Krysten Thompson w271: Homework 6

Professor Jeffrey Yau

In this homework, you are asked to conduct Time Series EDA, develop a Time Trend Model, conduct model diagnostic analysis, and use the model to make forecasts.

Load the file **"bls_unemployment.csv"** This file contains the monthly unemployment rate in the United States from January 2007 to January 2017. *Note that these data are NOT seasonally adjusted.*

1. Load the csv file into a data.frame, calling it *df*, and examine the structure of the "raw" series after you load it into a data.frame.

```
library(astsa)
library(ggplot2)
library(knitr)
library(timeSeries)
```

```
## Loading required package: timeDate
```

```
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
df <- read.csv('bls_unemployment.csv')
head(df)
```

```
##     Series.id Year Period Value
## 1 LNU04000000 2007    M01   5.0
## 2 LNU04000000 2007    M02   4.9
## 3 LNU04000000 2007    M03   4.5
## 4 LNU04000000 2007    M04   4.3
## 5 LNU04000000 2007    M05   4.3
## 6 LNU04000000 2007    M06   4.7
```

```
str(df)
```

```
## 'data.frame':    121 obs. of  4 variables:
##  $ Series.id: Factor w/ 1 level "LNU04000000": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Year     : int  2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  $ Period   : Factor w/ 12 levels "M01","M02","M03",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Value    : num  5 4.9 4.5 4.3 4.3 4.7 4.9 4.6 4.5 4.4 ...
```

2. Convert it into a R time-series object, and examine the structure of the series after you convert the *df* into a time series object. What is the difference between the two structures?

```
class(df)
```

```
## [1] "data.frame"
```

```
df <- ts(df, start = c(2007, 1), freq = 12)
class(df)
```

```
## [1] "mts"    "ts"     "matrix"
```

```r
str(df)
```

```
## Time-Series [1:121, 1:4] from 2007 to 2017: 1 1 1 1 1 1 1 1 1 1 ...
## - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:4] "Series.id" "Year" "Period" "Value"
```

```r
head(df)
```

```
##      Series.id Year Period Value
## [1,]         1 2007      1   5.0
## [2,]         1 2007      2   4.9
## [3,]         1 2007      3   4.5
## [4,]         1 2007      4   4.3
## [5,]         1 2007      5   4.3
## [6,]         1 2007      6   4.7
```

```r
#describe(df)

#because I changed 'df' to ts, the describe function won't run
```

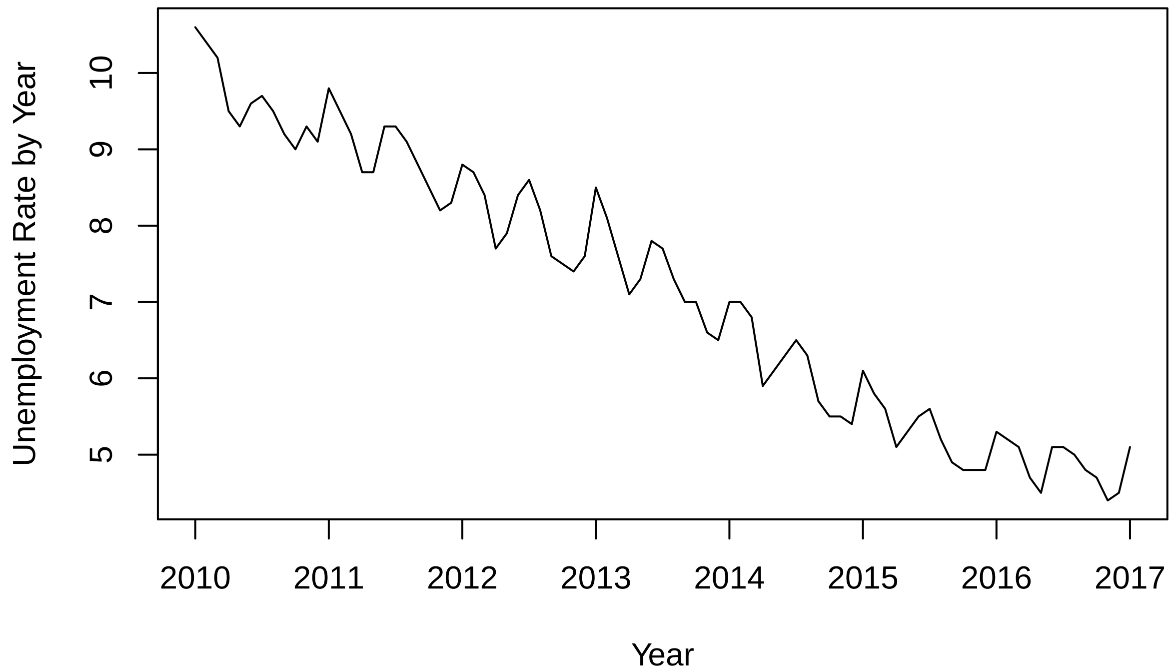
Suppose we want to approximate the unemployment rate between 2010 and 2017 using a linear time trend model. For this exercise, feel free to modify this dataset but be sure that you explain what you did and why.

