Analysis of U.S. Monthly Unemployment Rate

Carlos Sathler

3/31/2019

Contents

```r
# source module with time series data
suppressMessages(library(astsa))
# source plotting module
suppressMessages(library(ggplot2))
# source time series forecasting module
suppressMessages(library(forecast))
# source package to use tidy
suppressMessages(library(broom))
```

Motivation

Apply time series techniques and demonstrate ability to perform basic time series analysis using R.

Dataset selection

In this section of the code I explore the datasets delivered with package astsa. My goals is to chose a time series suitable for ARMA/ARIMA model, with enough oberservations. Code was "deactivated" after I made the final dataset selection for the project.

```r
if (1==2) {
datasets = data(package = 'astsa')
datasets = data.frame(datasets$results)
for (item in as.character(datasets$Item)) {
  dataset = get(item)
  if (class(dataset) == 'ts') {
    print(paste(item, length(dataset)))
  }
}
}
```

Data Description

I will use dataset "U.S. Unemployment Rate" (UnempRate R timeseries), which according to the astsa package documentation contains the "monthly U.S. unemployment rate in percent unemployed (Jan, 1948 - Nov, 2016)". The dataset contains 827 observations. More information can be found in the astsa package help pages, and in the website http://www.stat.pitt.edu/stoffer/tsa4/.

The series starts on January of 1948 and ends on November 2016. Frequency is monthly.

Mean unemployment rate is 5.812%, minimum is 2.4% and maximum is 11.4%.

Standard deviation is 1.6854.

```
# confirm dataset is time series type
class(UnempRate)
```

```
## [1] "ts"
```

```
# print number structure of data and number of observations
str(UnempRate)
```

```
##  Time-Series [1:827] from 1948 to 2017: 4 4.7 4.5 4 3.4 3.9 3.9 3.6 3.4 2.9 ...
```

```
# print summary of ts
summary(UnempRate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.400   4.700   5.600   5.812   6.900  11.400
```

```
# print standard deviation
sd(UnempRate)
```

```
## [1] 1.685388
```

```
# print frequency
frequency(UnempRate)
```

```
## [1] 12
```

```
# print start and end dates
u.start = start(UnempRate)
u.start
```

```
## [1] 1948    1
```

```
u.end = end(UnempRate)
u.end
```

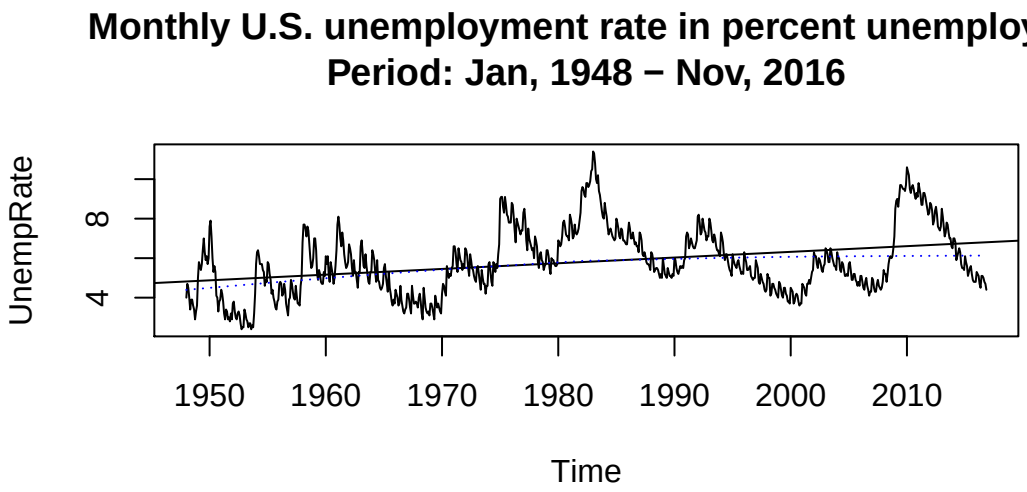
```
## [1] 2016   11
```

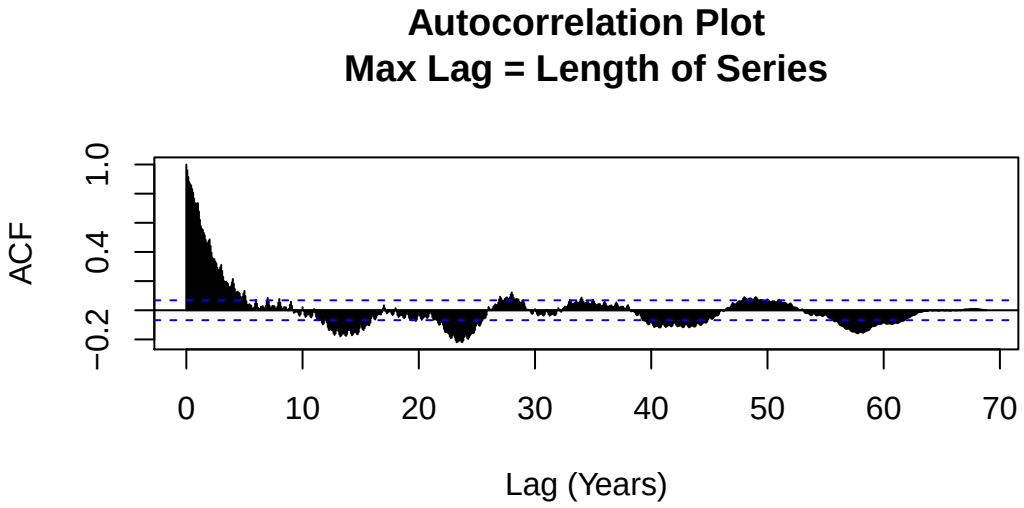
Data Exploration

Based on the time series and the auto correlation plots of Monthly Unemployment Rate between January 1948 and November 2016 we can state the following:

- The series is nonstationary: neither mean nor variance are constant over time.
- The series shows stochastic trends of various durations; the auto correlation plot displaying many periods of slow decay is consistent with the existance of trends.
- Trends have a duration of 5 to 10 years on average, are mostly declining trends, and typically follow an abrupt increase in unemployment.
- On average, trends are shorter from 1948 and 1980, and longer thereafter.
- The series is more choppy from 1948 to 1970, but variance is smaller during that period.
- Variance seems to increase particularly after 1975, when unemployment rate surpasses 8% for the first time.
- After 1970, lower rates of unemployment stay consistently above 4%, on average, and higher unemployment rates are above 8%.
- The series lowess smoother (dashed blue line) shows a slight steady rise in unemployment rates from 1948 through around 1975. After that the rate average seems to stabilize.
- We note the data does not seem to need transformation, since there is no exponential growth.

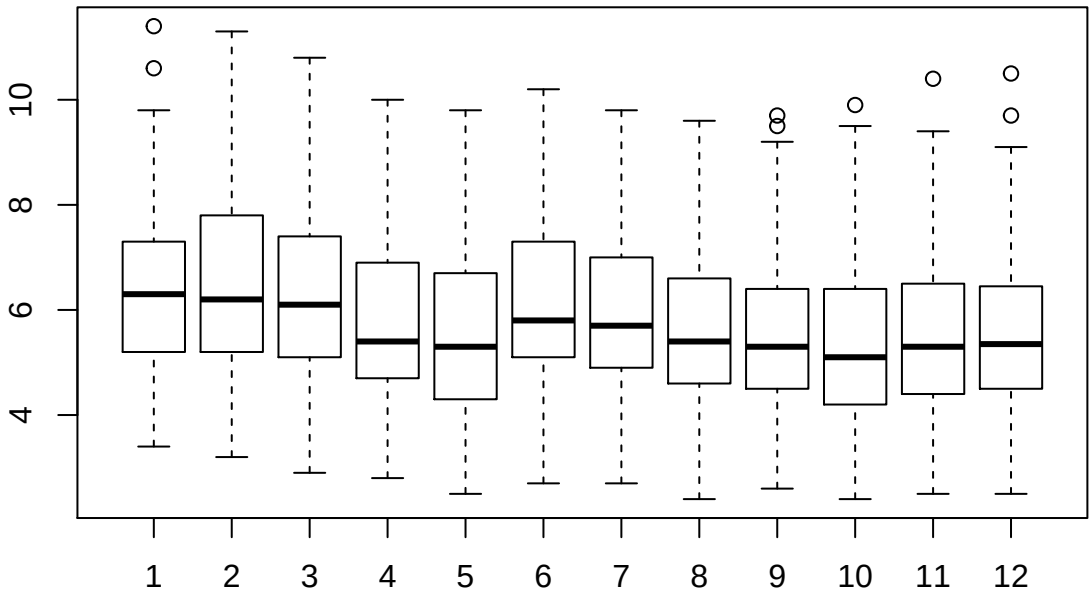
```
par(mfrow=c(2,1), mai = c(1, 1, 1, 1))
plot.ts(UnempRate, main="Monthly U.S. unemployment rate in percent unemployed\nPeriod: Jan, 1948 - Nov,
abline(reg=lm(UnempRate~time(UnempRate)), lty=1)
lines(lowess(UnempRate, f=1), col='blue', lty=3)
acf(UnempRate, length(UnempRate), xlab='Lag (Years)', main='Autocorrelation Plot\nMax Lag = Length of S
```





