732.79
##
## Training set error measures:
##                   ME     RMSE       MAE        MPE      MAPE      MASE
## Training set 0.07855 1.169426 0.7951325 0.05142494 0.3670891 0.3466312
##                    ACF1
## Training set 0.03277731
```

```
par(mfrow=c(1,2))
Acf(residuals(model_qrt.arima.2), lag.max=24)
Pacf(residuals(model_qrt.arima.2), lag.max=24)
```



There are some lags sticking outside of the confidence intervals, but they are very small. But just to be sure, I will consider a model with just one extra non-seasonal AR and MA coefficient.

```
# SMA1 insignificant
model_qrt.arima.3 <- arima(x=cpi_qrt.48_05, order=c(2, 1, 2), seasonal=list(order=c(1, 1, 2), period=4)
summary(model_qrt.arima.3)
```

```
##
## Call:
## arima(x = cpi_qrt.48_05, order = c(2, 1, 2), seasonal = list(order = c(1, 1,
##     2), period = 4))
##
## Coefficients:
##          ar1     ar2     ma1      ma2      sar1    sma1     sma2
##       0.2672  0.5381  0.4381  -0.5619  -0.9404  0.1721  -0.8161
## s.e.  0.0880  0.0610  0.0860   0.0848   0.0661  0.1555   0.1268
##
## sigma^2 estimated as 1.158:  log likelihood = -342.56,  aic = 701.11
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE      MAPE     MASE
## Training set 0.07660617 1.064599 0.7366151 0.04976084 0.3453034 0.321121
##                     ACF1
## Training set -0.04251661
```

```
# MA2, SMA1 insigificant
# model_qrt.arima.4 <- arima(x=cpi_qrt.48_05, order=c(1, 1, 2), seasonal=list(order=c(1, 1, 2), period=
# summary(model_qrt.arima.4)

# AR2 insigificant
# model_qrt.arima.5 <- arima(x=cpi_qrt.48_05, order=c(2, 1, 1), seasonal=list(order=c(1, 1, 2), period=
# summary(model_qrt.arima.5)
```

In the interest of brevity, the detailed summaries for some models have been ommitted. However, adding more non-seasonal components caused some of coefficients to become insigificant.

Now, I will use the auto.arima() function to see if I can create a better model than a brute force approach.

```
# Automatically identified model
model_qrt.auto.1 <- auto.arima(x=cpi_qrt.48_05)
summary(model_qrt.auto.1)
```

```
## Series:
## ARIMA(3,2,0)(0,0,2)[4]
##
## Coefficients:
##           ar1      ar2      ar3     sma1     sma2
##       -0.3635  -0.8118  -0.3029  -0.4014  -0.1347
## s.e.   0.0690   0.0794   0.0738   0.1066   0.0786
##
## sigma^2 estimated as 1.375:  log likelihood=-358.07
## AIC=728.14   AICc=728.52   BIC=748.71
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE      MAPE       MASE
## Training set 0.08241787 1.154423 0.8034731 0.05294763 0.3748763 0.08977501
##                    ACF1
## Training set 0.02901261
```

Interestingly enough, the auto.arima() function performed only a lag 2 difference. The SMA2 coefficient is insignificant at the 95% level. However, it has a slightly better log-likelihood than my preferred manually-fitted ARIMA model (although not better than model-qrt.arima.3–which has an insigifcant SMA1 term). Next, I'll run it again just but with my preferred level of differencing specified to see how it performs.

```
model_qrt.auto.2 <- auto.arima(x=cpi_qrt.48_05, d=1, D=1)
summary(model_qrt.auto.2)
```

```
## Series:
## ARIMA(1,1,3)(0,1,1)[4]
##
## Coefficients:
##          ar1      ma1      ma2     ma3     sma1
##       0.8067  -0.1286  -0.5170  0.4717  -0.8536
## s.e.  0.0844   0.1013   0.0749  0.0589   0.0543
##
## sigma^2 estimated as 1.252:  log likelihood=-346.11
## AIC=704.22   AICc=704.61   BIC=724.72
##
## Training set error measures:
##                      ME     RMSE       MAE        MPE      MAPE      MASE
## Training set 0.09277254 1.094267 0.7636444 0.05639576 0.3585664 0.0853248
##                     ACF1
## Training set -0.05319027
```

Not a big improvement on previous models.

Comparison of Fitting

```
fit.compare(model_qrt.arima.2, model_qrt.arima.3, model_qrt.auto.1, model_qrt.auto.2)
```

```
##                Model Order Log.Lik    AIC   AICc sigma2 Convergence.
## 1 model_qrt.arima.2     5 -360.40 732.79 733.06   1.40        Error
## 2 model_qrt.arima.3     7 -342.56 701.11 701.62   1.16          Yes
## 3  model_qrt.auto.1     5 -358.07 728.14 728.40   1.37          Yes
## 4  model_qrt.auto.2     5 -346.11 704.22 704.49   1.25          Yes
```

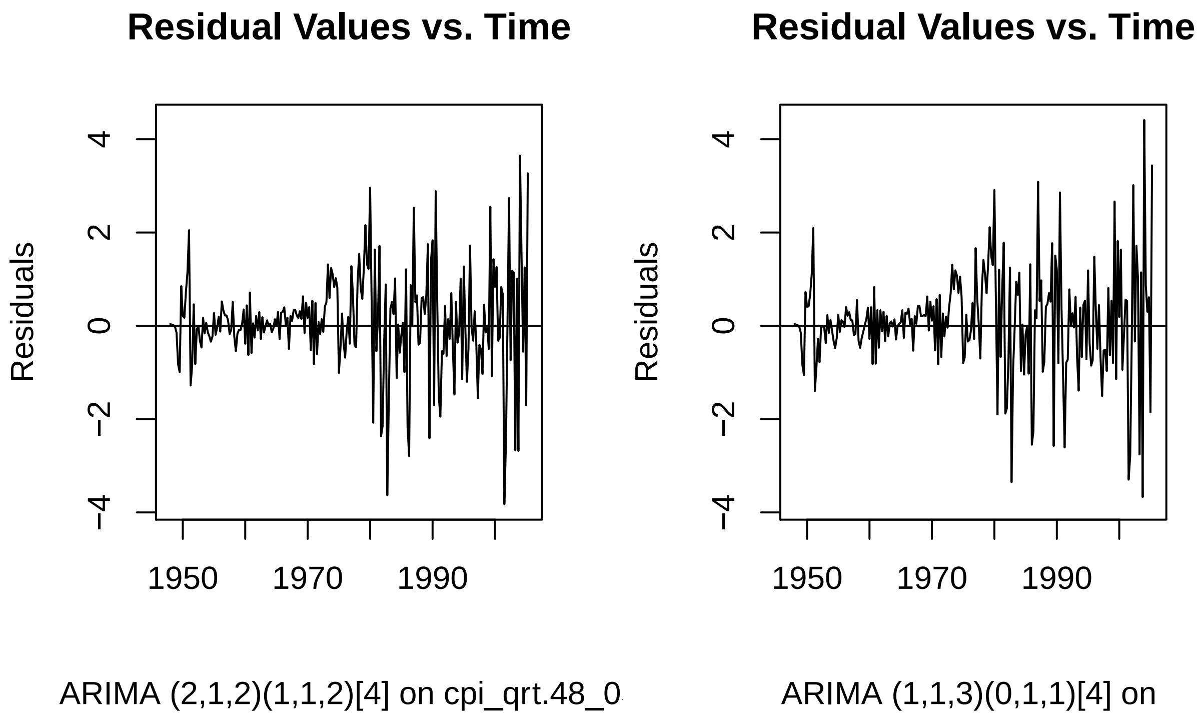
According to AICc, my best manually fitted model should be used, followed by the best automatically identified model.

Comparison of Residuals

There does not seem to be a big difference in terms of fitting when it comes to the models, but perhaps we should look at the residuals now. Although I am very confident in the model I manually fitted, I am still curious as to how auto.arima() compares.

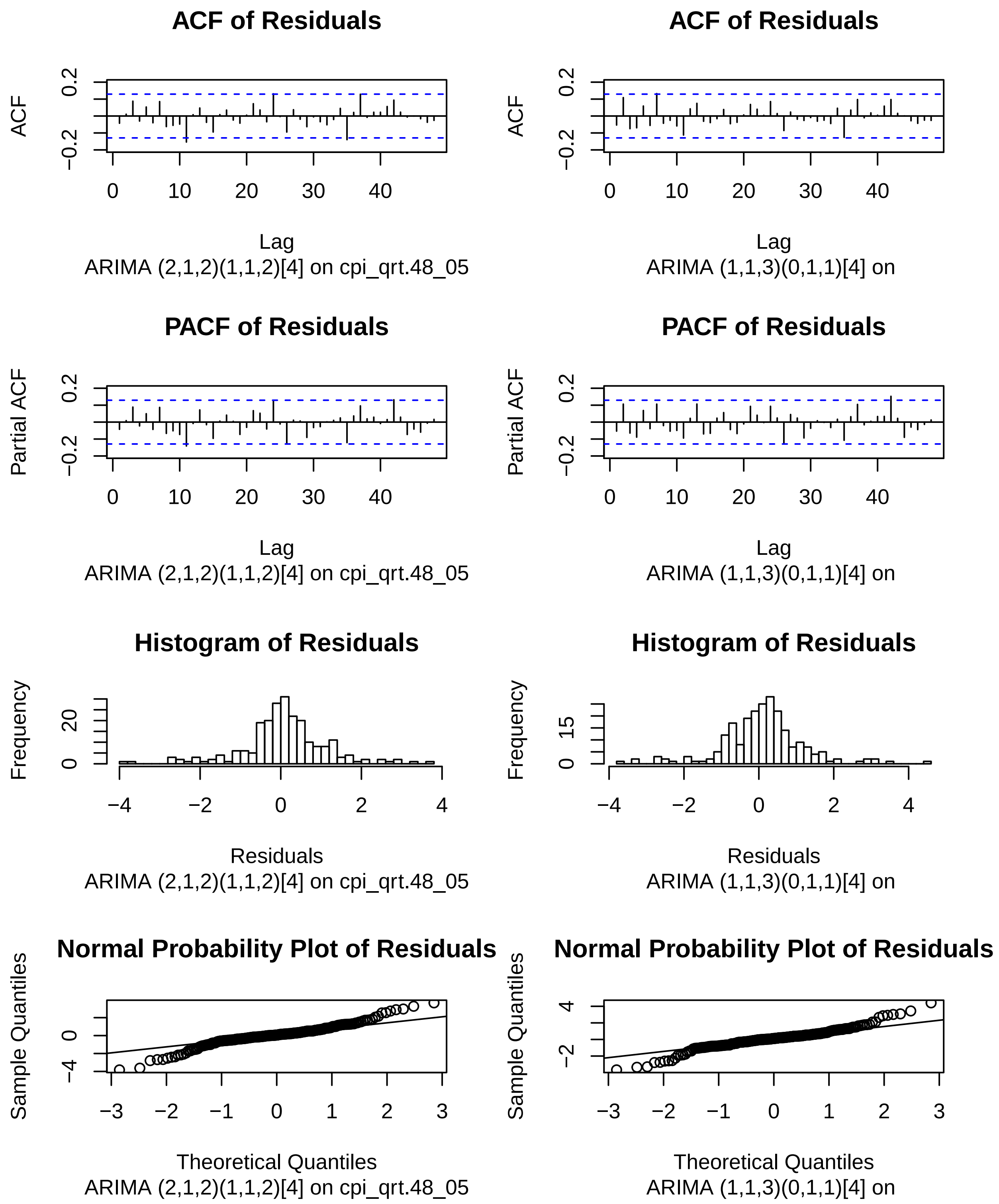
Both models are very similar in all regards.