I created a subset of the dataset, specifically data for 2010 through 2017. I created a new variable 'df.2010_2017'.

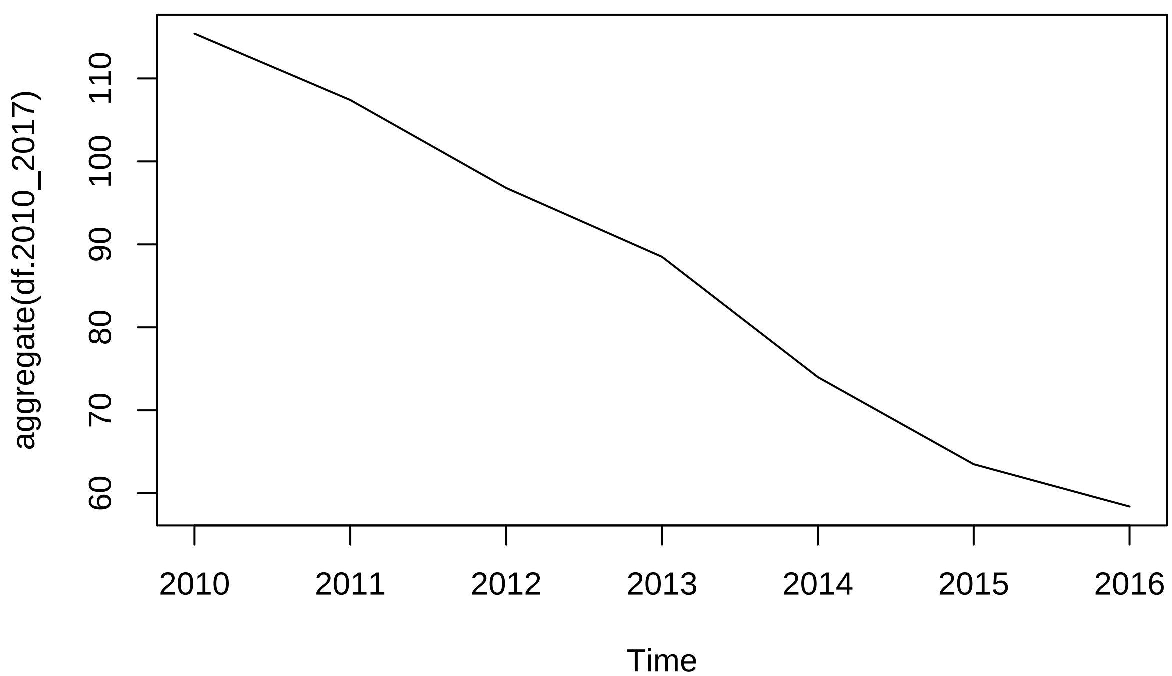
First, I plotted the variable with no modification. But this showed the in-year trend (e.g. monthly data).

