W271 Live Session 9: ARIMA and SARIMA

Jeffrey Yau, Devesh Tiwari

3/7/2017

Main topics covered in Week 9

- Mixed Autoregressive Moving Average (ARMA) Models
  - Mathematical formulation and derivation of key properties
  - Comparing ARMA models and AR models using simulated series
  - Comparing ARMA models and AR models using an example
- An introduction to non-stationary time series model
- Random walk and integrated processes
- Autoregressive Integrated Moving Average (ARIMA) Models
  - Review the steps to build ARIMA time series model
  - Simulation
  - Modeling with simulated data using the Box-Jenkins approach
  - Estimation, model diagnostics, model identification, model selection, assumption testing, and statistical inference / forecasting, backtesting
- Seasonal ARIMA (SARIMA) Models
  - Mathematical formulation
  - An empirical example
- Putting everything together: ARIMA modeling

Readings

CM2009: Paul S.P. Cowpertwait and Andrew V. Metcalfe. *Introductory Time Series with R.* Springer. 2009.

- Ch. 4.3 – 4.7, 6, 7.1 – 7.3

SS2016: Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Applications.* EZ Edition with R Examples:

- 3.7 – 3.10, review 3.1 – 3.6

HA: Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and Practice:

- Ch. 8.5 – 8.9

Agenda for this week's live session:

1. Recap (15 mins)
2. Review questions (15 mins)
3. Breakout Session 1: EDA (10 minutes breakout; 10 minutes discussion)
4. Group discussion: Non-seasonal modeling (15 minutes)
5. Breakout Session 2: Seasonal Modeling (time remaining)

Recap and overview

1. Last week, we were introduced to autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) models. These models are only appriorate for time-series that are weakly stationary (stationary in the mean and the variance).
2. We often are confronted with time-series that is not stationary in the mean and variance. Luckily for us, we can transform these series in order to make them stationary!
3. Here is an incomplete list of how/why time-series might not be stationary in the mean:
    a. Data has a trend
    b. Data contains a unit-root (next week)
    c. Data contains seasonal elements
4. We can take care of these problems either by detrending the data or by differencing the data. Once the data are transformed into a weakly stationary series, we can model the resulting series with an ARMA model. We call these models ARIMA models if the data do not exhibit any seasonality. If the data are seasonal, then these models are called SARIMA models.
5. Remember, here are the steps to building an ARIMA model!
    i. Conduct and EDA to determine if you need to transform the data in order to make it stationary.
    ii. Transform the data if needed.
    iii. Estimate several Arima(p,d,q) models. Remember, you set the value of d in the first step! So really, you are trying to find the apprioriate values of p and q.
    iv. Evaluate the residuals of models with the lowest AIC/BIC values and simpler models. Select the model where the residuals resemble white nose.
    v. If you still have some candidate models remaining, then conduct an out of sample test and select the model with the lowest forecasting error.
    vi. Answer your question / generate forecasts!

Breakout Session 1: Review Questions

1. Consider the pure Moving Average process below, where ω_t is a white noise process:

$$x_t = \omega_t + \beta_1 \omega_{t-1}$$

Under what circumstances is this process stationary in the mean?

(a) Always

(b) Never, this is not an AR(p) process.

(c) When the absolute value of β_1 is less than 1.

(d) When the absolute value of the roots of the characteristic polynomial is greater thane 1.

2. Consider a time series represented by the following equation:

$$(1 - \alpha_1 \mathbf{B} - \alpha_2 \mathbf{B}^2)(1 - \mathbf{B}) x_t = \omega_t (1 + \beta_1 \mathbf{B})$$

(a) Describe this time series using the ARIMA(p,d,q) notation.

(b) Can you tell if this time series is stationary? If so, is it? If you cannot determine whether it is stationary, what additional information do you need?

Breakout Session 2: EDA and data transformation

In this live session, we are going to build a SARIMA model on the relative search activity for the phrase, "flight prices." These data are provided by google correlate and they are weekly. For the sake of simplicity, we will focus on 2010 onward.

Remember that we can express a SARIMA model as: SARIMA(p,d,q)x(P,D,Q),m.

```
# Insert the function to *tidy up* the code when they are printed out
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

```
library(forecast)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

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

```
## This is forecast 7.3
```

```
library(astsa)
```

```
##
## Attaching package: 'astsa'
```

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

```
path <- "~/Documents/Projects/MIDS/"
setwd(path)
df <- read.csv("Summer 2016/materials for ISVC/Lab 3/correlate-flight_prices.csv")

names(df)
```

```
##   [1] "Date"                      "flight.prices"
##   [3] "the.florida.keys"          "florida.beach.resorts"
##   [5] "planning.a.trip"           "resorts.florida"
##   [7] "exercise"                  "florida.beach.house"
##   [9] "key.florida"               "keys.florida"
##  [11] "hotels.in.florida"         "longboat.key"
##  [13] "weights"                   "cay"
##  [15] "family.resorts"            "south.beach.miami"
##  [17] "buying.a.house"            "shore.excursions"
##  [19] "island.florida"            "longboat"
##  [21] "destin.florida.rentals"    "florida.homes.for.sale"
##  [23] "key.west.florida"          "breakfast.key.west"
##  [25] "hotels.clearwater"         "suites.orlando"
##  [27] "vacation.in.florida"       "axles"
##  [29] "clearwater.beach.florida"  "st.petersburg.beach"
##  [31] "resort.packages"           "key.largo"
##  [33] "marathon.key"              "all.inclusive.aruba"
```

```
##  [35] "family.beach.vacations"      "cruise.prices"
##  [37] "plan.a"                      "marathon.florida"
##  [39] "bed.and.breakfast.key.west"  "universal.studios.vacation"
##  [41] "florida.resorts"             "diet"
##  [43] "lido.key"                    "beach.houses.for.rent"
##  [45] "florida.disney"              "stanford.summer"
##  [47] "us.virgin.islands"           "hotels.daytona"
##  [49] "pine.key"                    "florida.for.sale"
##  [51] "to.do.in.cancun"             "vacation.house.rentals"
##  [53] "florida.cheap"               "cruise.bahamas"
##  [55] "diet.snacks"                 "vacation.all.inclusive"
##  [57] "key.west.beaches"            "house.key"
##  [59] "homes.for.sale.florida"      "hotels.in.san.juan"
##  [61] "key.west"                    "beach.houses"
##  [63] "all.inclusive.caribbean"     "high.school.internships"
##  [65] "las.vegas.flights"           "siesta.keys"
##  [67] "inclusive.resort"            "sirata.beach"
##  [69] "camping.in.florida"          "passport.renewal"
##  [71] "low.carb.meals"              "florida.vacation.spots"
##  [73] "sirata.beach.resort"         "petersburg"
##  [75] "body.fat"                    "high.school.summer.programs"
##  [77] "wedding.sandals"             "living.in.florida"
##  [79] "carb.meals"                  "treasure.island.florida"
##  [81] "hawaii.packages"             "hawaii.flights"
##  [83] "rentals.florida.keys"        "hopper.pass"
##  [85] "summer.hockey.camps"         "camp.themes"
##  [87] "creatine"                    "all.inclusive.resort"
##  [89] "rosacea"                     "banyan"
##  [91] "big.pine.key"                "hawaiian.inn"
##  [93] "sugar.alcohol"               "bride.dresses"
##  [95] "rentals.orlando"             "college.summer.programs"
##  [97] "summer.high"                 "duck.key"
##  [99] "west.florida"                "bavaro"
## [101] "miami.beach.hotels"
```

```
# We can actually analyze the relative seach activity for
# many phrases!

df <- df[, 1:2]  #focus on the phrase flight prices
str(df)
```

```
## 'data.frame':    626 obs. of  2 variables:
##  $ Date         : Factor w/ 626 levels "1/1/06","1/1/12",..: 43 4 16 30 211 256 222 235 247 307 ...
##  $ flight.prices: num  5.811 2.901 1.575 0.047 2.8 ...
```