```
resid.compare(model_qrt.arima.3, model_qrt.auto.2)
```



Residual Values vs. Time

Residual Values vs. Time


ACF of Residuals
ACF
0.2
−0.2
0
10
20
30
40
Lag
ARIMA (2,1,2)(1,1,2)[4] on cpi_qrt.48_05
ACF of Residuals
ACF
Lag
ARIMA (1,1,3)(0,1,1)[4] on
PACF of Residuals
Partial ACF
Lag
ARIMA (2,1,2)(1,1,2)[4] on cpi_qrt.48_05
PACF of Residuals
Partial ACF
Lag
ARIMA (1,1,3)(0,1,1)[4] on
Histogram of Residuals
Frequency
0
20
−4
−2
0
2
4
Residuals
ARIMA (2,1,2)(1,1,2)[4] on cpi_qrt.48_05
Histogram of Residuals
Frequency
0
15
Residuals
ARIMA (1,1,3)(0,1,1)[4] on
Normal Probability Plot of Residuals
Sample Quantiles
0
−4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
ARIMA (2,1,2)(1,1,2)[4] on cpi_qrt.48_05
Normal Probability Plot of Residuals
Sample Quantiles
4
−2
Theoretical Quantiles
ARIMA (1,1,3)(0,1,1)[4] on

```
## [[1]]
## NULL
##
## [[2]]
## NULL
```

None of the models passes the formal diagnostic checks.

```
test.compare(model_qrt.arima.2, model_qrt.arima.3, model_qrt.auto.1, model_qrt.auto.2)
```

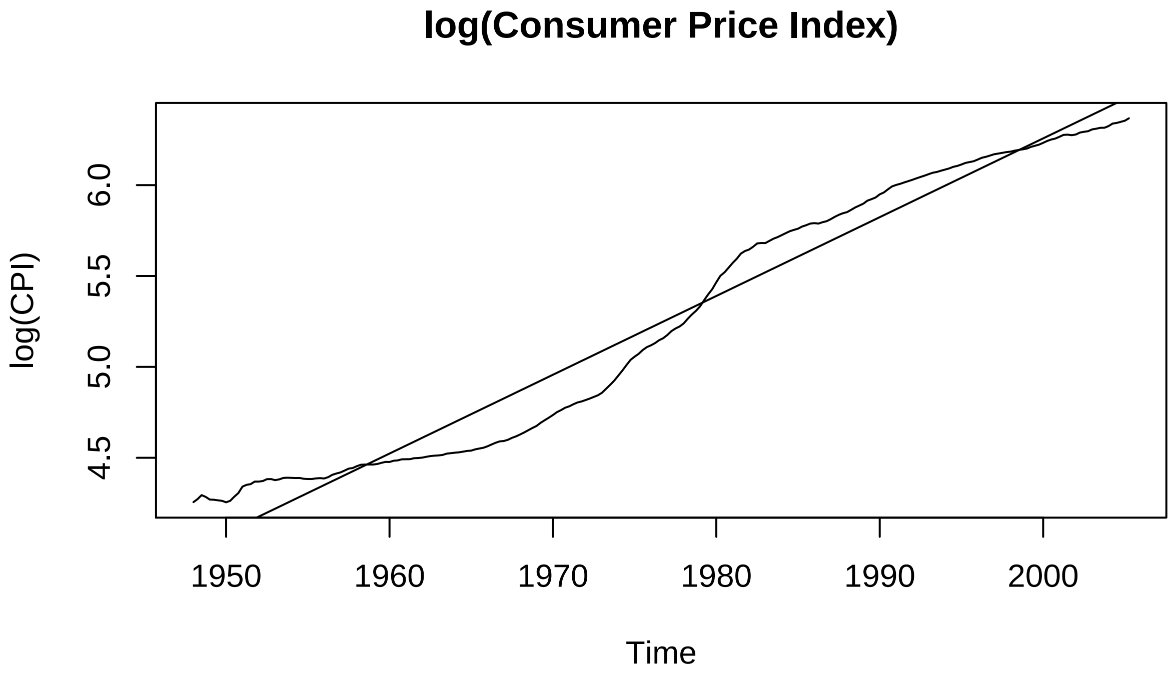
```
##                 Model             Test Statistic    df p.value
## 1  model_qrt.arima.2     Shapiro-Wilk      0.94 10.17    0.00
## 2  model_qrt.arima.2 Anderson-Darling      5.32 15.17    0.00
## 3  model_qrt.arima.2        Box-Ljung     23.36  8.17    0.01
## 4  model_qrt.arima.2        McLeod-Li    130.66 15.17    0.00
## 5  model_qrt.arima.3     Shapiro-Wilk      0.95 10.17    0.00
## 6  model_qrt.arima.3 Anderson-Darling      4.02 15.17    0.00
## 7  model_qrt.arima.3        Box-Ljung     16.34 10.17    0.04
## 8  model_qrt.arima.3        McLeod-Li    119.08 15.17    0.00
## 9   model_qrt.auto.1     Shapiro-Wilk      0.95 10.17    0.00
## 10  model_qrt.auto.1 Anderson-Darling      3.27 15.17    0.00
## 11  model_qrt.auto.1        Box-Ljung     37.85  8.17    0.00
## 12  model_qrt.auto.1        McLeod-Li    126.68 15.17    0.00
## 13  model_qrt.auto.2     Shapiro-Wilk      0.95 10.17    0.00
## 14  model_qrt.auto.2 Anderson-Darling      3.69 15.17    0.00
## 15  model_qrt.auto.2        Box-Ljung     18.55 10.17    0.05
## 16  model_qrt.auto.2        McLeod-Li    121.32 15.17    0.00
```