Then I plotted the data and called 'aggregate' so that the plot would reflect a smooth linear time trend line.

```r
df.2010_2017 <- window(df[, 4], start = c(2010, 1), end = c(2017, 1))
plot(df.2010_2017, ylab = 'Unemployment Rate by Year', xlab = 'Year')
```




```r
plot(aggregate(df.2010_2017))
```

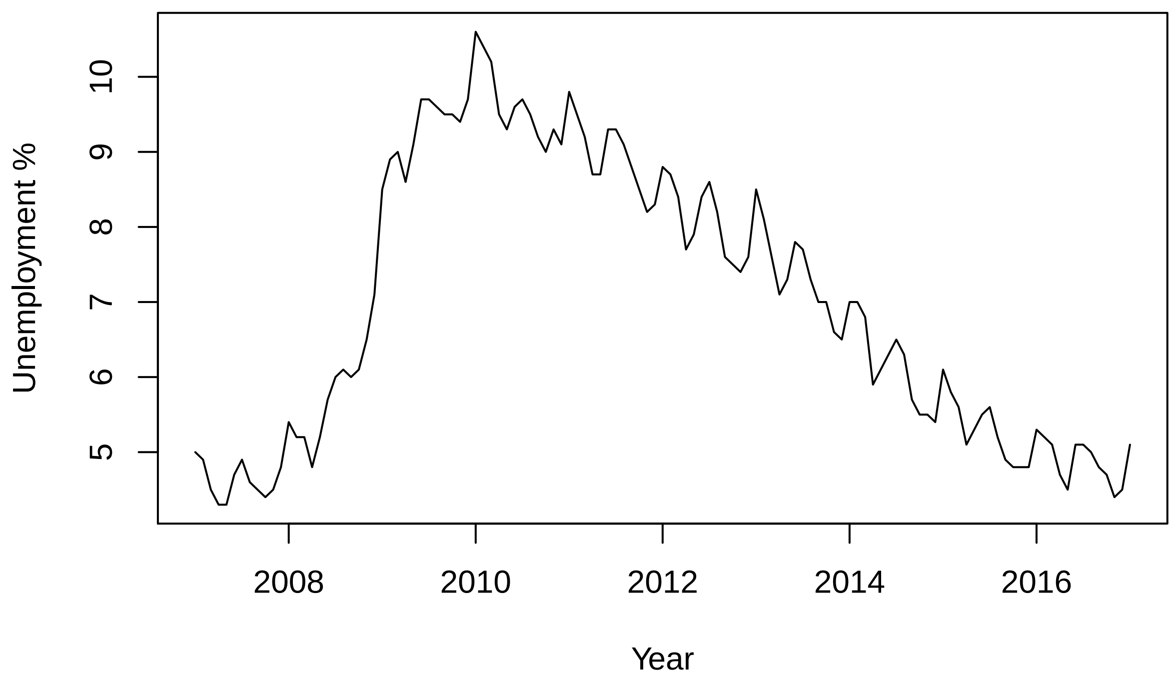


3. Conduct EDA on the series.

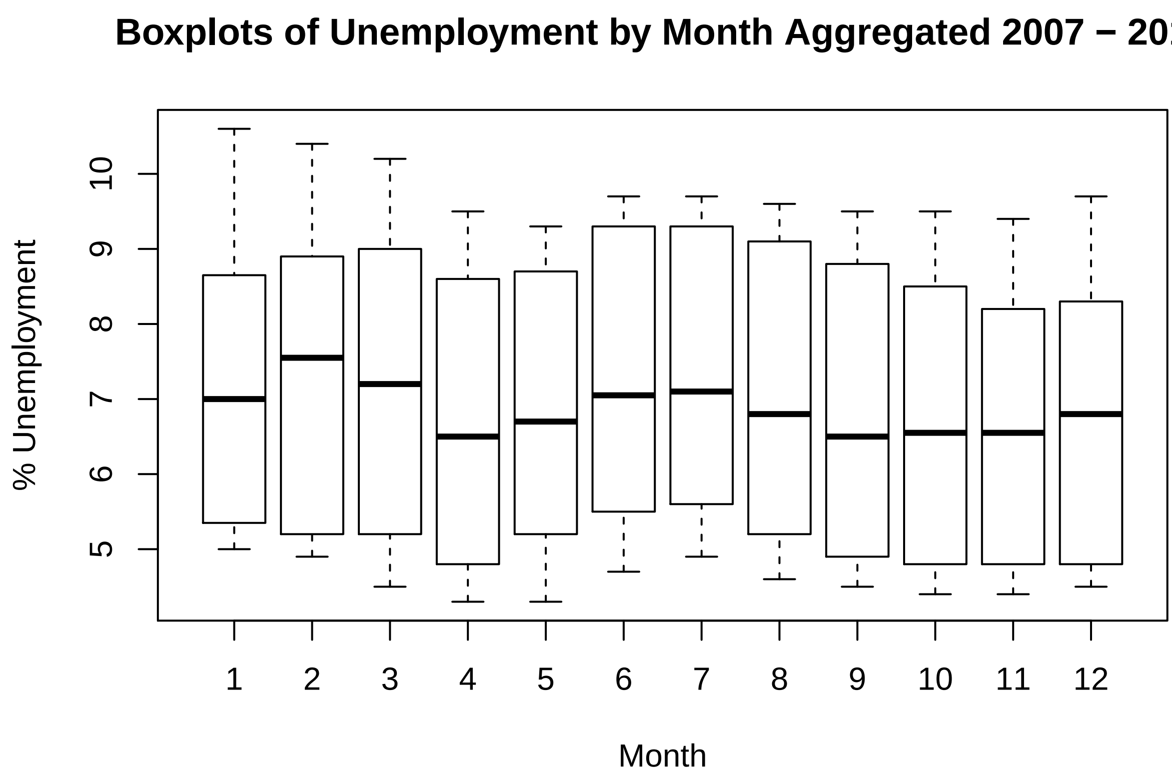
```r
start(df); end(df); frequency(df)
```

```
## [1] 2007    1

## [1] 2017    1

## [1] 12
```

```r
plot(df[,4], ylab = 'Unemployment %', xlab = 'Year')
```




```r
boxplot(df[,4] ~ cycle(df), ylab = '% Unemployment', xlab = 'Month', main = 'Boxplots of Unempl
```




```r
sum(is.na(df))
```

```
## [1] 0
```

4. Create a linear time trend model by regressing the unemployment rate on time. Interpret the model results.

```
mod.lin <- lm(df[,4] ~ Year, data=df)
summary(mod.lin)
```

```
##
## Call:
## lm(formula = df[, 4] ~ Year, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3778 -1.4251  0.1915  1.6860  3.3805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 314.25511  114.78648   2.738  0.00714 **
## Year         -0.15275    0.05706  -2.677  0.00848 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.822 on 119 degrees of freedom
## Multiple R-squared:  0.0568, Adjusted R-squared:  0.04887
## F-statistic: 7.166 on 1 and 119 DF,  p-value: 0.008479
```

Coefficient for Year (-0.15) **suggests that for every year after 2007, unemployment decreases by** ~15% **per year.**