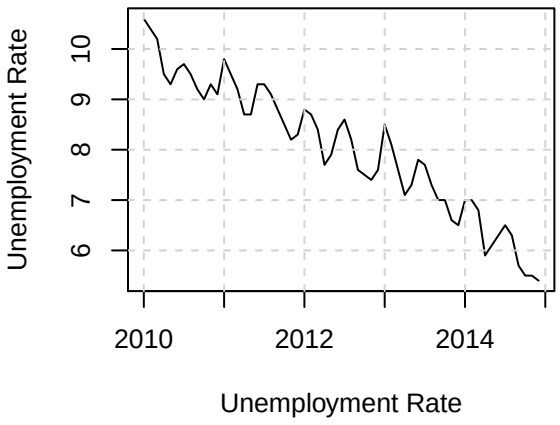
Below we see the monthly variation of unemployment rate across cycles. On average we notice that unemployment is higher in winter and summer months.

```
boxplot(UnempRate~cycle(UnempRate))
```



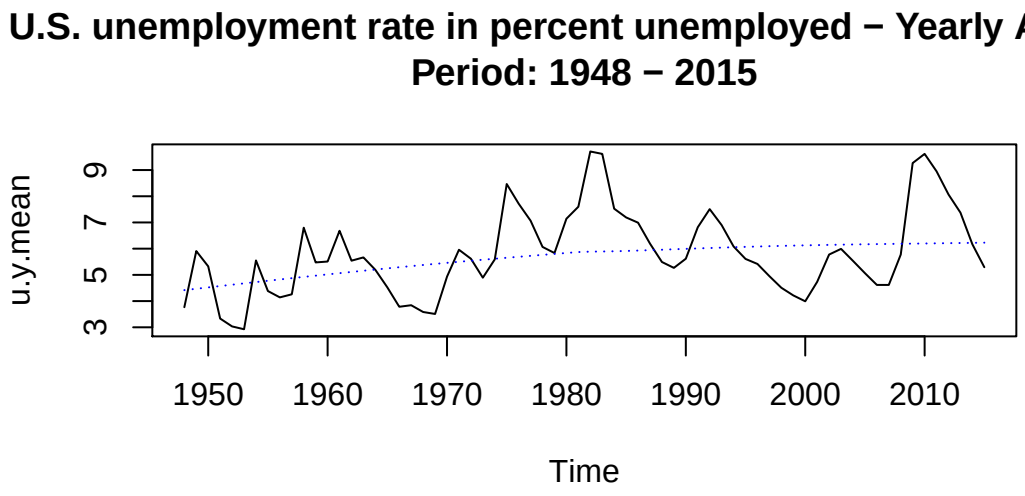
Next we look at a few 5 year windows to analyze seasonality. It appears rates are higher in the winter, likely following Christmas layoffs of temporary workers, and summer months, likely due to slower economic activity during summer vacations.

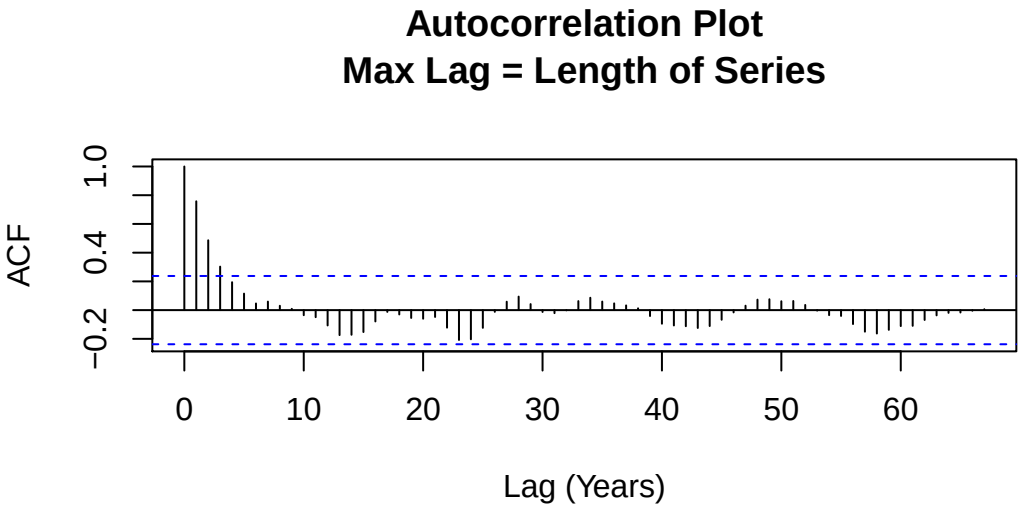
```r
ylab = 'Unemployment Rate'
xlab = '5 Year Sample Window'
blank=''
par(mfrow=c(2,2)) #, mai = c(1, 1, 1, 1))
plot(window(UnempRate, start=c(1950,1), end=c(1954,12), freq=12), ylab=ylab, xlab=ylab); grid(lty=2)
plot(window(UnempRate, start=c(1975,1), end=c(1979,12), freq=12), ylab=ylab, xlab=ylab); grid(lty=2)
plot(window(UnempRate, start=c(1980,1), end=c(1984,12), freq=12), ylab=ylab, xlab=ylab); grid(lty=2)
plot(window(UnempRate, start=c(2010,1), end=c(2014,12), freq=12), ylab=ylab, xlab=ylab); grid(lty=2)
```



The plot of yearly averages of unemployment rate (below), shows a slight upward trend during the observed period. The trend is confirmed by both the gradual decay in the autocorrelation plot, as well as the series smoother. The increase in variance after 1970 is more easily noticeable in the plot of yearly averages.

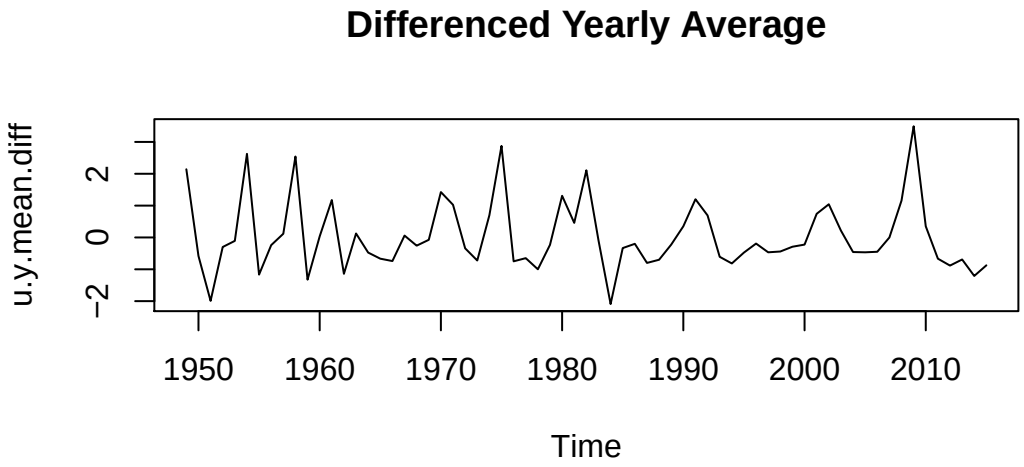
```r
u.y.mean = aggregate(UnempRate)/12
par(mfrow=c(2,1), mai = c(1, 1, 1, 1))
plot.ts(u.y.mean, main="U.S. unemployment rate in percent unemployed - Yearly Average\nPeriod: 1948 - 20
lines(lowess(u.y.mean, f=1), col='blue', lty=3)
acf(u.y.mean, length(u.y.mean), xlab='Lag (Years)', main='Autocorrelation Plot\nMax Lag = Length of Seri
```