```
model_qrt.arima <- model_qrt.arima.3
```

Log-Quarterly Model

Because we saw unequal error variance in our residuals, perhaps we should try to stablize it with a log transformation–a common transformation used for economic data.

```
plot(log(cpi_qrt.48_05), main="log(Consumer Price Index)", ylab="log(CPI)")

model_log.lm <- tslm(log(cpi_qrt.48_05) ~ trend)
lines(fitted(model_log.lm))
```



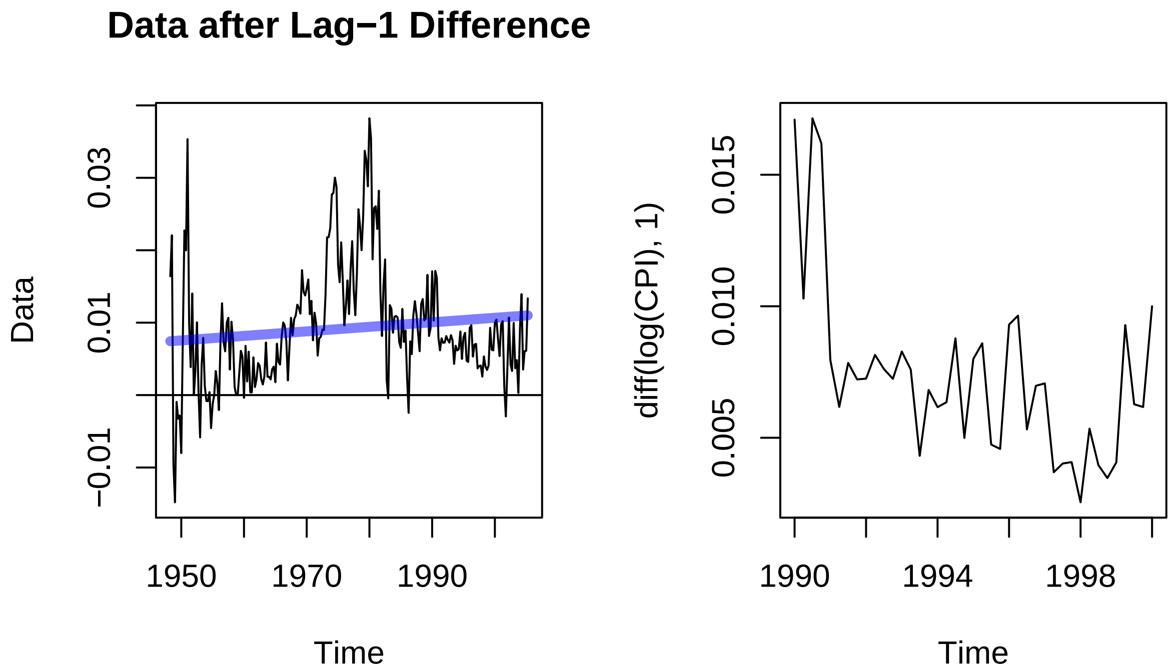
Because this graph shows an linear upward trend, we'll difference at lag 1.

Differencing

```
par(mfrow=c(1,2))
cpi_log.48_05.diff1 <- arma.diff(log(cpi_qrt.48_05), 1)
```

```
## Variance of the original data: 0.5399399
## Variance of the lag-1 differenced data: 6.797263e-05
```

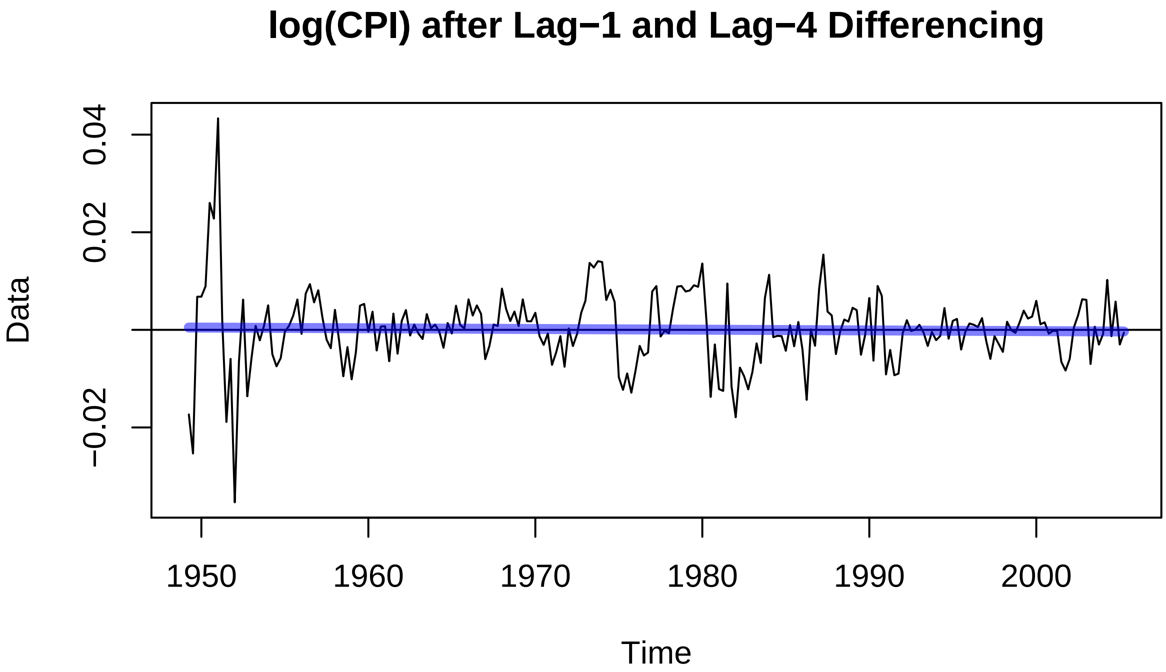
```
plot(window(diff(log(cpi_qrt.48_05), 1), start=c(1990, 1), end=c(2000, 1)),
     ylab="diff(log(CPI), 1)")
```



After differencing at lag 1, we still have a slight trend. Furthermore, after zooming in on a decade, we see there might be seasonal effects. Therefore, we will also consider differencing at lag 4.

De-Trend and De-Seaonsalize