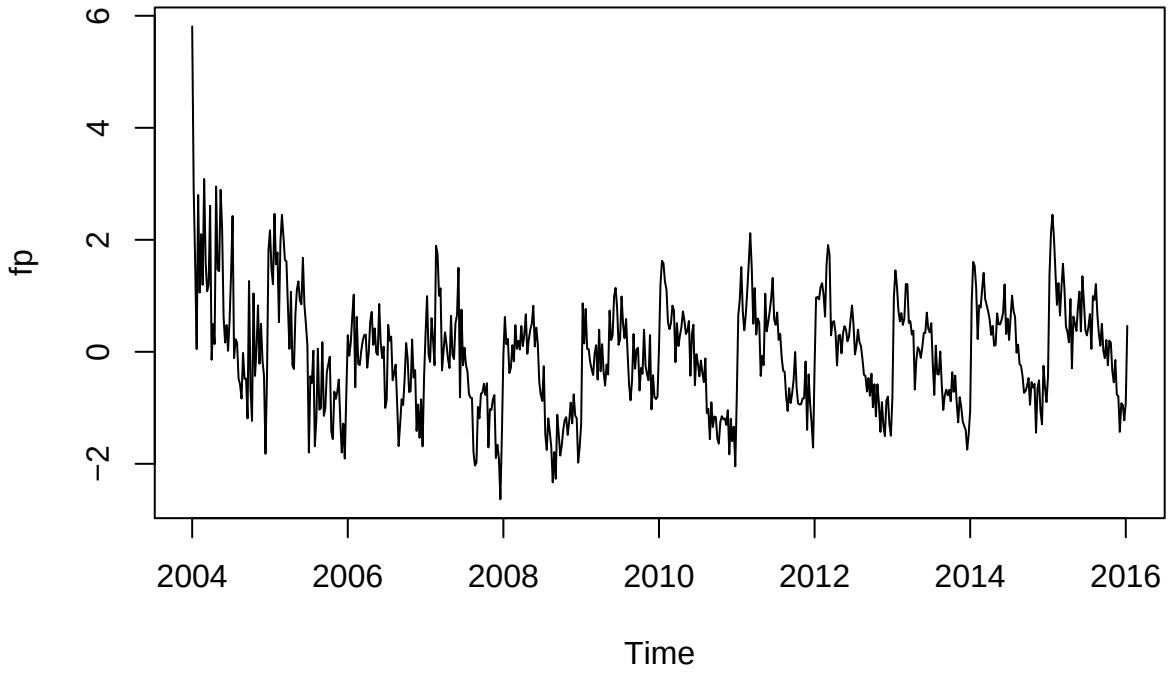
```
cbind(head(df), tail(df))
```

```
##      Date flight.prices     Date flight.prices
## 1  1/4/04         5.811 11/22/15        -1.430
## 2 1/11/04         2.901 11/29/15        -0.918
## 3 1/18/04         1.575  12/6/15        -0.952
## 4 1/25/04         0.047 12/13/15        -1.224
## 5  2/1/04         2.800 12/20/15        -0.897
## 6  2/8/04         1.058 12/27/15         0.462
```

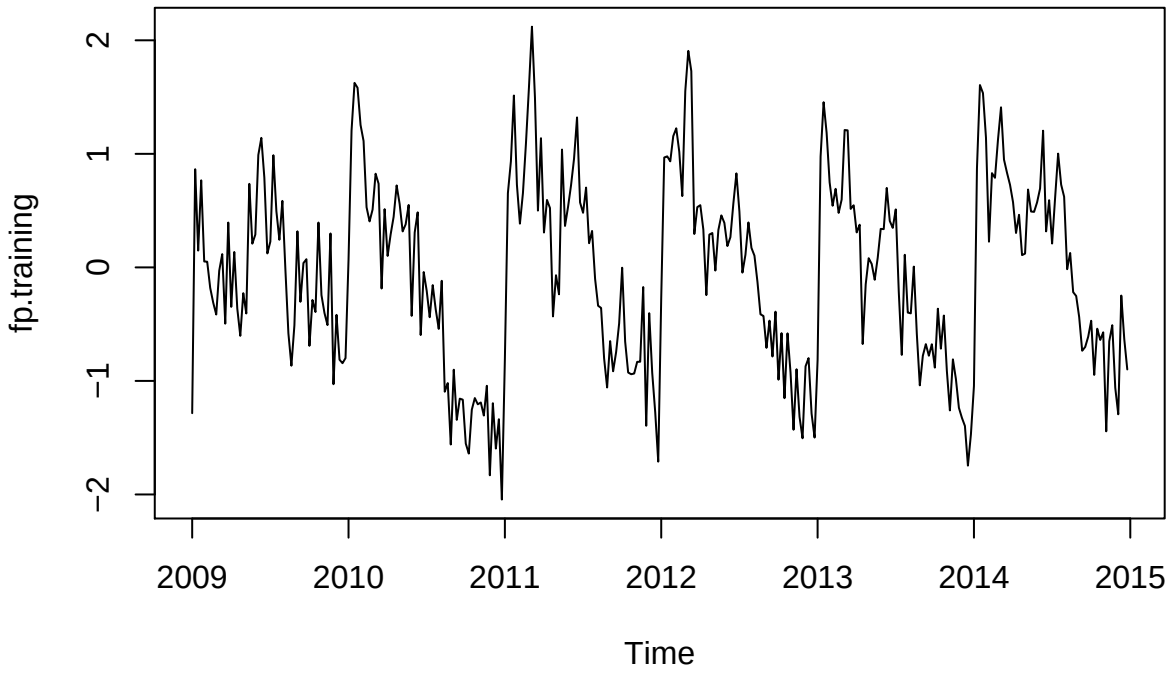
```r
summary(df)
```

```
##       Date     flight.prices
##  1/1/06 :  1   Min.   :-2.6330
##  1/1/12 :  1   1st Qu.:-0.7480
##  1/10/10:  1   Median : 0.0175
##  1/11/04:  1   Mean   :-0.0123
##  1/11/09:  1   3rd Qu.: 0.5727
##  1/11/15:  1   Max.   : 5.8110
##  (Other):620
```

```r
# Create a time-series object

fp <- ts(df$flight.prices, frequency = 52, start = c(2004, 1))
plot(fp)
```