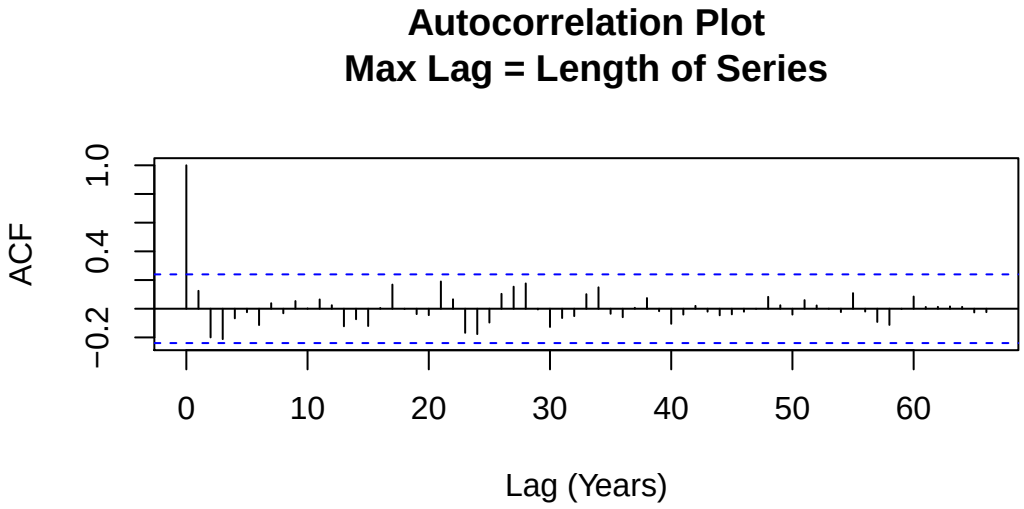




Differencing the anual average series (next page) creates a stationary series, similar to white noise.

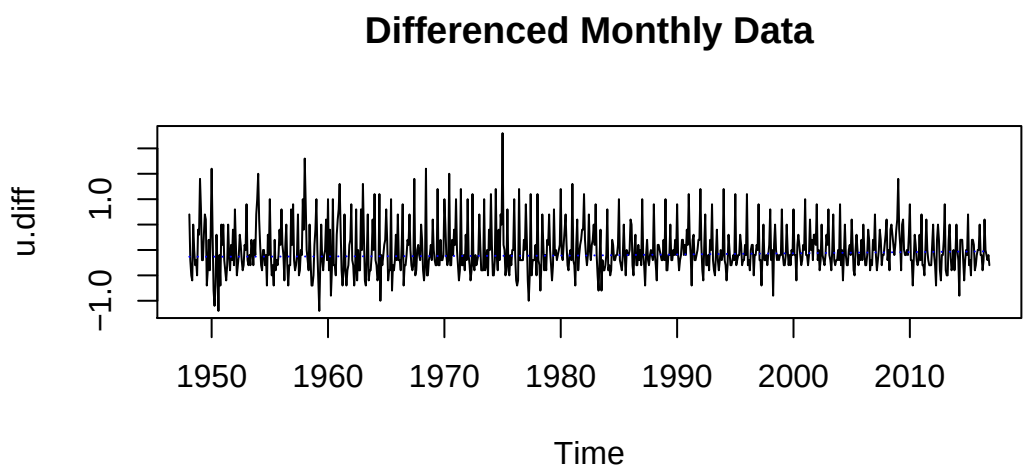
```r
u.y.mean.diff = diff(u.y.mean)
par(mfrow=c(2,1), mai = c(1, 1, 1, 1))
plot.ts(u.y.mean.diff, main="Differenced Yearly Average")
acf(u.y.mean.diff, length(u.y.mean.diff), xlab='Lag (Years)', main='Autocorrelation Plot\nMax Lag = Leng
```

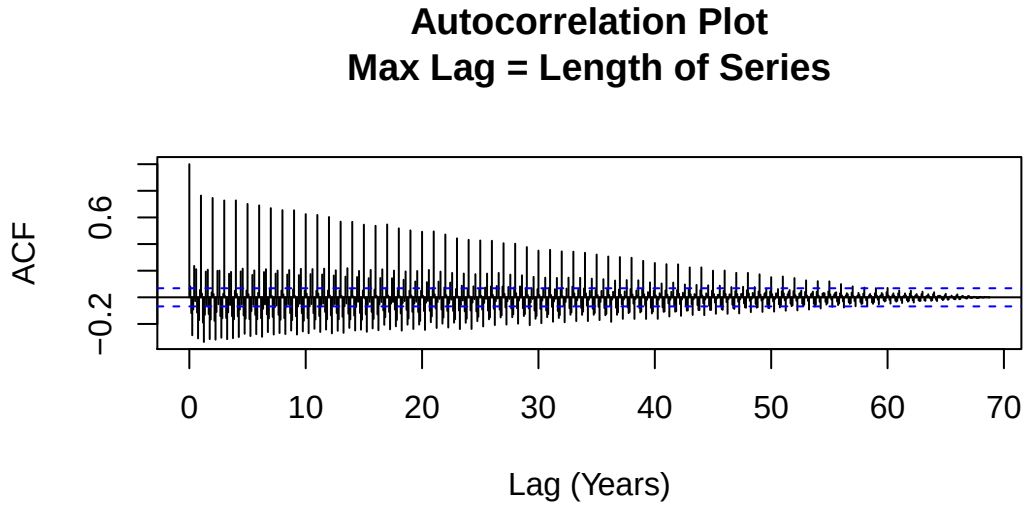




Below I show that differencing the monthly series, however, does not produce a stationary series. At this point, it makes sense to detrend the data first. But instead of doing that manually, I will use the decompose function.

```r
par(mfrow=c(2,1), mai = c(1, 1, 1, 1))
u.diff = diff(UnempRate)
plot.ts(u.diff, main="Differenced Monthly Data")
lines(lowess(u.diff, f=1), col='blue', lty=3)
acf(u.diff, length(u.diff), xlab='Lag (Years)', main='Autocorrelation Plot\nMax Lag = Length of Series')
```