```
log_cpi.diff1_4 <- arma.diff(cpi_log.48_05.diff1, 4,
                             title="log(CPI) after Lag-1 and Lag-4 Differencing")
```

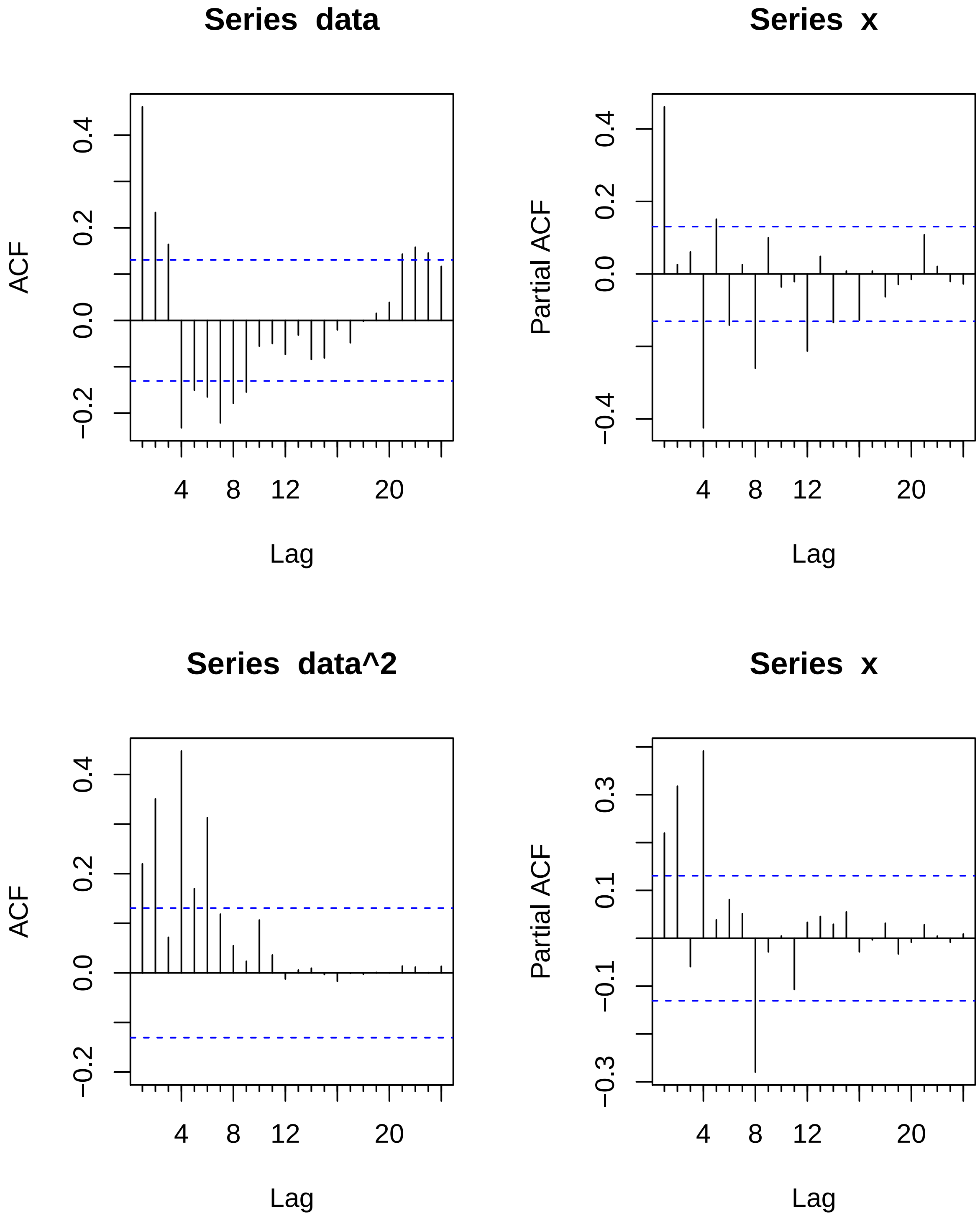


```
## Variance of the original data: 6.797263e-05
## Variance of the lag-4 differenced data: 5.971964e-05
```

Because, an additional lag 4 difference removes the trend and reduces the variance, we should strongly consider using it.

Identification

```r
arma.id(log_cpi.diff1_4, lag.max=24)
```



A decaying ACF pattern suggests an AR model, possibly of order 3. Furthermore, there is also evidence of seasonal effects. A tailing off PACF at seasonal lags suggests SMA components.

There also still appear to be ARCH/GARCH effects, as expected for economic data.

Fitting

For brevity, we will use the auto.arima() function.

```
model_log.auto <- auto.arima(log(cpi_qrt.48_05), d=1, D=1)
summary(model_log.auto)
```

```
## Series: log(cpi_qrt.48_05)
## ARIMA(3,1,3)(1,1,2)[4]
##
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2     ma3    sar1     sma1
##       1.2680  -0.7321  0.3498  -0.5282  0.1234  0.4107  0.2481  -1.5742
## s.e.  0.1247   0.2464  0.2199   0.1233  0.2021  0.1076  0.1722   0.1215
##         sma2
##       0.6770
## s.e.  0.1282
##
## sigma^2 estimated as 2.407e-05:  log likelihood=879.55
## AIC=-1739.11   AICc=-1738.08   BIC=-1704.95
##
## Training set error measures:
##                         ME        RMSE         MAE          MPE      MAPE
## Training set -2.128611e-06 0.004754818 0.003425056 0.0006151963 0.0676276
##                    MASE        ACF1
## Training set 0.09113998 0.006886982
```

The auto.arima() function identified AR(3) and SMA components as suggested, but also fitted MA(3) components which were all significant. It also fitted a non-significant SAR(1) component.

Now, we drop insignificant coefficients.

```
model_log.arima <- arima(log(cpi_qrt.48_05), order=c(2, 1, 3), seasonal=list(order=c(0, 1, 2), period=4
summary(model_log.arima)
```