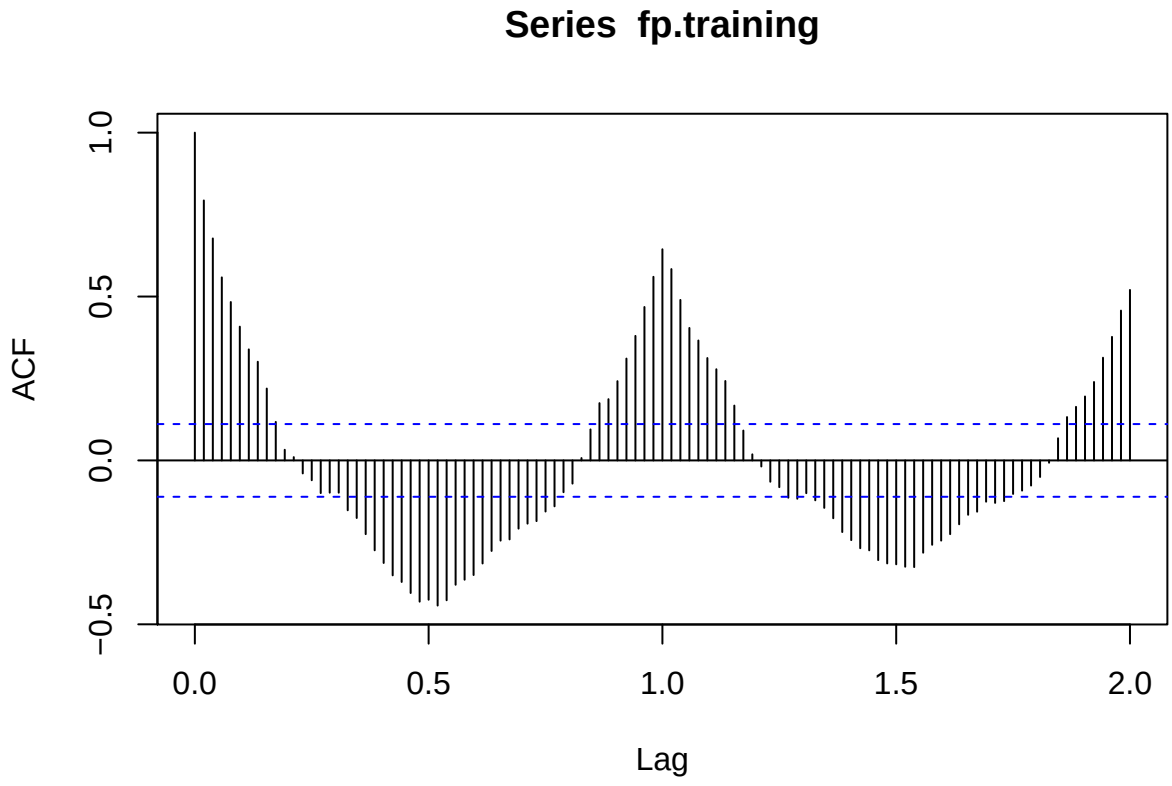



```r
# Lets keep data between 2009 and 2014. Let's hold out 2015
# as test data that you can use later.

fp.training <- fp[time(fp) >= 2009 & time(fp) < 2015]
fp.training <- ts(fp.training, frequency = 52, start = c(2009,
    1))

fp.test <- fp[time(fp) >= 2015]
fp.test <- ts(fp.test, frequency = 52, start = c(2015, 1))
plot(fp.training)
```

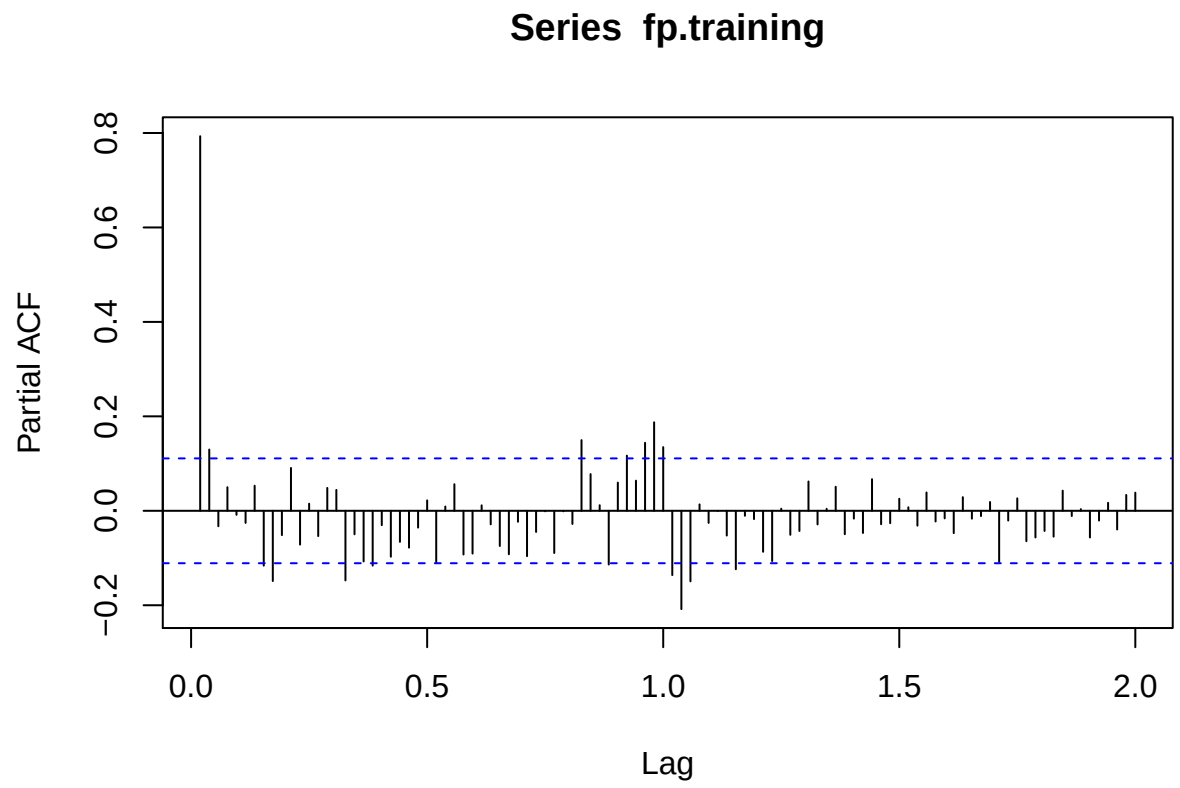


As we can see, this time-series is clearly not stationary in the mean! It is also pretty apparent that the time-series exhibits a lot of seasonality! Remember, at this stage we want to determine the values for d and D!

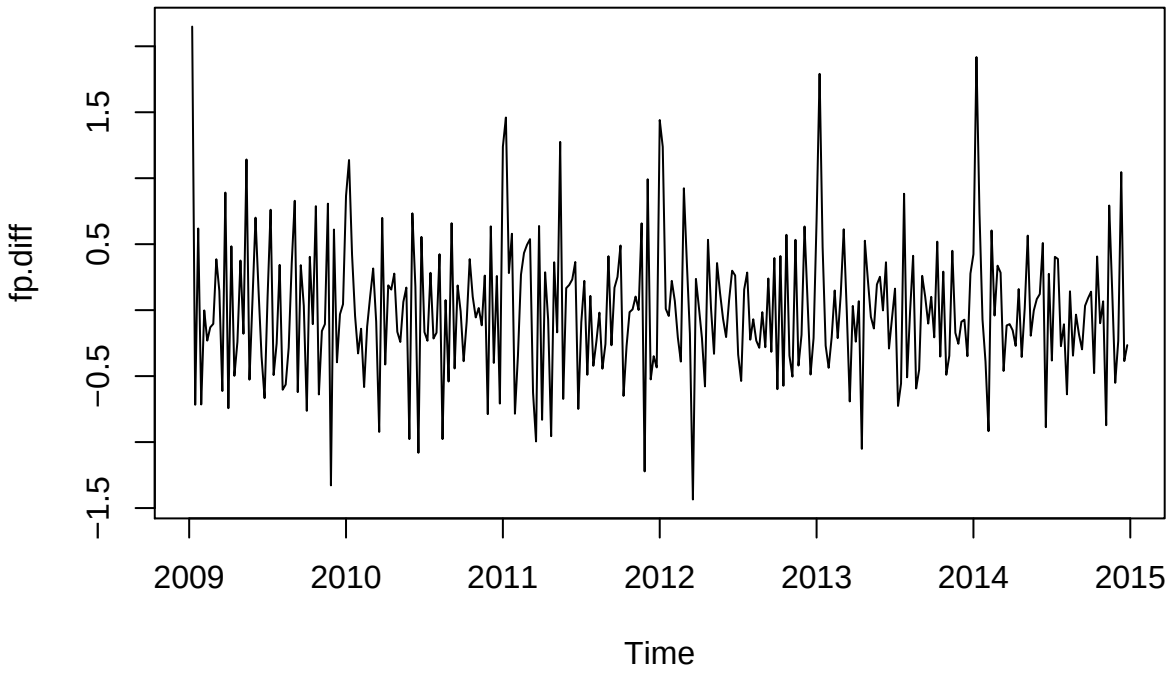
```
# Non-seasonal component

acf(fp.training, lag.max = 104)
```