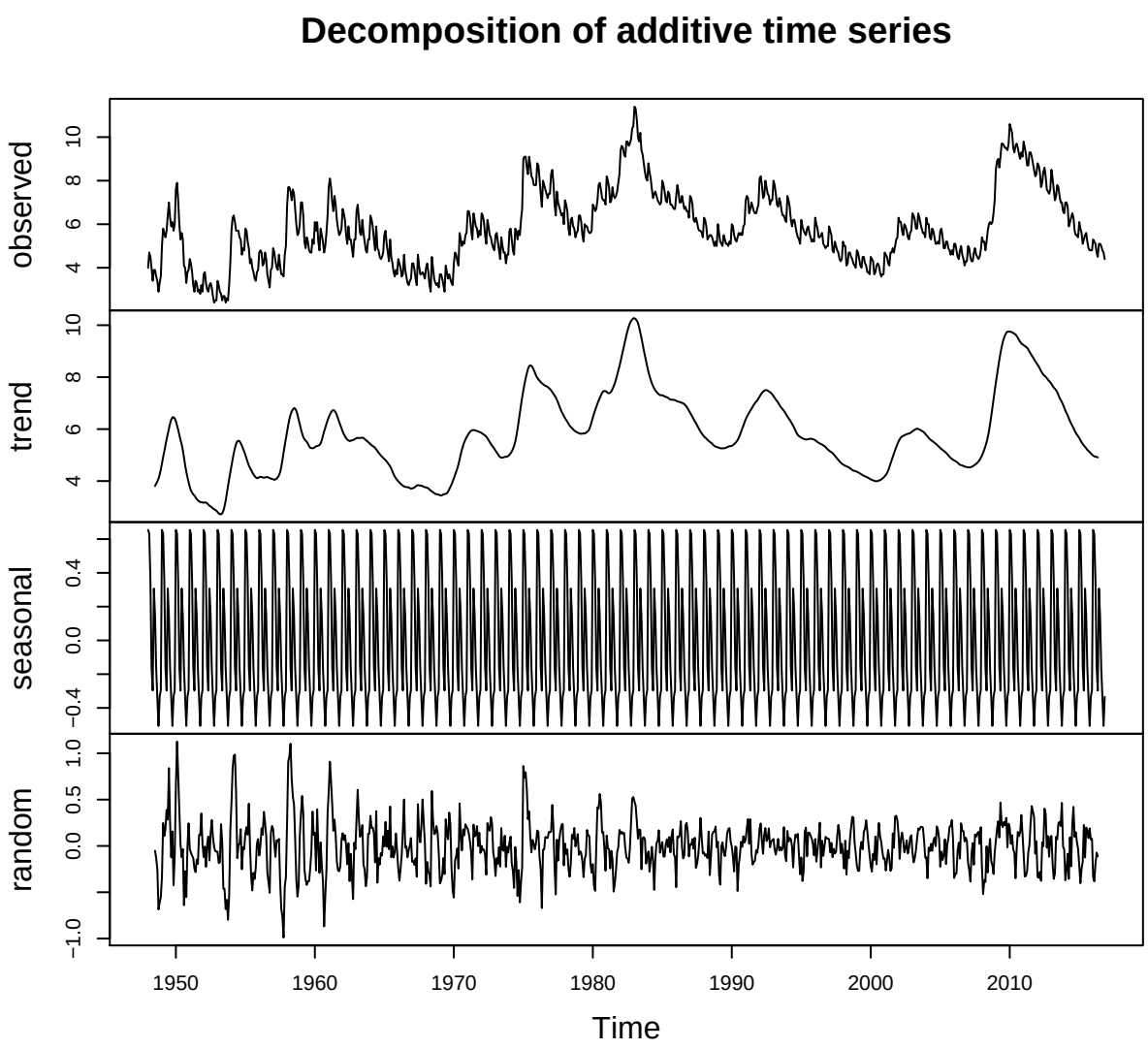
Data Decomposition

Here I perform data decomposition using the R decompose function. I noticed that the acf plot of the random component (next page) of the series extracted via the decompose funtion shows a large number of ρ_k values outside the $|5\%|$ threshold, suggesting autocorrelation in the random component of the decomposed series.

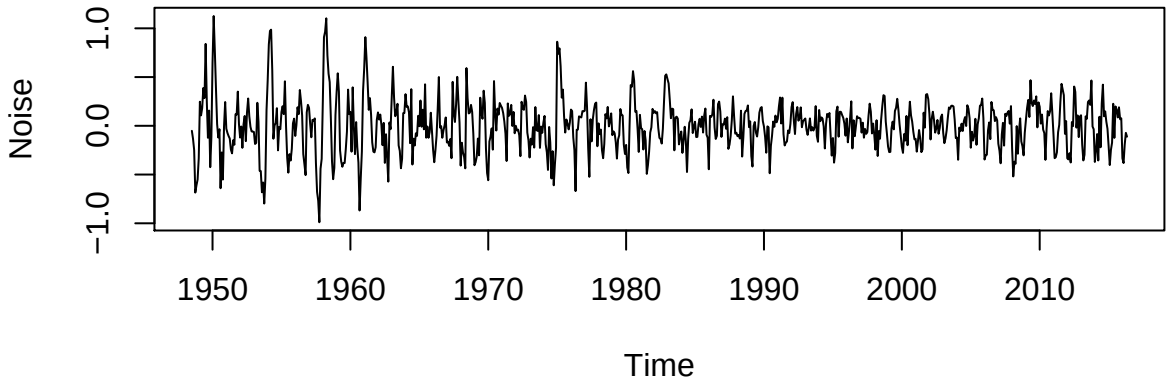
Additionally, the plot of the random component shows heteroscedasticity not only at the extremes of the data range, but also towards the middle of the series. All things considered, there is reason to believe the random component is not pure noise.

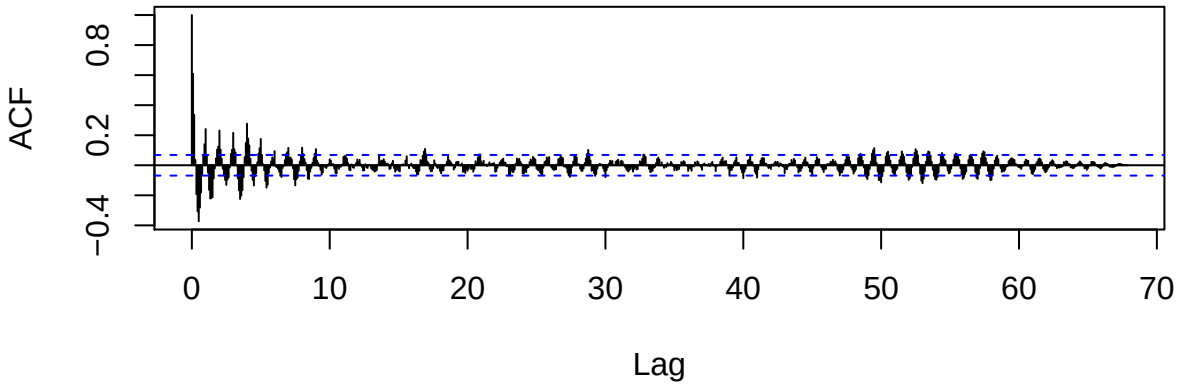
Still, albeit imperfect, smoothing through decomposition produced a trend component that confirms the stochastic nature of the trend that I highleted in the previous section.

```
u.d = decompose(UnempRate, type='additive')
u.d.random = ts(u.d$random[7:821], start=c(1948,7), end=c(2016,5), freq=12)
plot(u.d)
```




```
par(mfrow=c(2,1))
plot(u.d.random, ylab = 'Noise')
acf(u.d.random, length(u.d.random), main='')
```





Analysis of the seasonal component of the decomposed series is included next, for a sample year. It highlights our finding during exploratory analysis. The spike in unemployement at the beginning of the year likely reflects slow down in the economy following Christmas and end of year holidays. The spike in the summer likely reflects slow down in economic activities due to summer vacation.

```
u.d.seasonal = ts(u.d$seasonal[7:821], start=c(1948,7), end=c(2016,5), freq=12)
plot(window(u.d.seasonal, start=c(1970,1), end=c(1970,12)), ylab='Seasonal Effect', xlab='Sample Year')
```

Regression

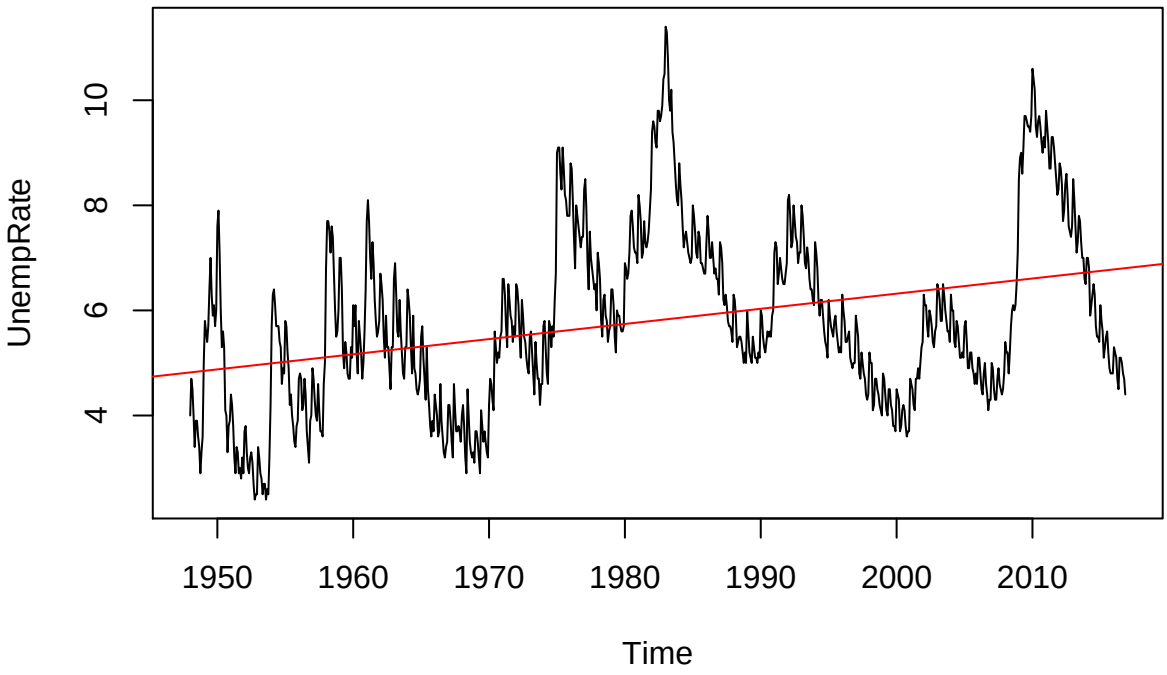
I will fit a linear model to the data: $y_t = \beta_0 + \beta_1 x_t + w_t$