```
##
## Call:
## arima(x = log(cpi_qrt.48_05), order = c(2, 1, 3), seasonal = list(order = c(0,
##     1, 2), period = 4))
##
## Coefficients:
##           ar1      ar2     ma1     ma2     ma3     sma1     sma2
##       -0.3044  -0.4693  1.0595  1.0594  0.9999  -0.3336  -0.1670
## s.e.   0.0652   0.0681  0.0214  0.0592  0.0591   0.0815   0.0681
##
## sigma^2 estimated as 2.228e-05:  log likelihood = 877.32,  aic = -1738.65
##
## Training set error measures:
##                        ME        RMSE         MAE         MPE       MAPE
## Training set 2.320136e-05 0.004710637 0.003354574 0.001097302 0.06653546
##                   MASE       ACF1
## Training set 0.3439384 0.02845539
```

Comparison of Fit

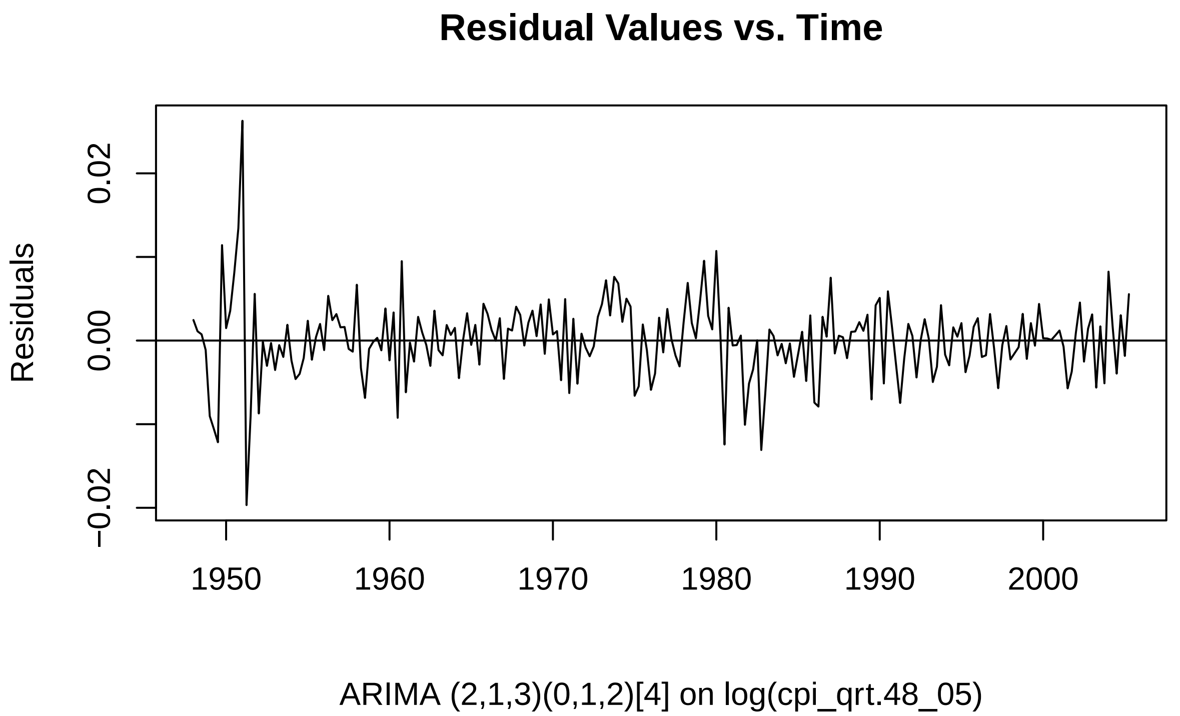
```r
fit.compare(model_log.auto, model_log.arima)
```

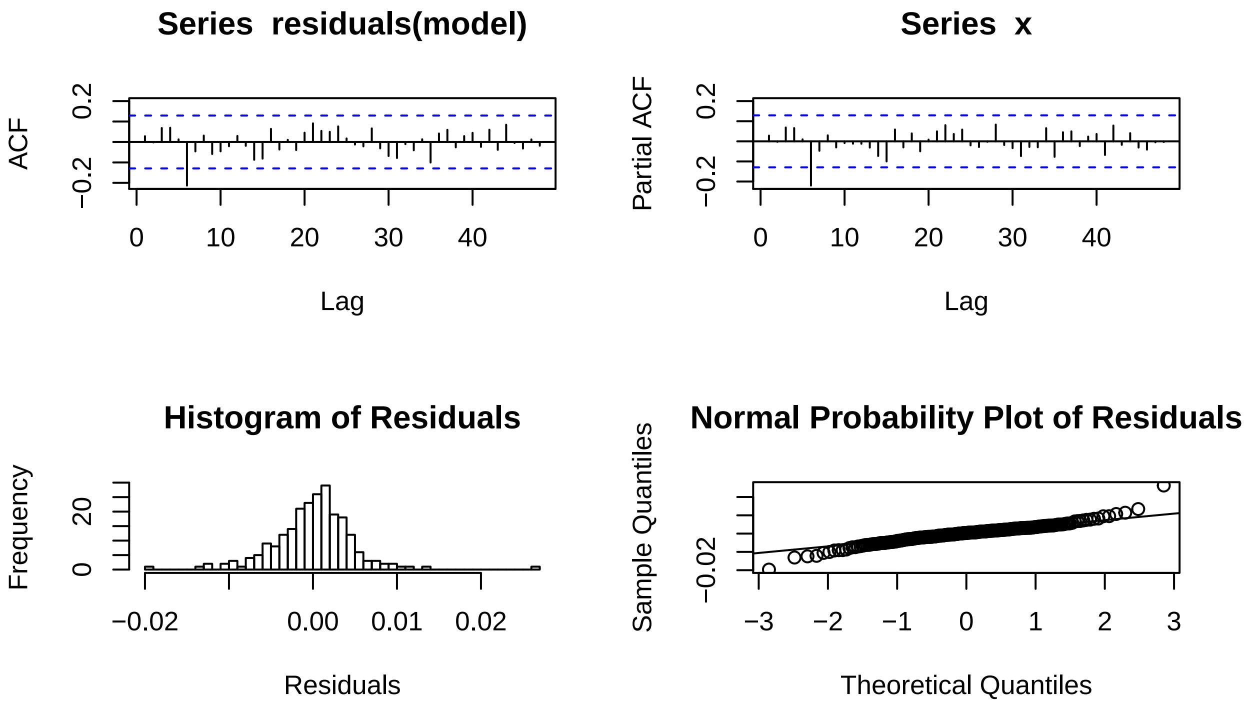
```
##              Model Order Log.Lik      AIC     AICc sigma2 Convergence.
## 1  model_log.auto     9  879.55 -1739.11 -1738.29      0          Yes
## 2 model_log.arima     7  877.32 -1738.65 -1738.14      0          Yes
```

As expected, the model with dropped insignificant coefficients has a lower AICc (barely). Because it is simpler and has a lower AICc, we choose the second log model.

Residual Analysis

```r
arma.diag(model_log.arima)
```





Aside from a large spike at the beginning of the data set, the residual graphs have very nice features. The residuals have outliers, but the tails are much lighter than the standard ARIMA models above.

```
arma.test(model_log.arima)
```

```
## [[1]]
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.94201, p-value = 6.373e-08
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  residuals(model)
## A = 2.3924, p-value = 4.542e-06
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  residuals(model)
## X-squared = 19.35, df = 8.1658, p-value = 0.01437
##
##
## [[4]]
```

```
##
##  Box-Ljung test
##
## data:  residuals(model)^2
## X-squared = 97.095, df = 15.166, p-value = 5.44e-14
```

Despite the improvements, this model still fails most tests with the exception of the Box-Ljung test on the residuals. With a p-value of 0.1217, barely passes our test for White Noise at 90% confidence.

Monthly vs. Quarterly vs. Log-Quarterly Model

Now, we will compare the best monthly and best quarterly models to create a baseline model for future improvements.

Fitting