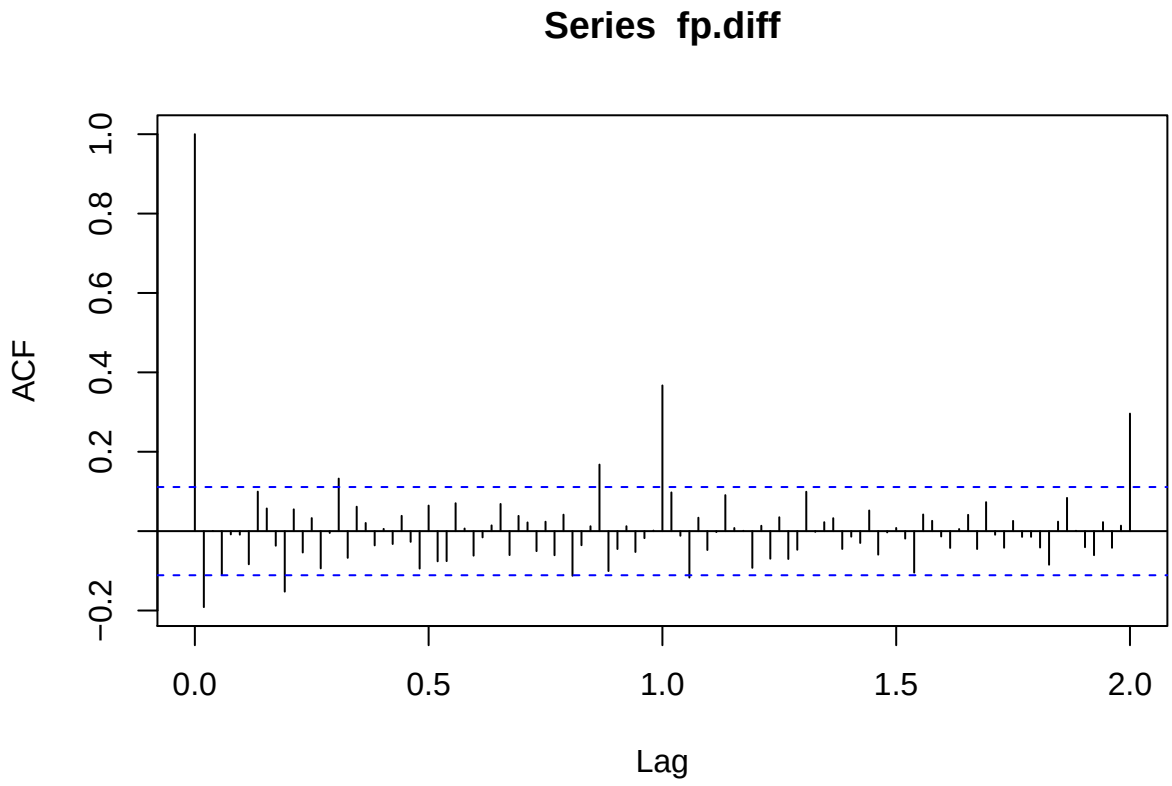
```r
pacf(fp.training, lag.max = 104)
```



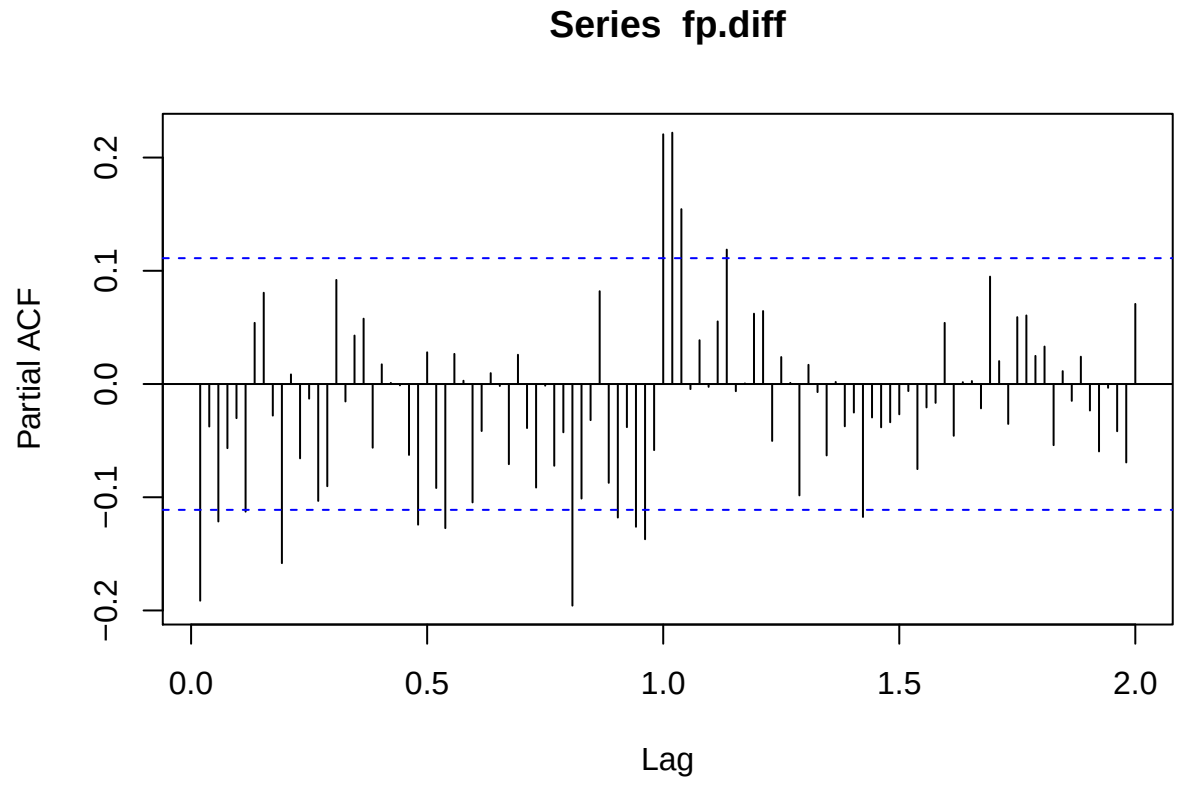

```r
fp.diff <- diff(fp.training, lag = 1)
plot(fp.diff)
```



```
acf(fp.diff, lag.max = 104)
```




```
pacf(fp.diff, lag.max = 104)
```