The fit reflects the slight upward trend in the data over the years. That had already been captured by the abline required for the exploratory data analsys section of this report.

```
fit = lm(UnempRate ~ time(UnempRate))
tidy(fit)
```

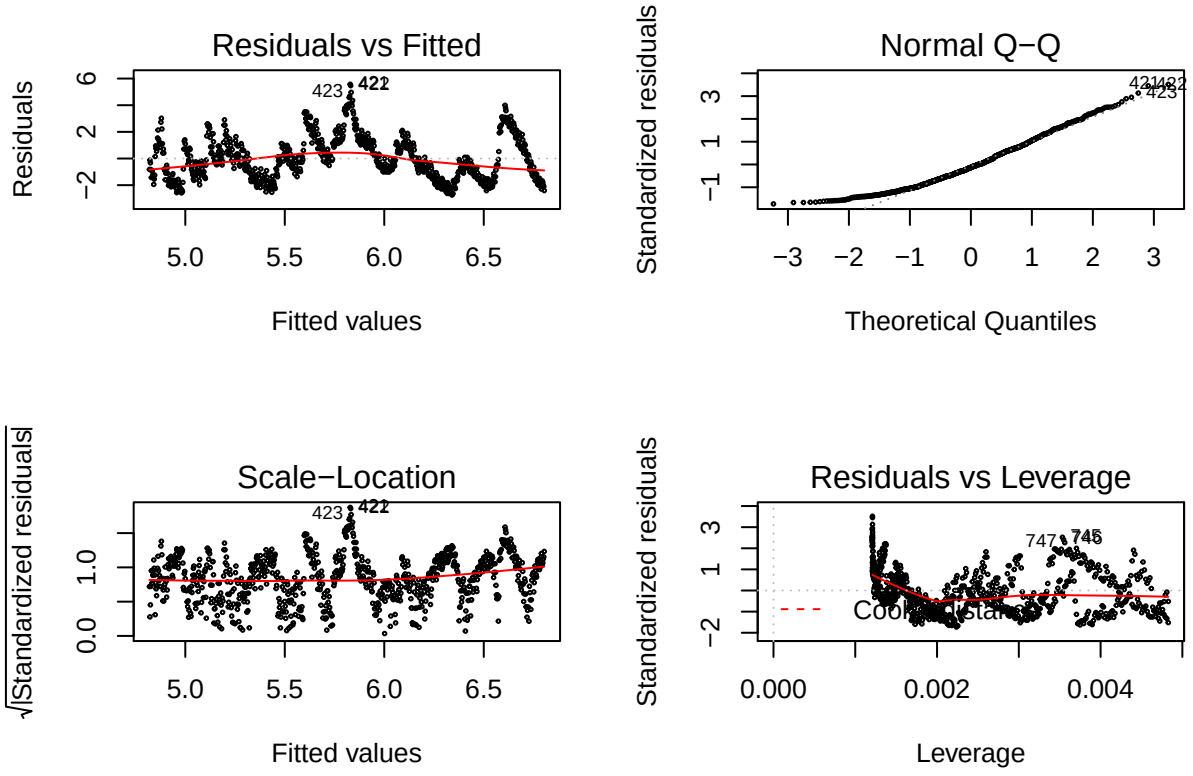
```
## # A tibble: 2 x 5
##   term             estimate std.error statistic  p.value
##   <chr>               <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     -51.3      5.49        -9.34 8.48e-20
## 2 time(UnempRate)   0.0288   0.00277     10.4  6.76e-24
```

```
fit.df = augment(fit)
plot(UnempRate)
abline(coef = fit$coefficients, col='red')
```



The top left plot of residuals vs. fitted values shows a clear pattern. Residuals are not just random noise (w_t); they contain information, so the model is not a good fit.

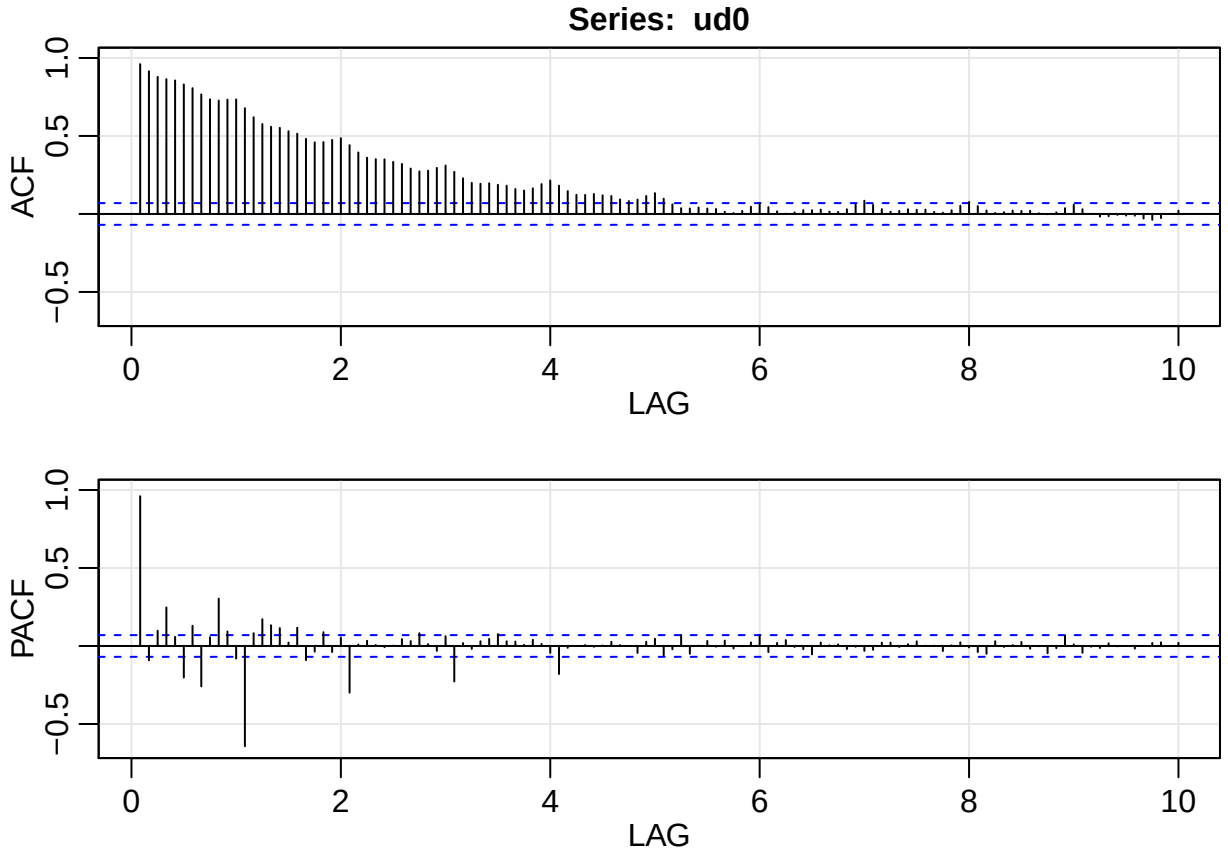
```
par(mfrow=c(2,2))
plot(fit, cex=0.3)
```