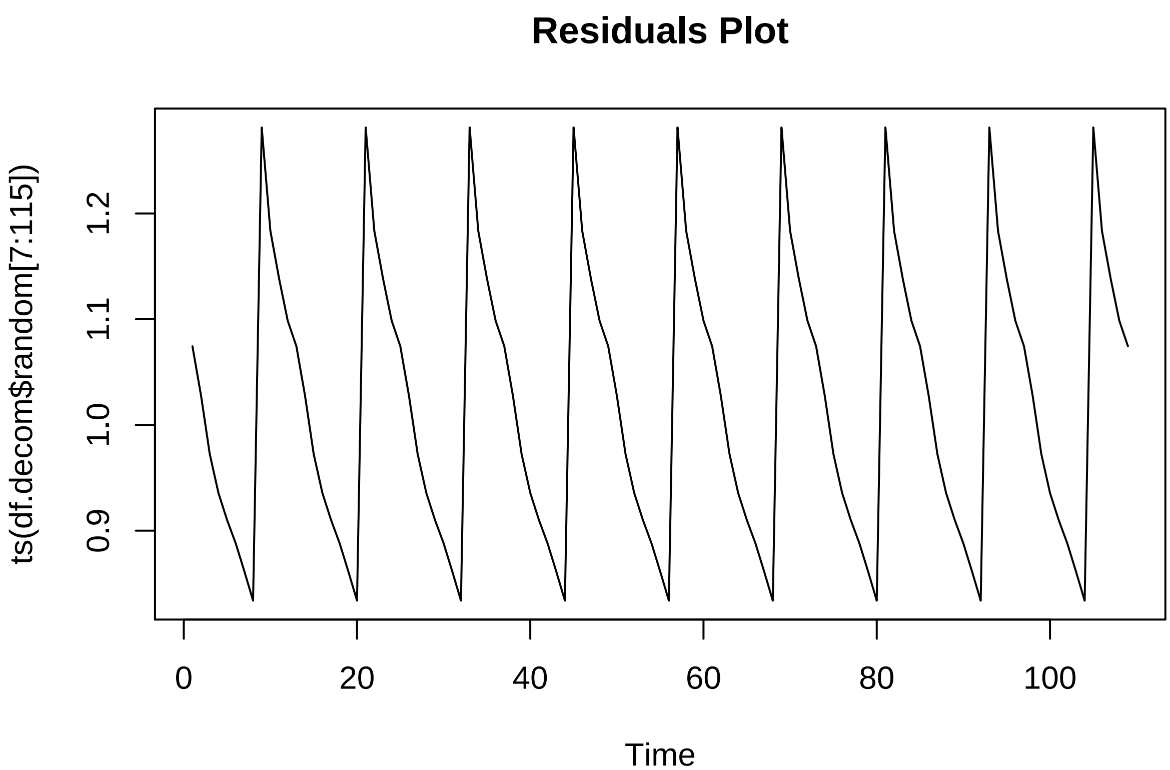
The intercept value is high likely due to the Great Recession starting in late 2007 and into 2008. The significant Standard Error of 115 **is a result of 2007 being low and then the unemployment rate spiking in 2008 and extending into 2010.**

While the p-value of 0.0085 **is** $<0.05\%$**, it's important to note the Multiple** R^2 **value of** 0.0568**. Approximately** 6% **of the variance in unemployment rate is due to time. This is because there are many confounding variables not included in this model that better explain the unemployment rate.**

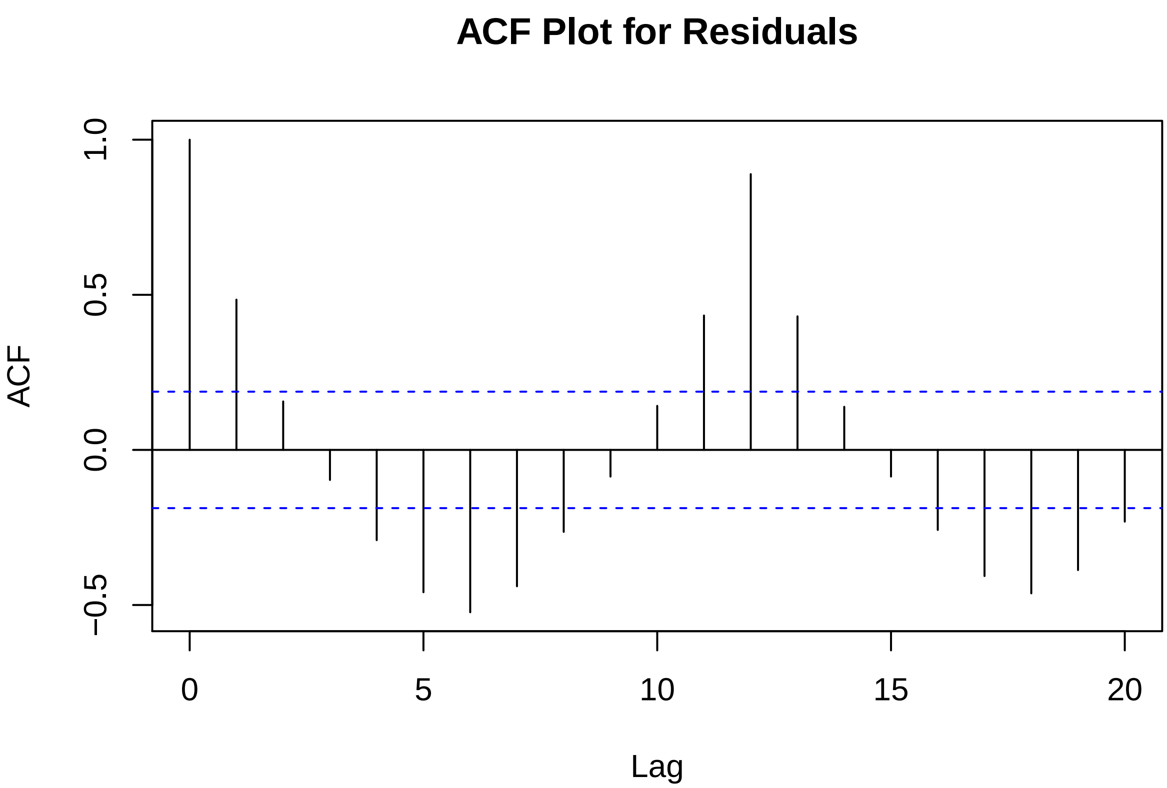
5. Examine the residuals of this model as you would if it were a classical linear model. In addition, generate ACF and PACF plots of the residuals.