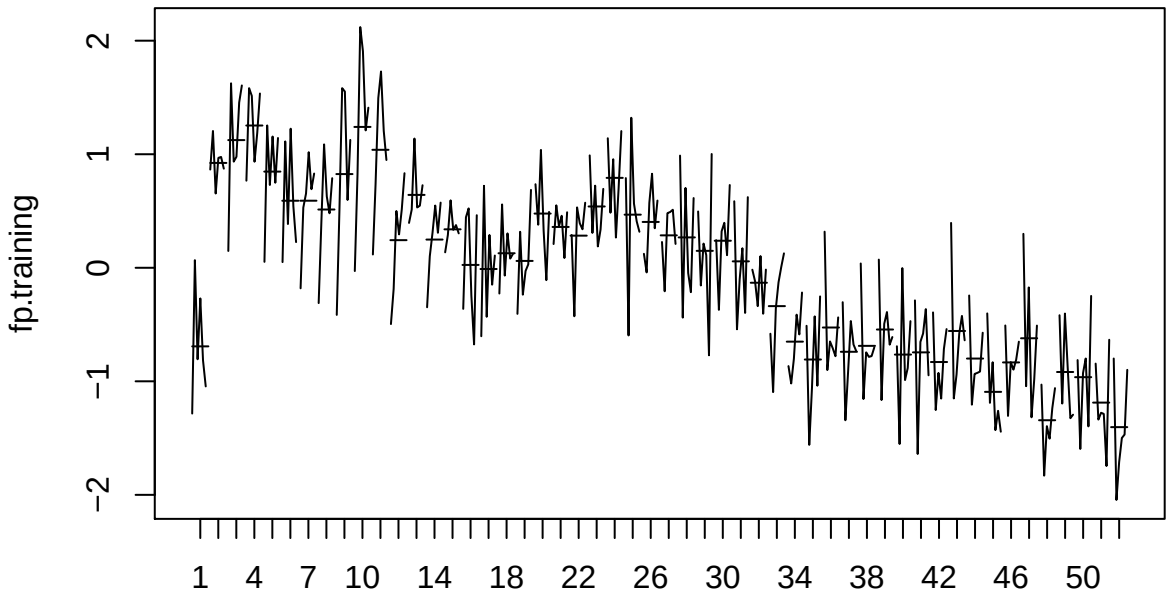
1. Based on these plots, do you think that the raw series, *fp.training* needs to be differenced? Why or why not?

2. Examine the ACF/PACF plots of the raw series. What type of model (AR, MA, ARMA) model do you think would be appropriate for this data? Focus only on the seasonal compnent!
3. Examine the ACF/PACF plots of the *differened* series. What type of model (AR, MA, ARMA) model do you think would be appropriate for this data? Focus only on the seasonal compnent!

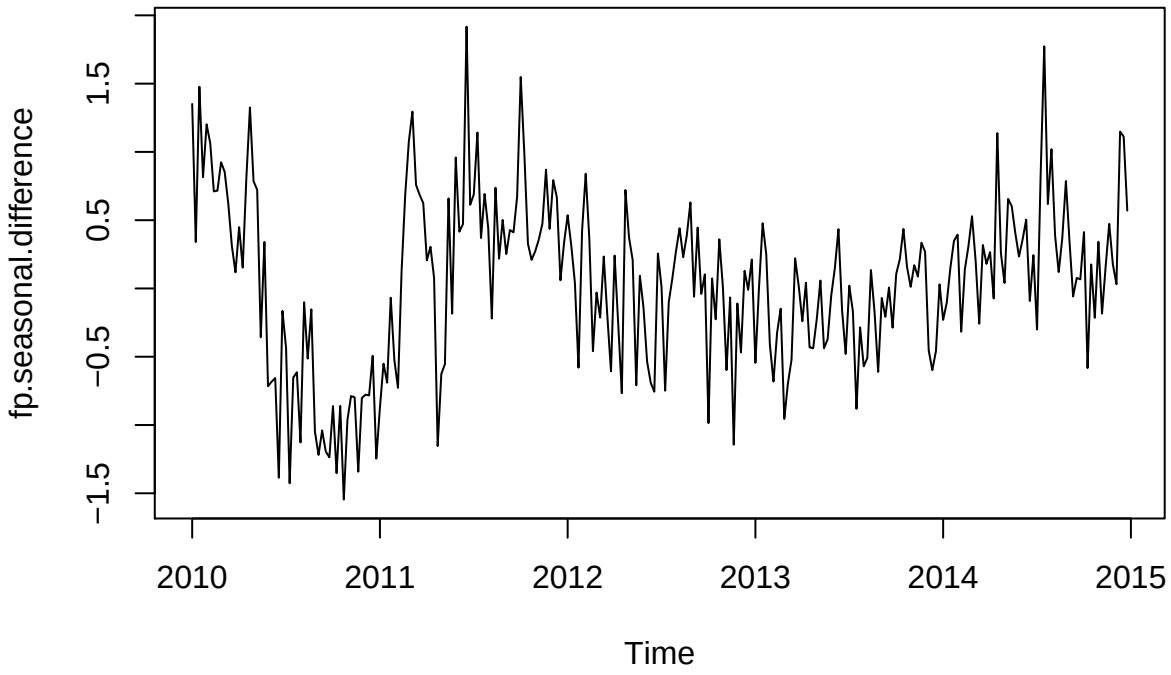
Moving on to the seasonal component...

These are weekly data and they exhibit a yearly seasonal pattern. Therfore, m = 52. One way to determine whether or not we need to apply a seasonal difference is by examining whether or not the mean value of the time-series differs by each period. We can use the function *monthplot* to examine this.

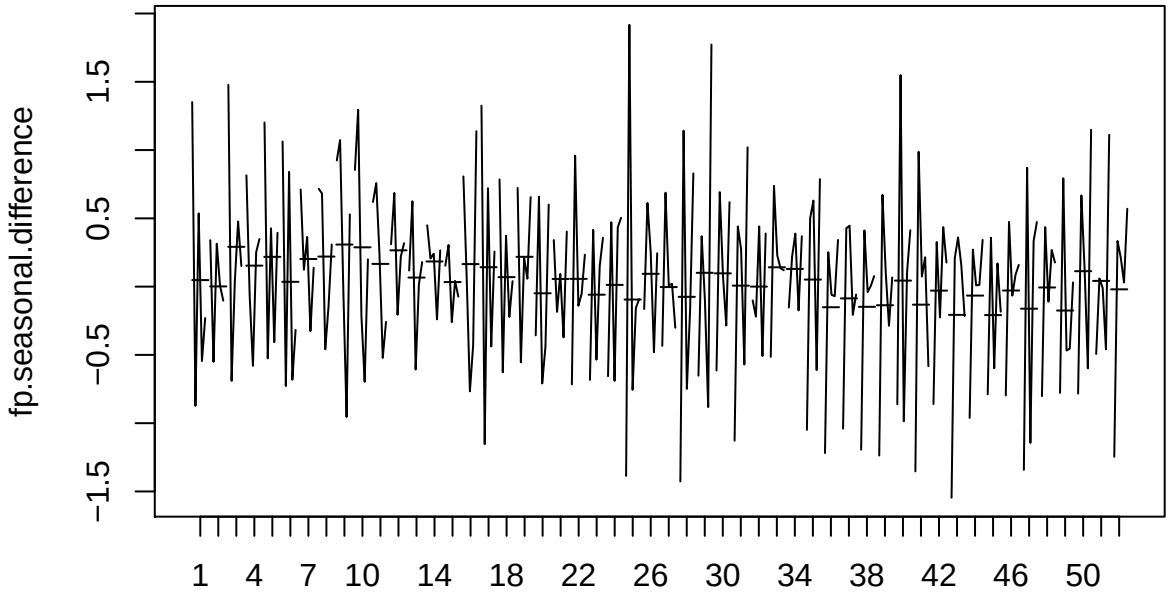
```
monthplot(fp.training)
```




```
fp.seasonal.difference <- diff(fp.training, lag = 52)

plot(fp.seasonal.difference)
```




```
monthplot(fp.seasonal.difference)  # Mean seems stable over each period now
```