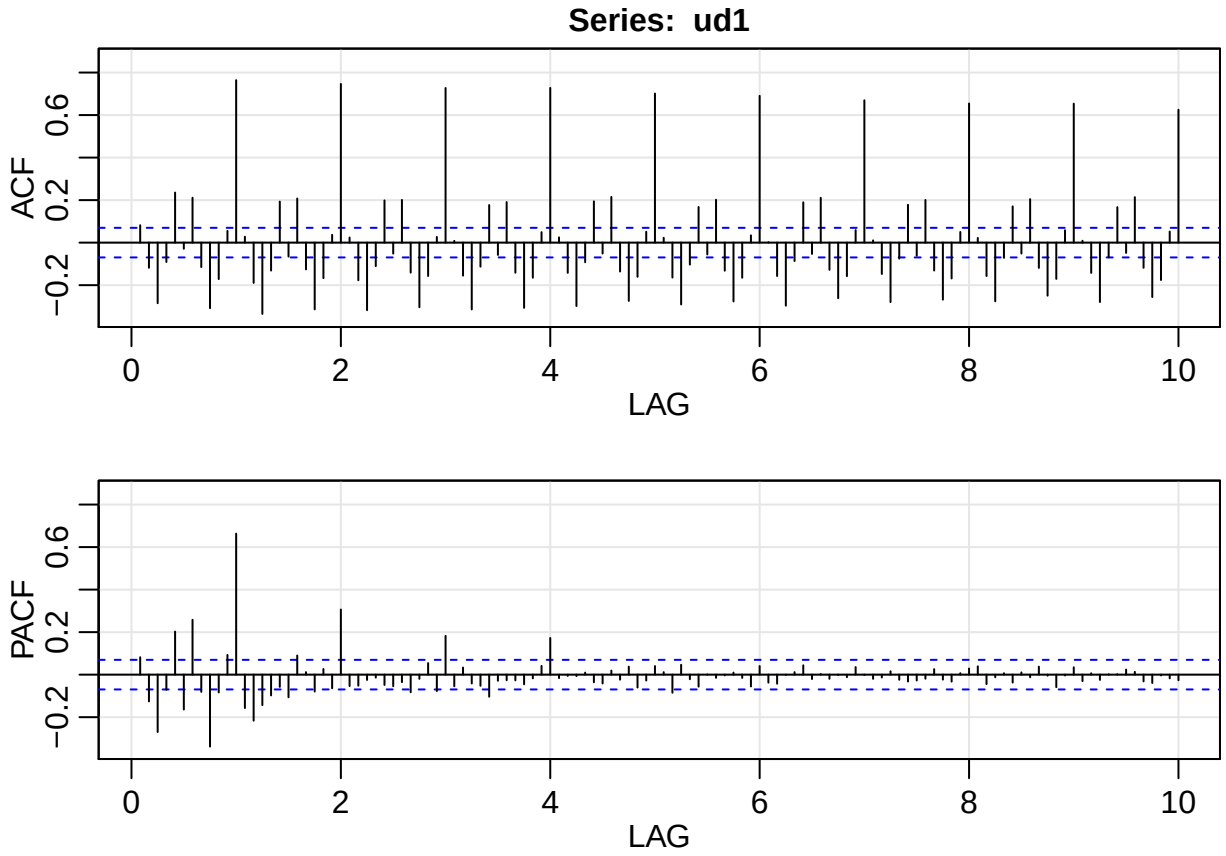
ARMA/ARIMA model

Here I produce 2 ACF/PACF plot pairs: one for the original series, and one for the differenced series. (Note that acf2 produces plots shifted 1 lag to the right). The first plot pair (no differencing) rules out simple AR(p) and MA(p) models, since both ACF and PACF tail off. The second plot didn't simplify the problem that much since, again, both ACF and PACF tail off. Differencing the series one time yields autocorrelation = 0 for lag 0, so it fully removes short term autocorrelation. It makes no sense to difference the series further.

```
ud0 = UnempRate
ud1 = diff(ud0)
# I will plot 10 years
acf2(ud0, 120)
```