```
fit.compare(model.arima, model_qrt.arima, model_log.arima)
```

```
##             Model Order Log.Lik      AIC     AICc sigma2 Convergence.
## 1     model.arima     7 -767.71  1551.42  1551.58   0.52          Yes
## 2 model_qrt.arima     7 -342.56   701.11   701.62   1.16          Yes
## 3 model_log.arima     7  877.32 -1738.65 -1738.14   0.00          Yes
```

As for as standard models go, the quarterly model an AICc that is more than half as small as the monthly model. Based on that reason alone, we should use it over the montly model. The much higher log-likelihood supports this decision. Interestingly enough, however, the monthly model has a lower variance.

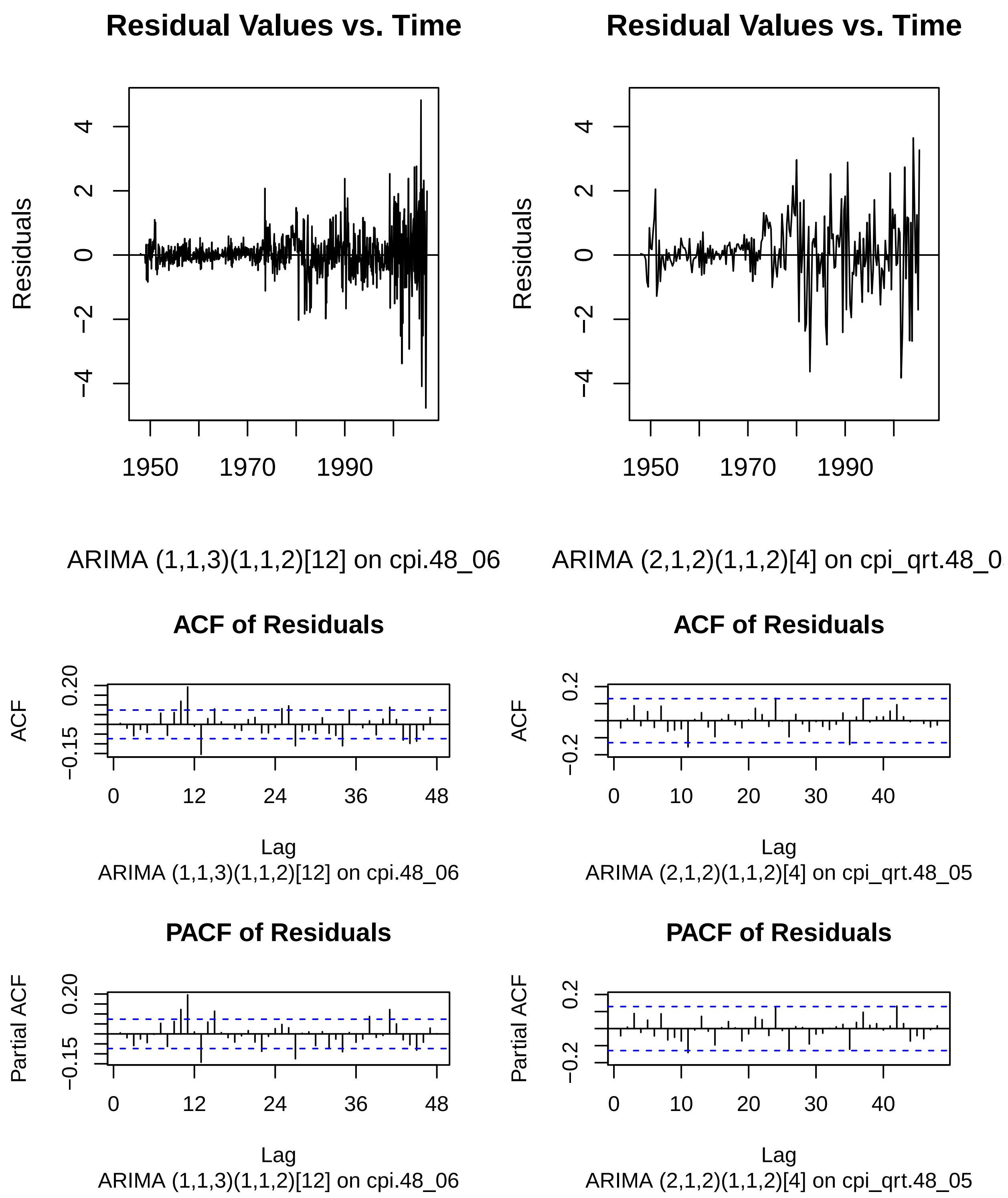
However, in terms of AICc and log-likelihood, the log-quarterly model is the best.

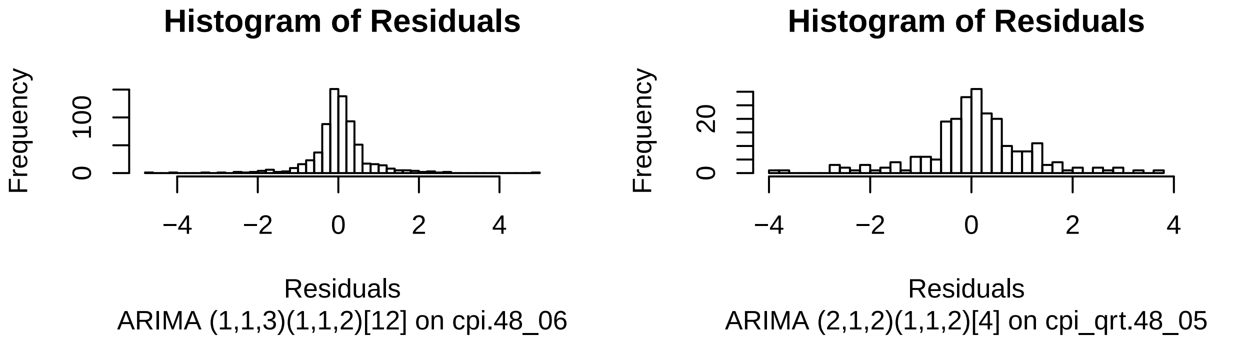
Residuals

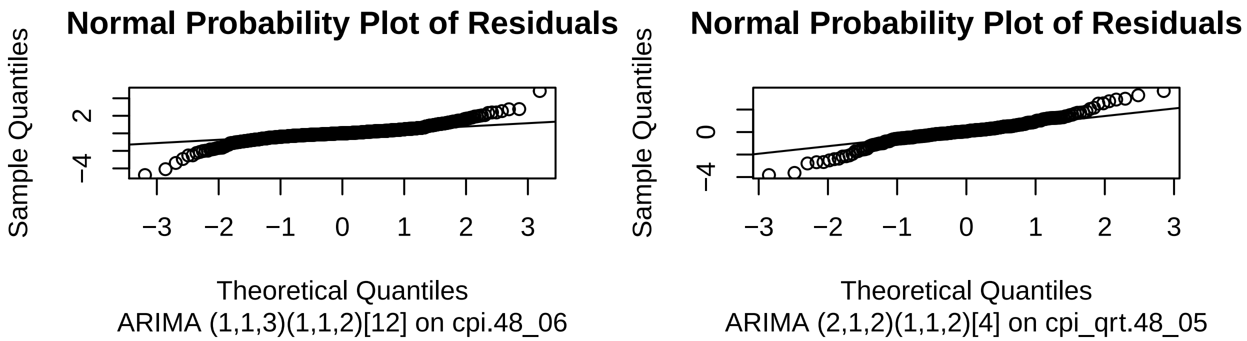
Both the monthly and quarterly models suffer from heteroskedasticity in later years. It should noted however that the change in variance is not exactly linear. The 1990s appear to be a very “calm” period sandwiched in between two volatile decades.

Both models suffer from a heavy-tail distribution of residuals. However, the ACF/PACF graphs show that the quarterly data has almost no significant serial correlation–the same which cannot be said for the monthly data.

```r
resid.compare(model.arima, model_qrt.arima)
```







```
## [[1]]
## NULL
##
## [[2]]
## NULL
```

I will not reprint the results for the log model here as they are only a few pages above.

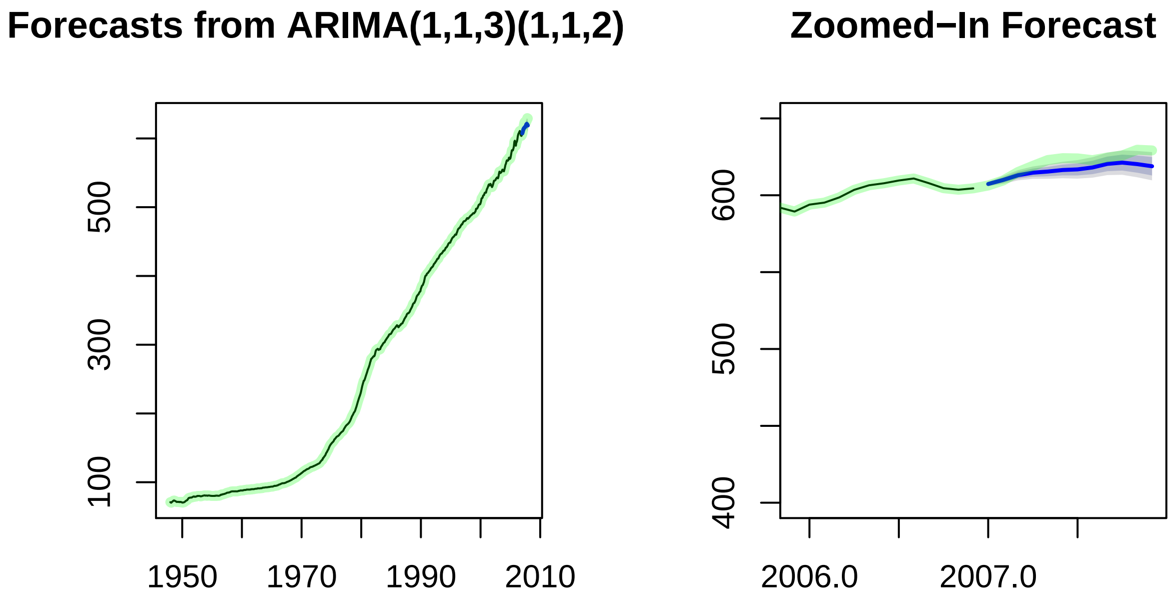
Cross-Validation

As you may have noticed, I left out 10-12 periods in both models. This is so I can try to "predict" those 10 periods using my models to test how accurate they are. This is another step of checking how adequate our models and tells us something that simplying looking at residuals cannot.

Monthly Model

The monthly model fails the cross-validation stage. Even though the testing set is only one year long, the majority of the actual CPI growth occurred outside of the confidence intervals for this model.

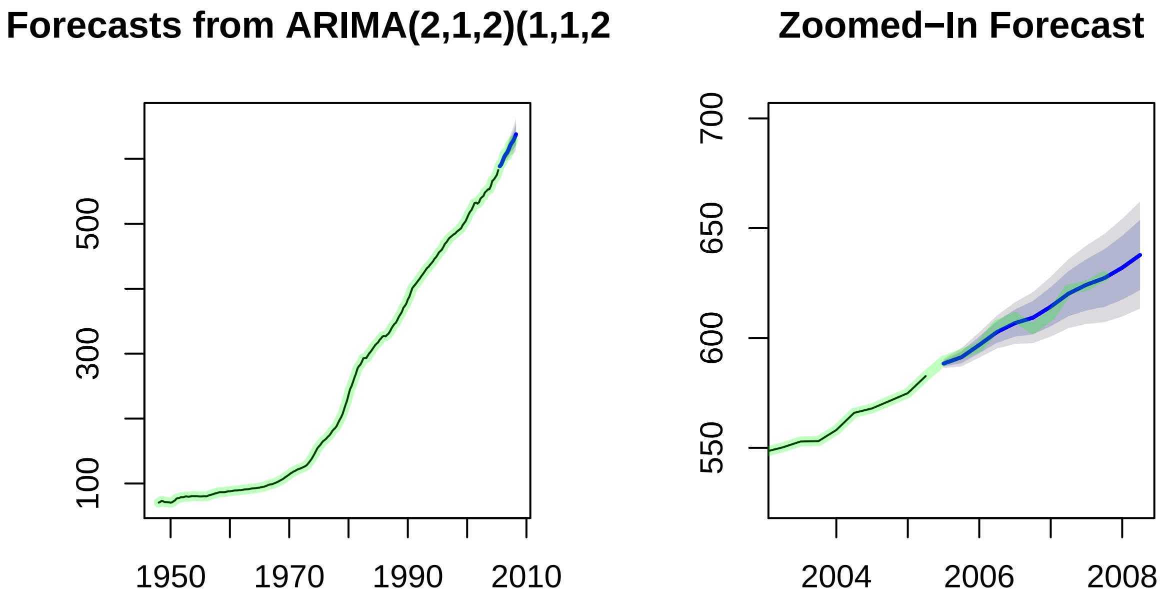
```
# Load full pre-recession dataset
cpi.48_07 <- window(cpi.ts, start=c(1948, 1), end=c(2007, 12), freq=12)