Because I discussed the residuals of summary(model) above, I created a residuals plot below using decompose and plotting "random".

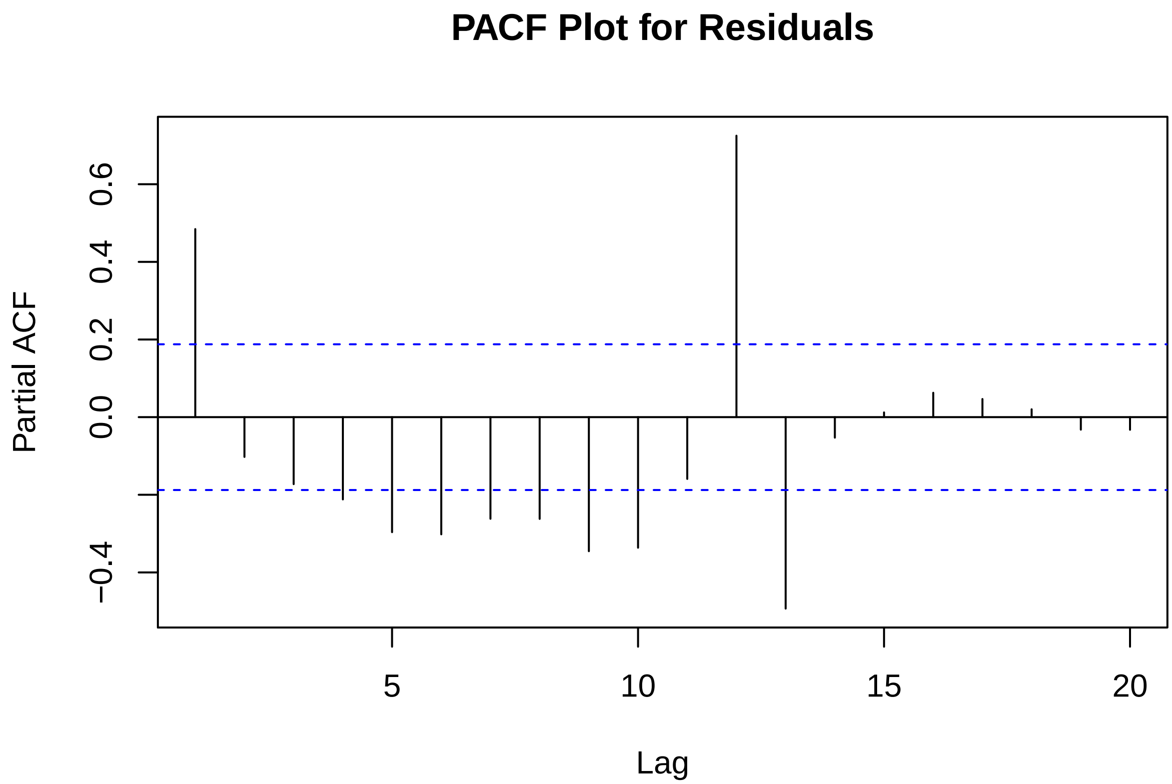
```
df.decom <- decompose(df, "multiplicative")
plot(ts(df.decom$random[7:115]), main= "Residuals Plot")
```




```r
acf(df.decom$random[7:115], main="ACF Plot for Residuals")
```