Group discussion: Modeling the non-seasonal component

We have established that for our modeling purposes, d = 1 and D = 1. Now, we need to model the non-seasonal component of the raw series. In order to do that, we are going to use the Arima function in the forecast package.

```
# Let's model the non-seasonal component using a pure MA

for (q in 0:5) {
    mod <- Arima(fp.training, order = c(0, 1, q), seasonal = list(order = c(0,
```

```
        1, 0), 52), method = "ML")

    print(c(q, mod$aic))
}
```

```
## [1]   0.0000 413.9218
## [1]   1.0000 341.1046
## [1]   2.0000 342.3915
## [1]   3.0000 343.5495
## [1]   4.0000 344.7474
## [1]   5.0000 346.6972
```

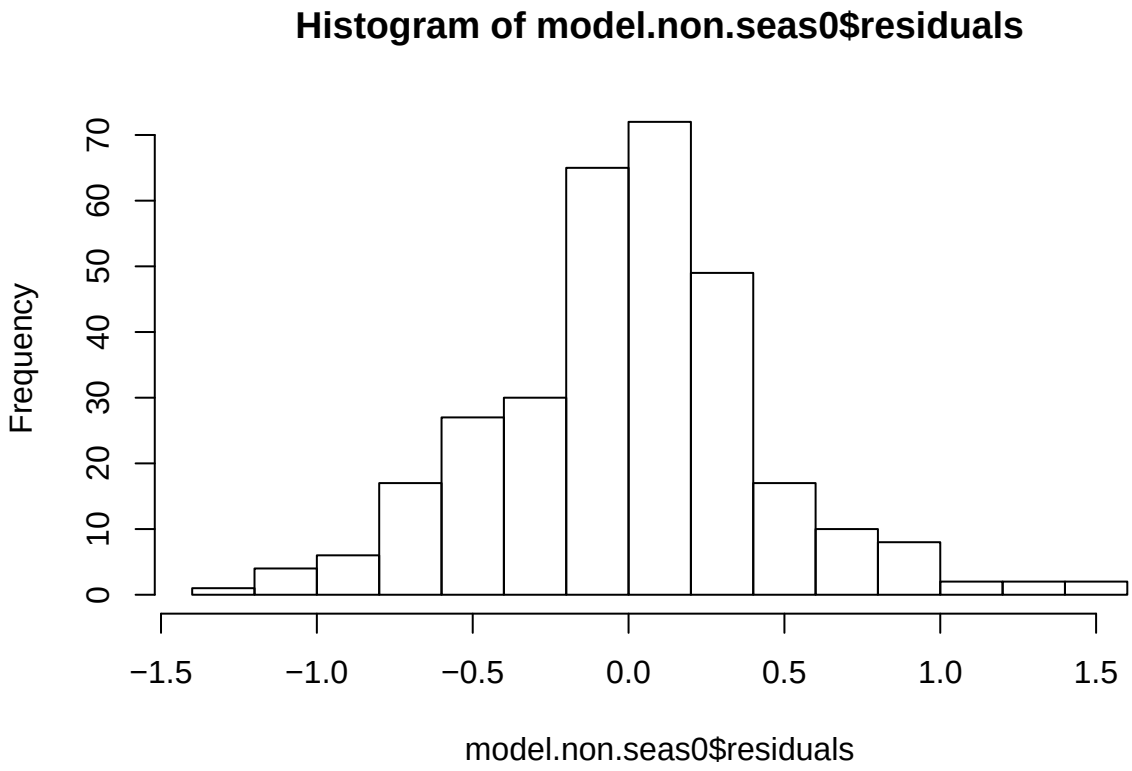
```
# Let's look at some ARIMA models
for (q in 0:3) {
    for (p in 0:3) {
        mod <- Arima(fp.training, order = c(p, 1, 0), seasonal = list(order = c(0,
            1, 0), 52), method = "ML")

        print(c(p, q, mod$aic, mod$bic))
    }
}
```

```
## [1]   0.0000   0.0000 413.9218 417.4786
## [1]   1.0000   0.0000 364.5806 371.6943
## [1]   2.0000   0.0000 351.6540 362.3244
## [1]   3.0000   0.0000 349.7552 363.9825
## [1]   0.0000   1.0000 413.9218 417.4786
## [1]   1.0000   1.0000 364.5806 371.6943
## [1]   2.0000   1.0000 351.6540 362.3244
## [1]   3.0000   1.0000 349.7552 363.9825
## [1]   0.0000   2.0000 413.9218 417.4786
## [1]   1.0000   2.0000 364.5806 371.6943
## [1]   2.0000   2.0000 351.6540 362.3244
## [1]   3.0000   2.0000 349.7552 363.9825
## [1]   0.0000   3.0000 413.9218 417.4786
## [1]   1.0000   3.0000 364.5806 371.6943
## [1]   2.0000   3.0000 351.6540 362.3244
## [1]   3.0000   3.0000 349.7552 363.9825
```

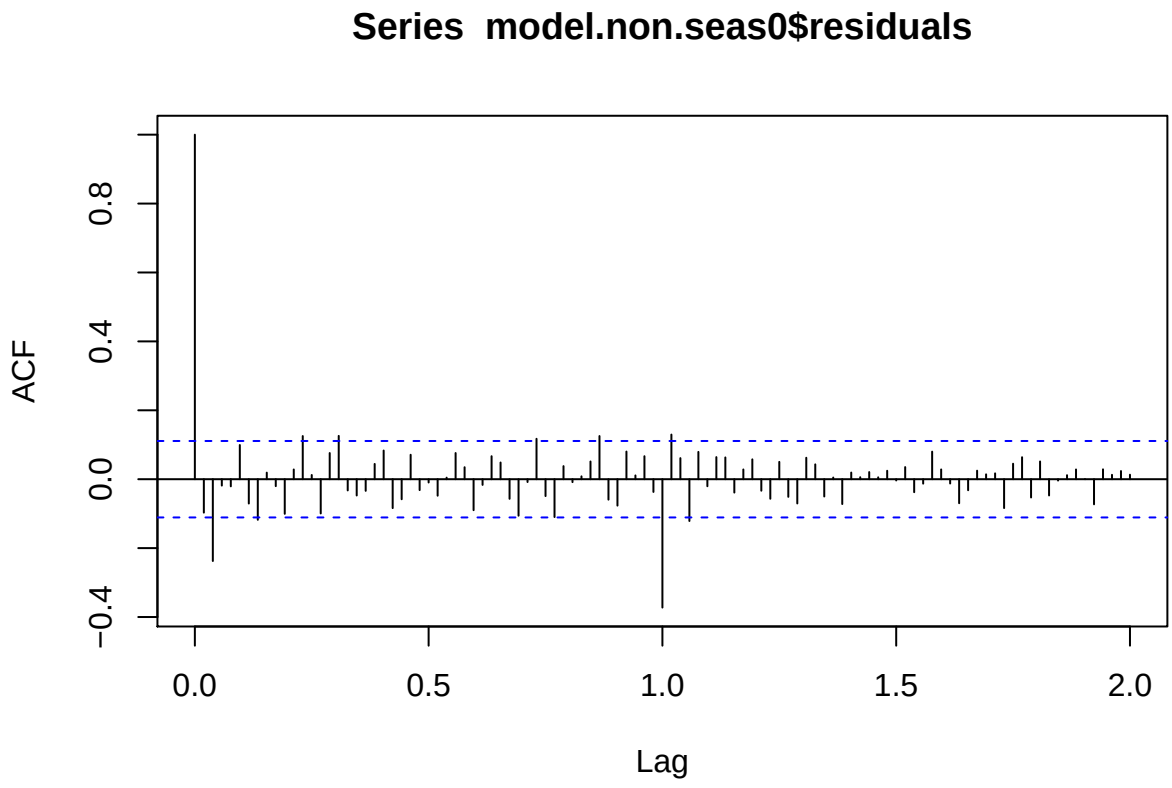
```
# The AIC's are all pretty similar (except for ARIMA(0,1,0))
# Let's look at the residuals for the simplest models

model.non.seas0 <- Arima(fp.training, order = c(1, 1, 0), seasonal = list(order = c(0,
    1, 0), 52), method = "ML")
model.non.seas0
```

```
## Series: fp.training
## ARIMA(1,1,0)(0,1,0)[52]
##
## Coefficients:
##           ar1
##       -0.4269
## s.e.   0.0566
##
## sigma^2 estimated as 0.2363:  log likelihood=-180.29
## AIC=364.58   AICc=364.63   BIC=371.69
```