# Forecast
par(mfrow=c(1, 2))
arma.forecast(data=cpi.48_07, forecast=list(object=model.arima, h=12))
arma.forecast(data=cpi.48_07, forecast=list(object=model.arima, h=12),
              zoom=2, zoom.ylim=range(400, 650))
```



Quarterly Model

The quarterly model on the other hand, does very well. Despite being tested against a longer time period (10 observations, or 2.5 years), the predicted mean goes right through the middle of the actual data.

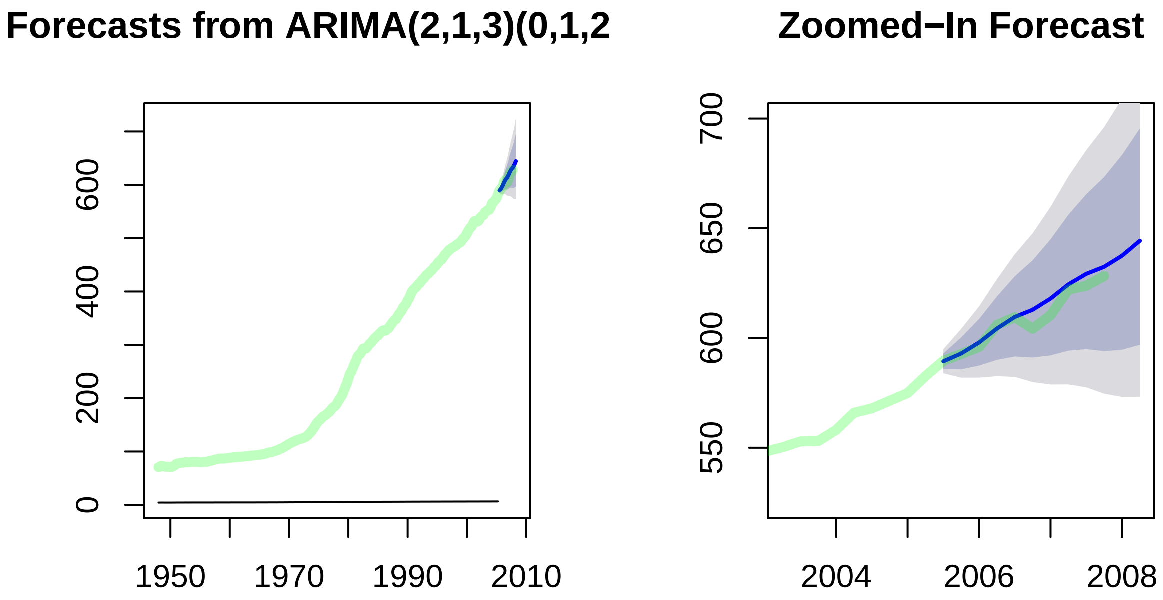
```r
# Load full pre-recession dataset
cpi_qrt.48_07 <- window(cpi_qrt.ts, start=c(1948, 1), end=c(2007, 4), freq=4)

# Forecast
par(mfrow=c(1, 2))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_qrt.arima, h=12))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_qrt.arima, h=12),
              zoom=5, zoom.ylim=range(525, 700))
```



Log-Quarterly Model

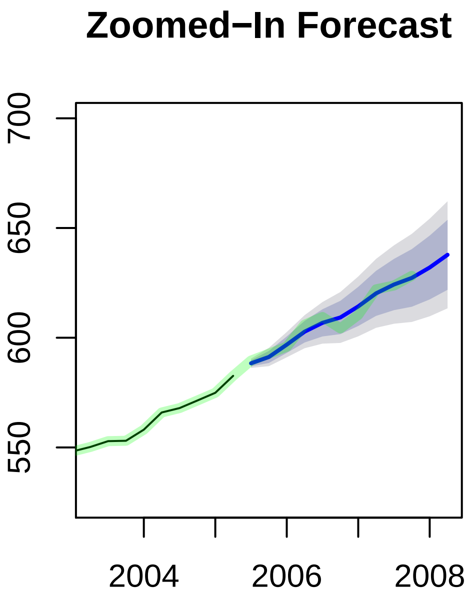
The log-quarterly model also predicts the mean accurately.

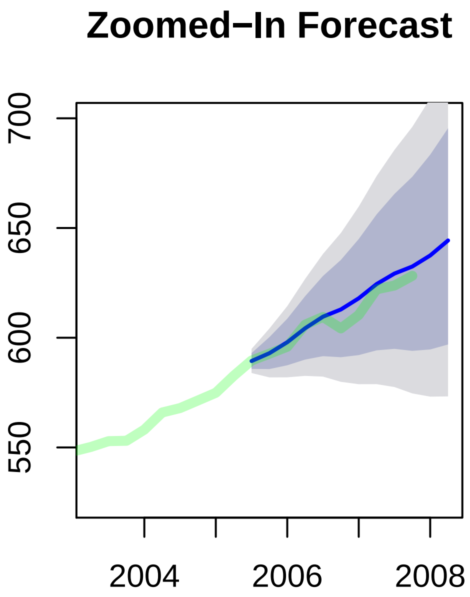
```
# Forecast
par(mfrow=c(1, 2))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_log.arima, h=12, lambda=0))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_log.arima, h=12, lambda=0),
              zoom=5, zoom.ylim=range(525, 700))
```



A drawback of the log-quarterly model (right) is that the confidence intervals are very large. For example, the 95% (dark gray) confidence interval contains the possibility that the CPI might increase by 100 points in just 2.5 years! Even in recent years, this amount of growth usually takes about an entire decade to occur.