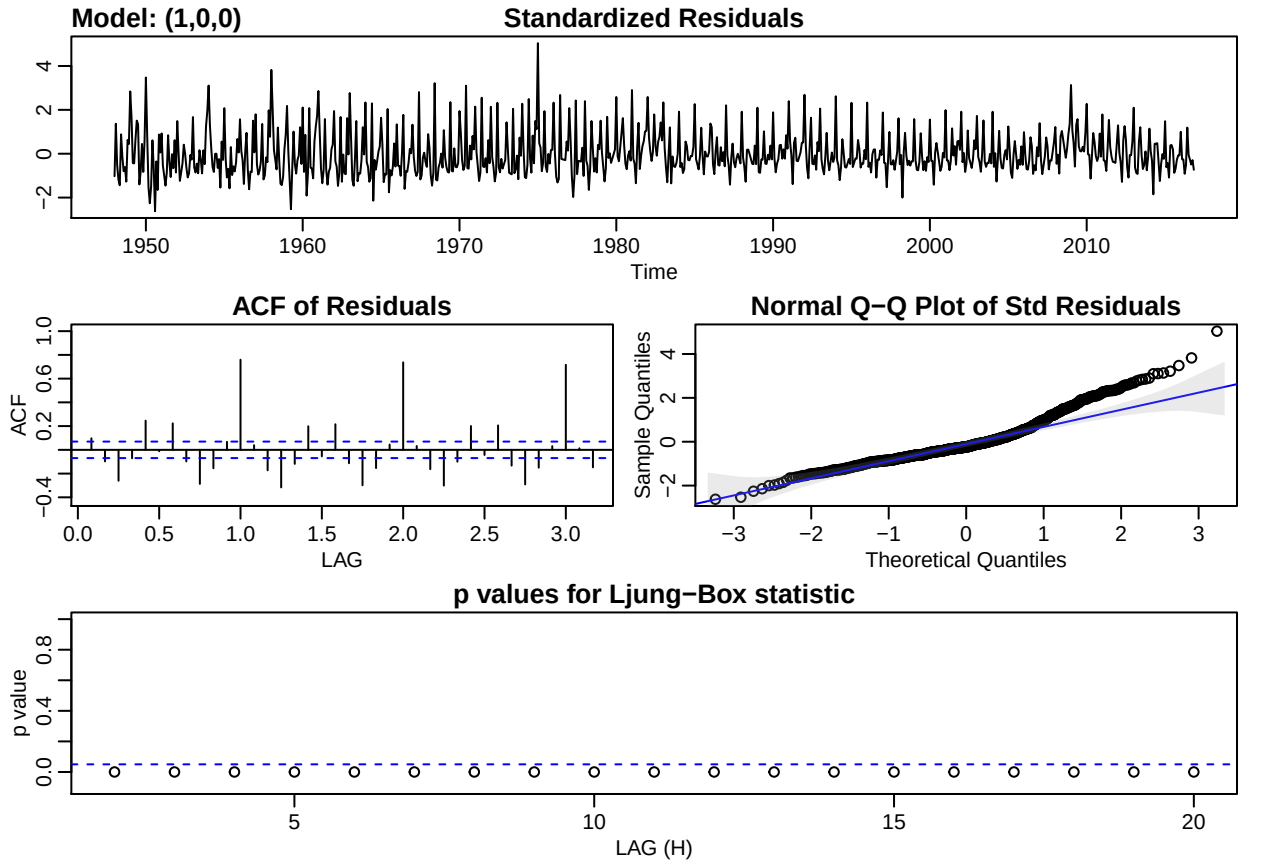

```
acf2(ud1, 120)
```


Series: ud1
ACF
0.6
0.2
−0.2
0
2
4
6
8
10
LAG
PACF
0.6
0.2
−0.2
0
2
4
6
8
10
LAG

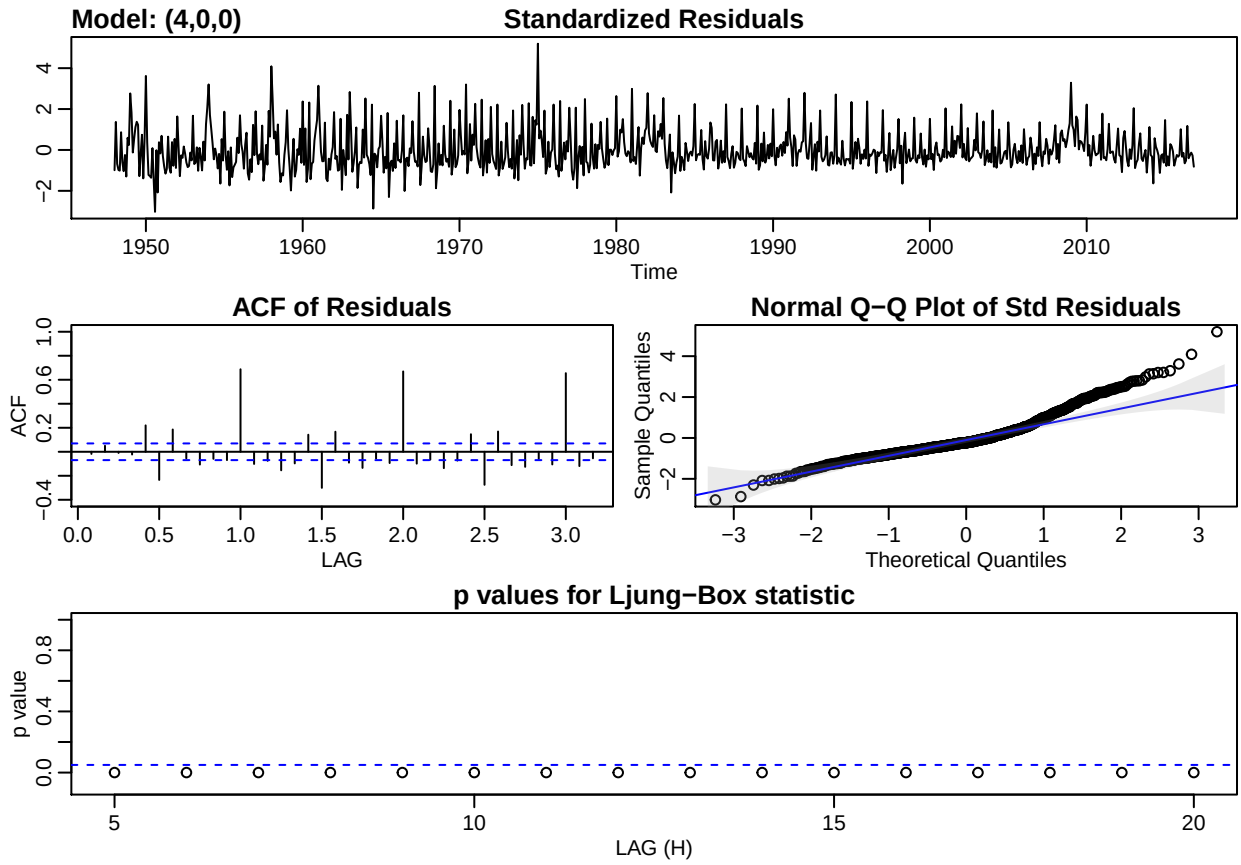
Following what we learned in class, I try fitting AR(1) and AR(4) models based on the first plot pair above, since ACF tails off and PACF partially cuts off at lag 1 and 4 (maybe). The cut offs are not pronouced so it is no surprise these and other AR(p) models tried did not work. Analysis of the residual plots of all AR(p) models showed a very poor fit.

The second plot pair invites attempts at AR models for differenced data, using ARIMA(12, 1, 0) and ARIMA(12, 1, 0), for example. I tried those and they also didn't fit well. Testing different values of q, such as ARIMA(1,0,12) and ARIMA(1, 0, 24) did not fit well to the data either.

```
sarima(ud0, 1, 0, 0)
```




```
sarima(ud0, 4, 0, 0)
```


Model: (4,0,0)
Standardized Residuals
Time
ACF of Residuals
ACF
LAG
Normal Q−Q Plot of Std Residuals
Sample Quantiles
Theoretical Quantiles
p values for Ljung−Box statistic
p value
LAG (H)

Further reading the Shumway & Stoffer book I came upon SARMA and SARIMA models. These models seem to be good candidates for unemployment rate modeling. During exploratory data analysis I did note that trends in the data were recurrent and had a 5 to 10 years duration cycle. Additionally, we note that the ACF of the differenced series shows autocorrelation at yearly intervals. The time series therefore does display long term seasonality.

In our class we did not cover analysis of ACF/PACF plots for the modeling of time series with SARMA and SARIMA. So, at this point I resort to the auto.arima function in the forecast package to identify the SARIMA model that best fits the data. The model found by auto.arima is ARIMA(3,0,1)(2,1,2)[12].

```
set.seed(123)
model = auto.arima(ud0)
summary(model)
```

```
## Series: ud0
## ARIMA(3,0,1)(2,1,2)[12]
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1      ar2      ar3      ma1    sar1    sar2    sma1     sma2
##       1.6804  -0.5685  -0.1234  -0.6064  -0.776  0.0479  0.0277  -0.6174
## s.e.  0.0678   0.1031   0.0425   0.0694     NaN  0.0498     NaN   0.0323
##
## sigma^2 estimated as 0.05456:  log likelihood=26.09
## AIC=-34.19   AICc=-33.96   BIC=8.14
##
## Training set error measures:
##                       ME      RMSE       MAE           MPE     MAPE
## Training set 0.003489937 0.2307453 0.1753223 -0.0009580578 3.287863
##                   MASE        ACF1
## Training set 0.199536 0.003265617
```

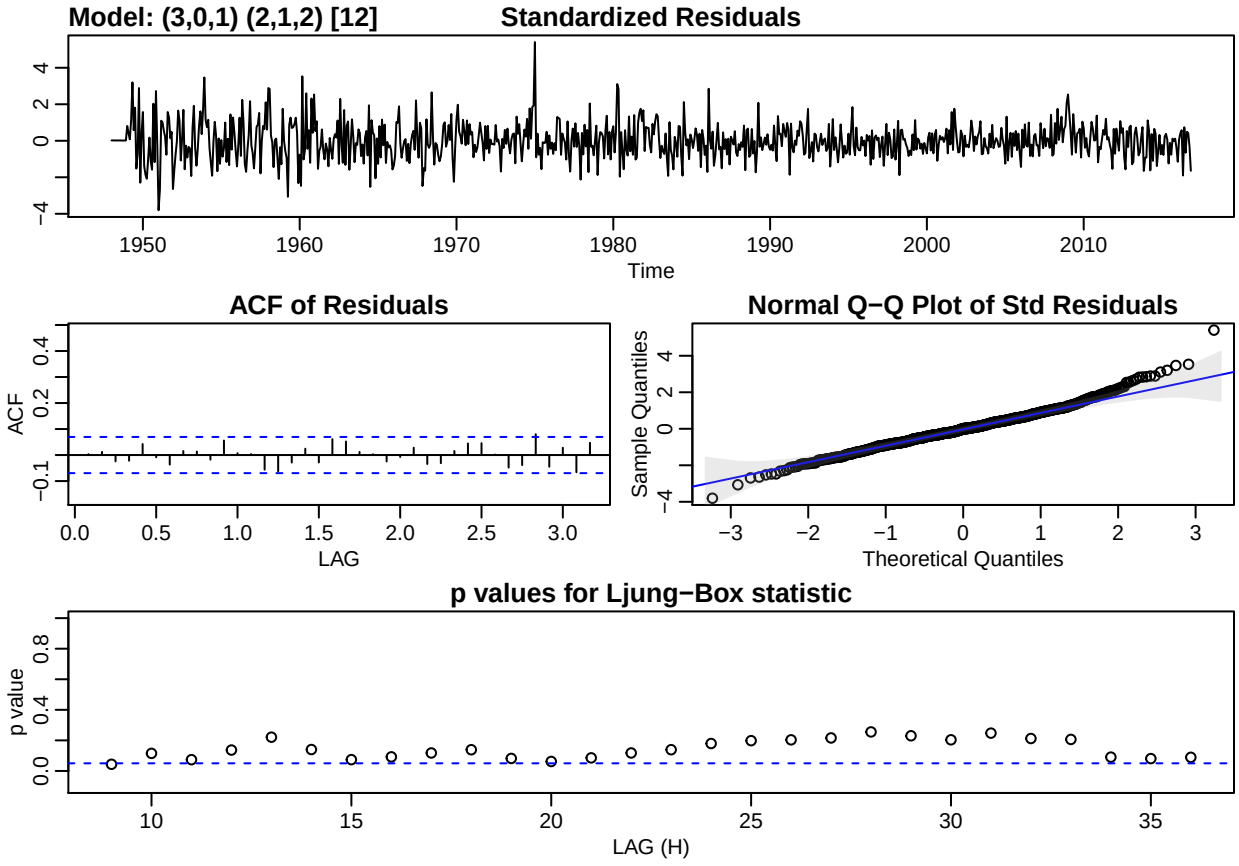
Model diagnostic - ARIMA(3,0,1)(2,1,2)[12]

Residual plots for the model selected by auto.arima are included here. The plot of the standardized residuals is similar to white noise. The ACF plot of residuals shows autocorrelation within the 5% band (no evidence to reject the null that $\rho_k = 0$ at $\alpha = 5\%$). The Q-Q plot approximates a straight line. Finally, the p-values for the Ljung-Box statistic shows values greater than 0.05, confirming residuals are normally distributed and not correlated (we cannot reject the null).

In summary, the model is a good fit.