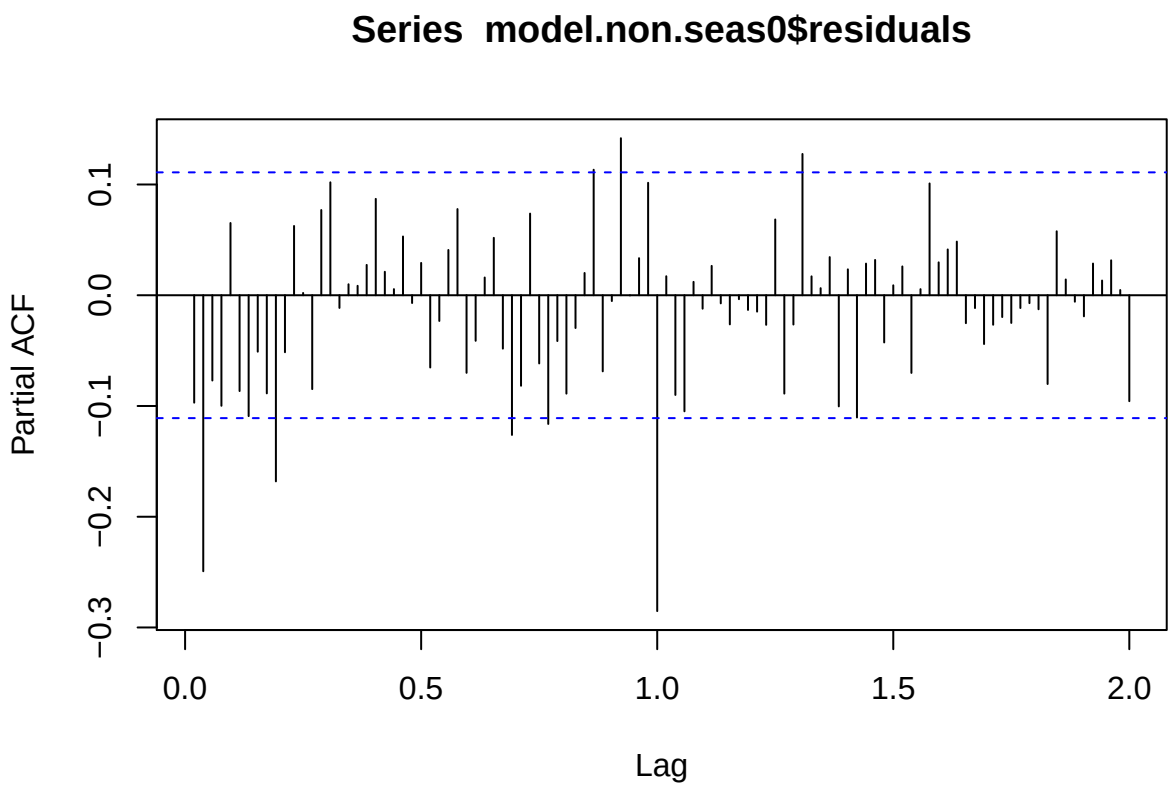
```
hist(model.non.seas0$residuals)
```




```
acf(model.non.seas0$residuals, lag.max = 104)
```



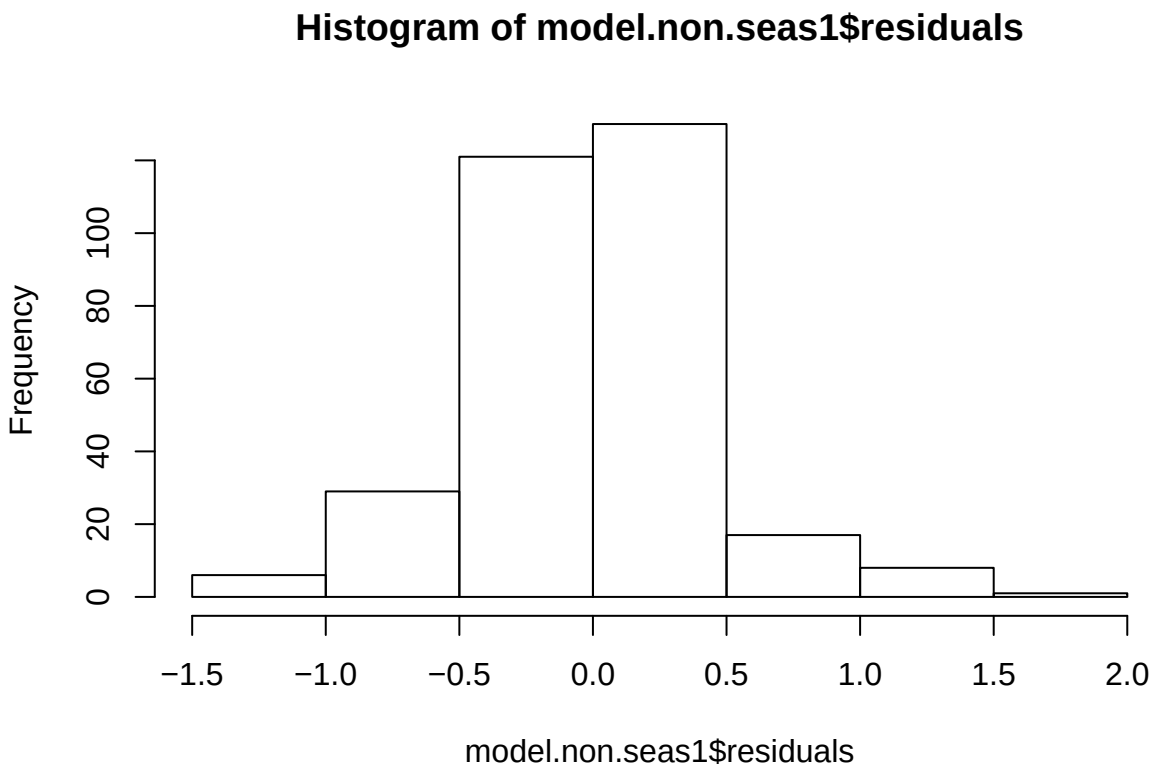
```r
pacf(model.non.seas0$residuals, lag.max = 104)
```