```
par(mfrow=c(1, 2))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_qrt.arima, h=12),
              zoom=5, zoom.ylim=range(525, 700))
arma.forecast(data=cpi_qrt.48_07, forecast=list(object=model_log.arima, h=12, lambda=0),
              zoom=5, zoom.ylim=range(525, 700))
```





Therefore, for accessing the impact of the recession, I will prefer the conclusions given by the untransformed model.

Fitted Model

The final model based on quarterly data is a SARIMA(2, 1, 2)x(1, 1, 2) model which can be written as

$$(1-0.34B-0.47B^2)(1+0.87B^4)(1-B)(1-B^4)CPI = (1+0.28B-0.67B^2)(1+0.21B^4-0.76B^5)\epsilon_t$$

where the ϵ_t are white noise.

Spectral Analysis

Now that I have identified by preferred model, I will perform spectral analysis on the model and the dataset. This is a non-linear alternative to the Box-Jenkins approach which attempts to model data as a series of sines and cosines.

Periodogram

```
periods <- periodogram(cpi.48_07)
```

There does not appear to be much periodicity to the data. This graph shows that there are no dominant frquencies which may be modeled by trigonometric functions.

Residuals of Fitted Model

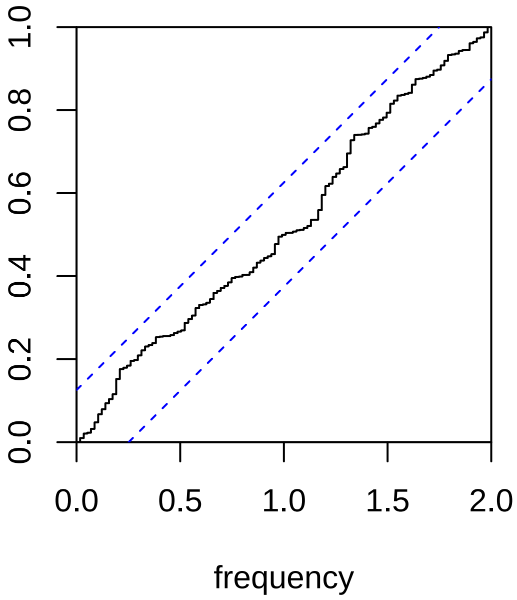
```
fisher.g.test(residuals(model_qrt.arima))
```

```
## [1] 0.9673159
```

My preferred model passes the Fisher test for periodicity. However, this should not be surprising because the original data set did not appear to be periodic in nature.

KS Test of Residuals

```
cpgram(residuals(model_qrt.arima), main="")
```



Again, the KS test for periodicity confirms the results of the Fisher test.

Impact of the 2008 Recession

Re-Train Model

Before accessing the impact of the 2008 recession, I will re-train my models to include all data prior to 2008.

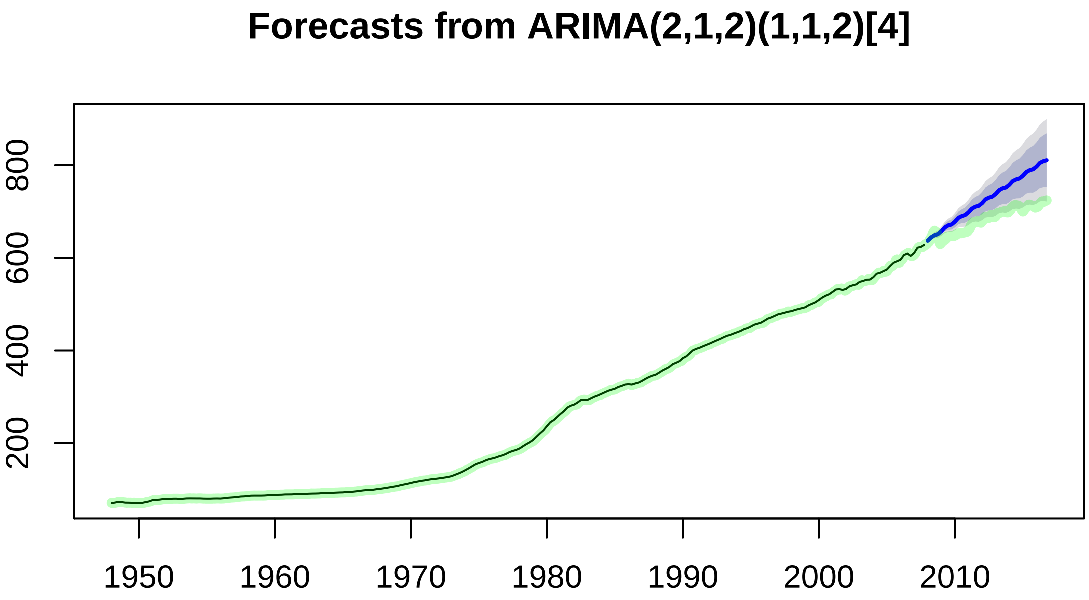
```
model_qrt08.arima <- arima(x=cpi_qrt.48_07, order=c(2, 1, 2), seasonal=list(order=c(1, 1, 2), period=4)
summary(model_qrt08.arima)
```

```
##
## Call:
## arima(x = cpi_qrt.48_07, order = c(2, 1, 2), seasonal = list(order = c(1, 1,
##     2), period = 4))
##
## Coefficients:
##          ar1     ar2     ma1      ma2     sar1    sma1     sma2
##       0.3368  0.4747  0.2824  -0.6678  -0.8706  0.2116  -0.7590
## s.e.  0.1041  0.0678  0.0962   0.0833   0.0659  0.0937   0.0698
##
## sigma^2 estimated as 1.686:  log likelihood = -399.8,  aic = 815.6
##
## Training set error measures:
##                       ME     RMSE       MAE        MPE      MAPE      MASE
## Training set 0.09629587 1.284779 0.8644458 0.05647795 0.3643527 0.3553839
##                      ACF1
## Training set -0.07230581
```

Everything looks good to go.

Recession

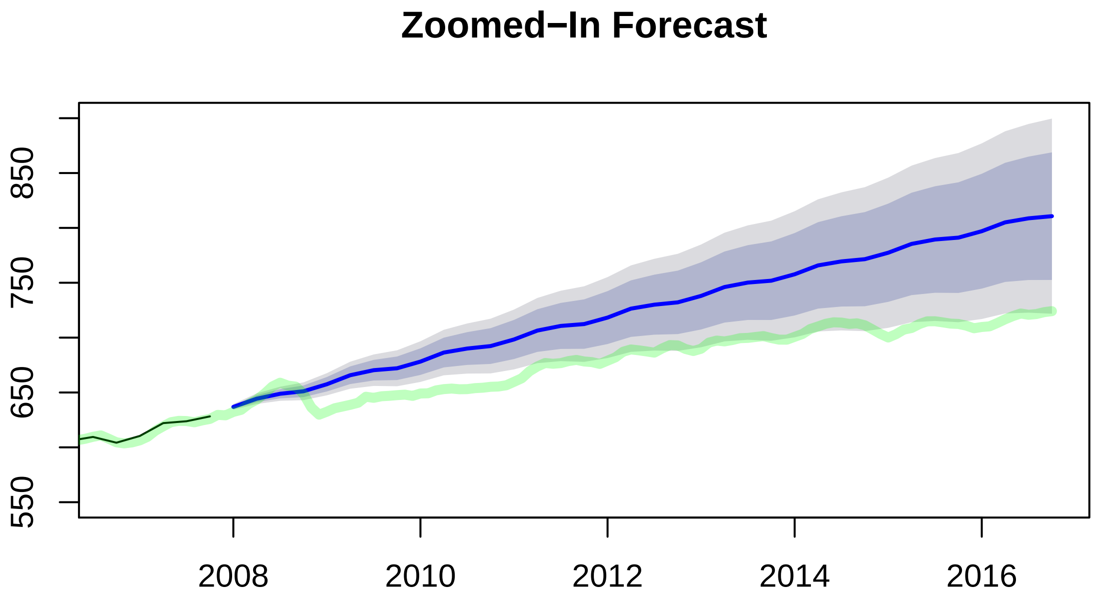
```
arma.forecast(data=cpi.ts, forecast=list(object=model_qrt08.arima, h=4*9))
```



Zoom-In

The actual path of the Consumer Price Index straddles the lower 80% confidence interval based on our model created from pre-recession data. The CPI took a large plunge during the height of the recession, but appears to be recovering its original trend before the recession.

```
arma.forecast(data=cpi.ts, forecast=list(object=model_qrt08.arima, h=4*9),
              zoom=10, zoom.ylim=range(550, 900))
```



Therefore, this model provides some evidence that the recession had a significant impact on the Consumer Price Index.

Appendix

Source code for functions.r

This was a script of helper functions I used in this report. Because it is rather long, it is sent as a separate attachment.

References

Consumer Price Index

The Consumer Price Index can be downloaded from https://www.bls.gov/cpi/. "‘