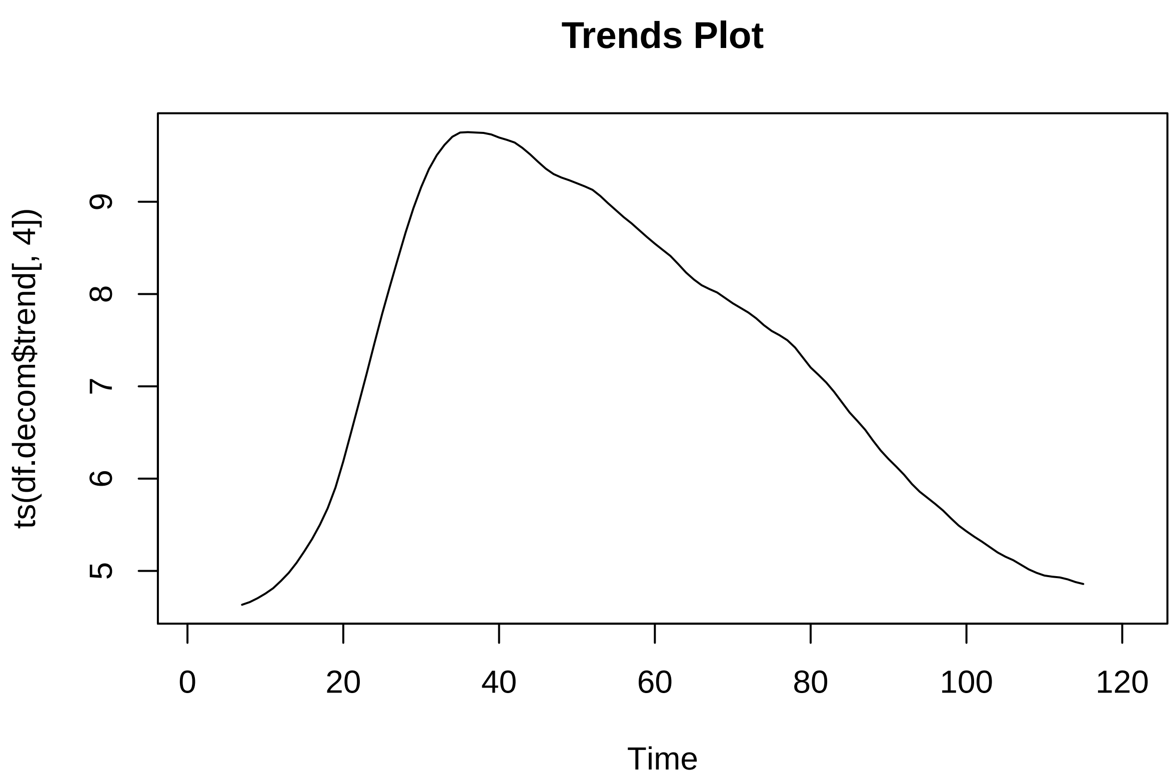



```r
pacf(df.decom$random[7:115], main = "PACF Plot for Residuals")
```