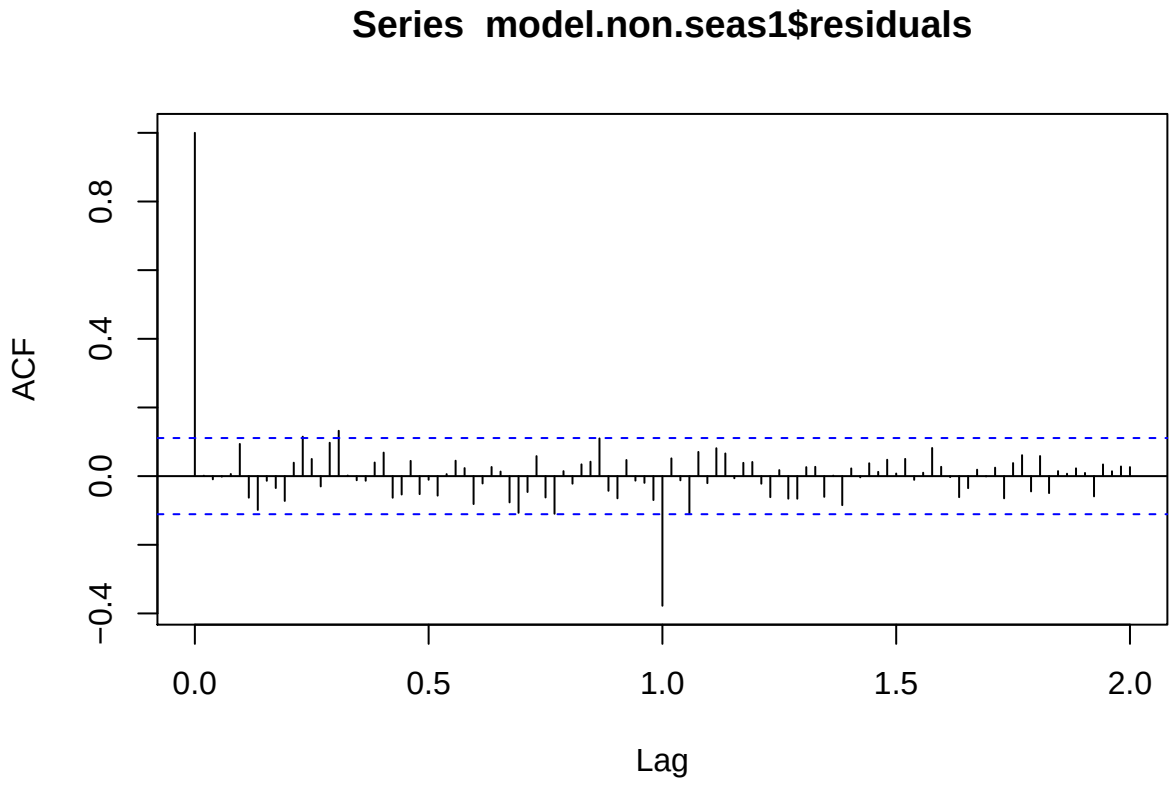

```r
# Let's look at a more complicated model
model.non.seas1 <- Arima(fp.training, order = c(3, 1, 1), seasonal = list(order = c(0,
    1, 0), 52), method = "ML")
model.non.seas1
```

```
## Series: fp.training
## ARIMA(3,1,1)(0,1,0)[52]
##
## Coefficients:
##          ar1     ar2     ar3      ma1
##       0.2536  0.1245  0.0697  -0.8464
## s.e.  0.0975  0.0807  0.0743   0.0741
##
## sigma^2 estimated as 0.2153:  log likelihood=-166.9
## AIC=343.79   AICc=344.03   BIC=361.58
```

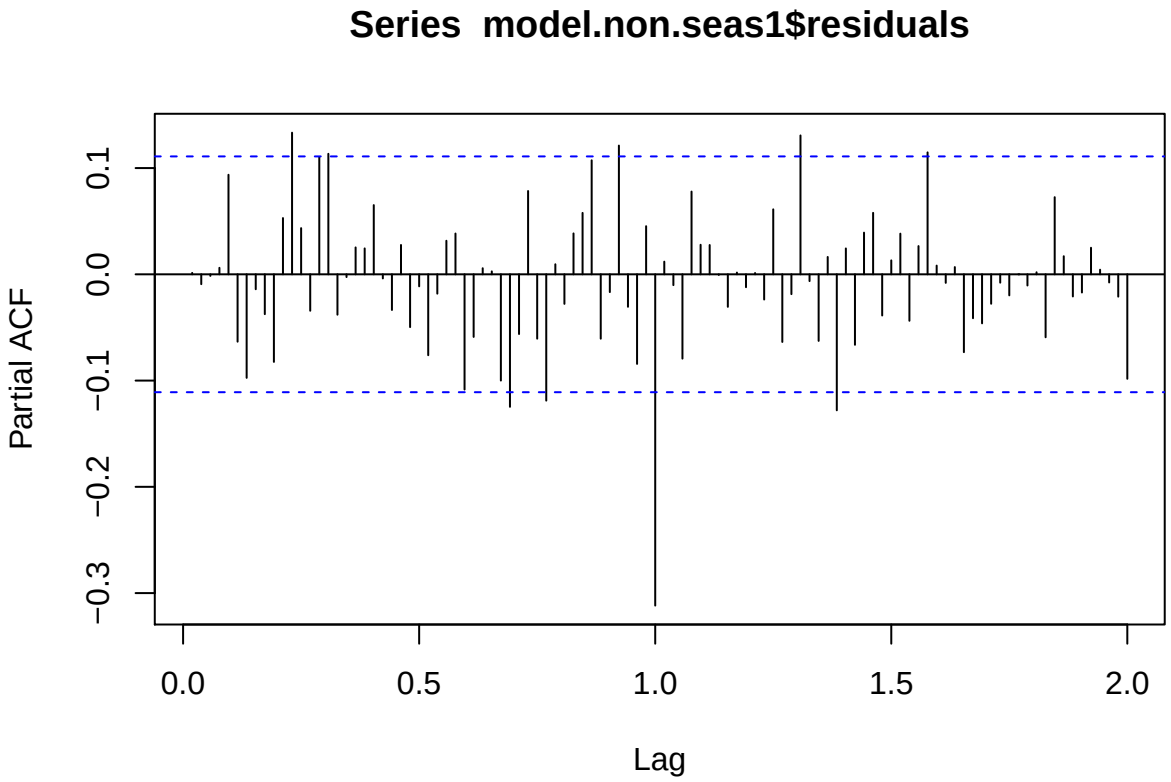
```r
hist(model.non.seas1$residuals)
```




```
acf(model.non.seas1$residuals, lag.max = 104)
```




```
pacf(model.non.seas1$residuals, lag.max = 104)
```




```
# Let's set p and q at 1 and 0 for now.  We should bear in
# mind that once we fit the seasonal component, we should
# examine simpler models
```

Breakout Session: Modeling the seasonal component

In your group, try to find appropriate values for P and Q. Use the code from above if you wish or you can use your own. For now, set p,d,q to 1,1,0 respectively, but keep in mind that we might have to change our values for p and q after we add the seasonal component!

```
# INSERT CODE
```

Take home exercise

By now, you may have more than one candidate model. For each model:

1. Examine the residuals of your candidate models. Which ones produce well behaved residuals?

2. Conduct out of sample tests on the test data. Which model has the lowest forecasting error?