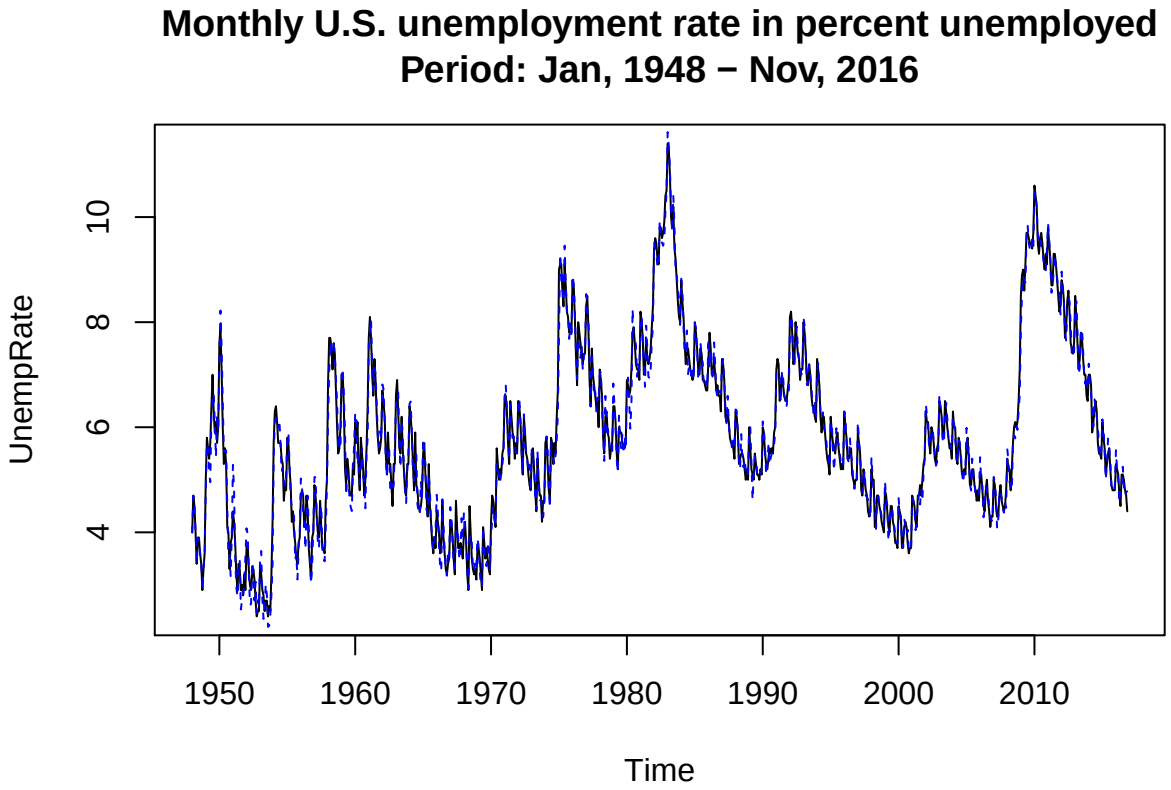
```
sarima(ud0, 3, 0, 1, 2, 1, 2, 12)
```

```
## Warning in stats::arima(xdata, order = c(p, d, q), seasonal = list(order =
## c(P, : possible convergence problem: optim gave code = 1
```



Below I plot the original timeseries in black and fitted values from the model in blue, dashed line. Clearly, the fit of the model is very close to observed data.

```
plot.ts(UnempRate, main="Monthly U.S. unemployment rate in percent unemployed\nPeriod: Jan, 1948 - Nov,
lines(model$fitted, col='blue', lty=2)
```

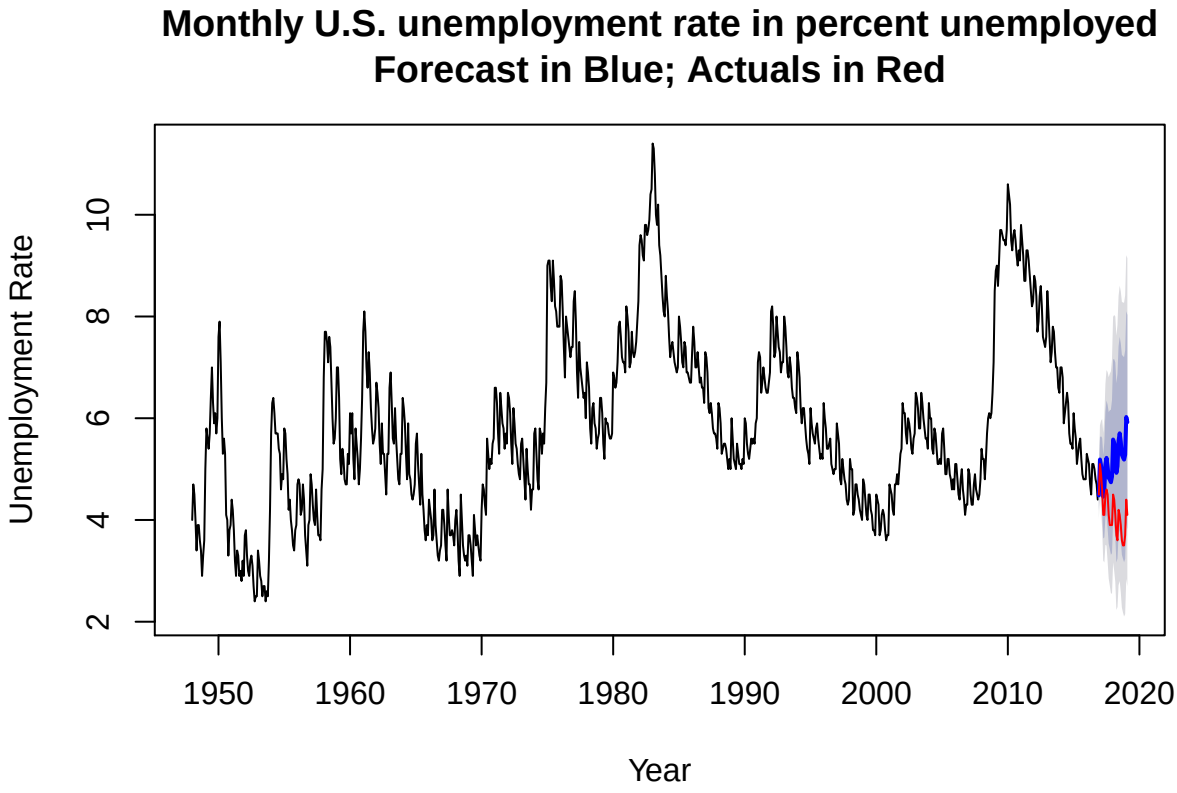


But did the model overfit?

To answer this question I downloaded additional unemployment data from website https://data.bls.gov/timeseries/LNU04000000/, which is the same source of data for the astsa package UnempRage dataset. The data downloaded is for the range Dec 2016 through Feb 2019.

On the next page I read the data, perform forecast using the selected SARIMA model and plot forecast against downloaded data. Forecast from the selected model appear in blue, actual values appear in red. The forecast is far from accurate, however, actual values are within the 80% probability band of the forecast.

```r
# ud0.future has 25 months of data, from Dec 2016 to Feb 2019
setwd("~/_pgms/IU_MSDS/Time_Series/CodePortfolio")
ud0.future = ts(read.csv('UnempRate_Dec2016_Feb2019.csv', header = T), start=c(2016,12), end=c(2019,2),
predictions = model %>% forecast(h = 27)
plot(predictions, main='Monthly U.S. unemployment rate in percent unemployed\nForecast in Blue; Actuals
lines(ud0.future, col='red')
```



Conclusion

In this project I analyzed US monthly unemployment rates from Jan 1948 through Nov 2016. The time series of unemployment rate shows stochatic trends, with short and long term seasonality. Variance in the data did not justify data transformation. I analyzed ACF/PACF plots on the original series and differenced series, in an attemp to identify AR(p), MA(q), ARMA(p,q) and ARIMA(p,d,q) models based on materials taught in class. No ARMA/ARIMA model fit the data well, so I used auto.arima to identify a SARIMA model that would work for the series. I discovered that the model ARIMA(3,0,1)(2,1,2)[12] provided a good fit for the data. The model was tested against values of unemployment for the period from Dec 2016 through Feb 2019. It generated a forecast with an 80% confidence interval band that contained the actual data, demonstrating the selected model works reasonably well for this series.