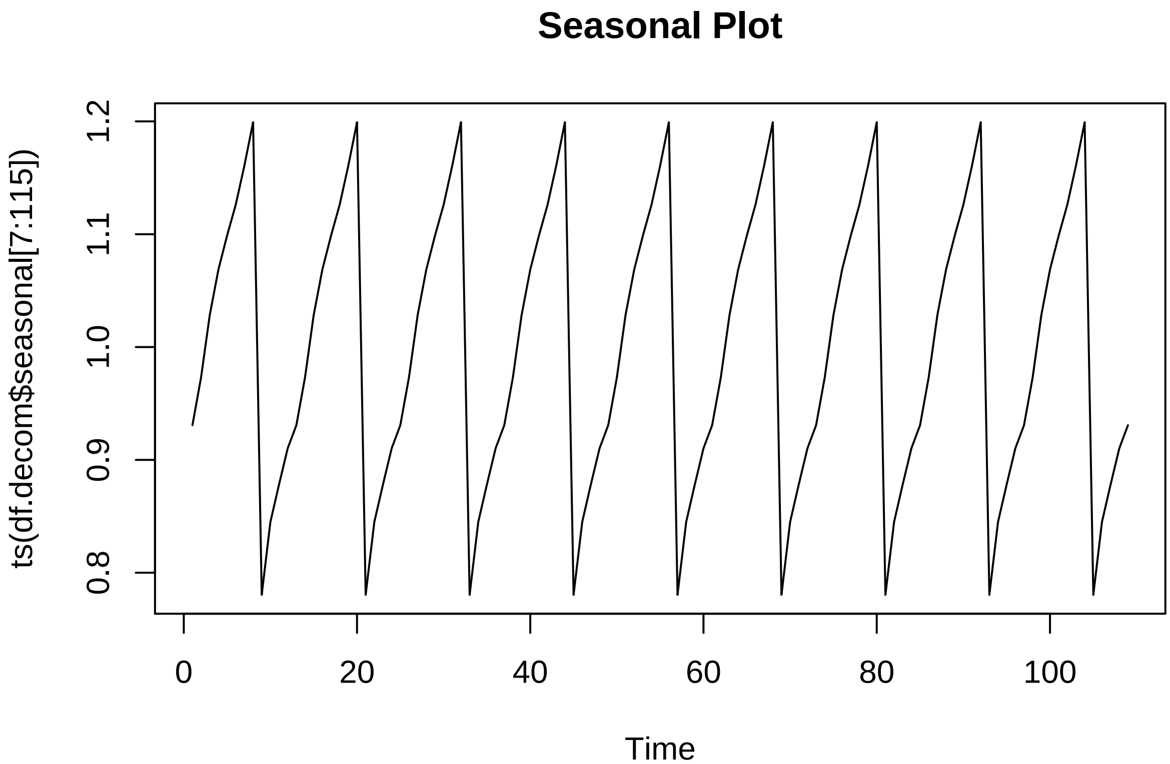


I wanted to play around and see what happened when I plotted "Trends" and "Seasonal".

```
plot(ts(df.decom$trend[,4]), main= "Trends Plot")
```



```r
plot(ts(df.decom$seasonal[7:115]), main= "Seasonal Plot")
```



6. Use this model to predict the unemployment rate in 2018 January (i.e. 12 months from the end of the sample). Do the result make sene? How about a prediction of the unemployment rate in 2020 December?

Gerard, I couldn't figure this out after extensive online searching. Gave it a try using 'auto.arima' but couldn't get it to print for Jan 2018.

```r
#
library(forecast)
```

```
##
## Attaching package: 'forecast'

## The following object is masked from 'package:astsa':
##
##     gas
```

```r
#
# frcst_Jan <- forecast(mod.lin, n.ahead=12)
# frcst_Jan
```

```r
value <- data.frame(df[, 4])
.months <- ts(value, frequency=12)
fit <- auto.arima(.months)
pred <- predict(fit, n.ahead=14)
pred
```

```
## $pred
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 11          4.927169 4.715364 4.178803 4.279346 4.707262 4.792005 4.557602
## 12 4.806632 4.653993 4.442189
##           Sep      Oct      Nov      Dec
## 11 4.245227 4.138223 4.049022 4.127068
## 12
##
## $se
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 11           0.2100037 0.3304441 0.4174746 0.4892614 0.5517866 0.6079147
## 12 0.8718141 0.9199513 0.9682661
##            Aug       Sep       Oct       Nov       Dec
## 11 0.6592814 0.7069255 0.7515553 0.7936795 0.8336779
## 12
```