A Song of Ice and Fire

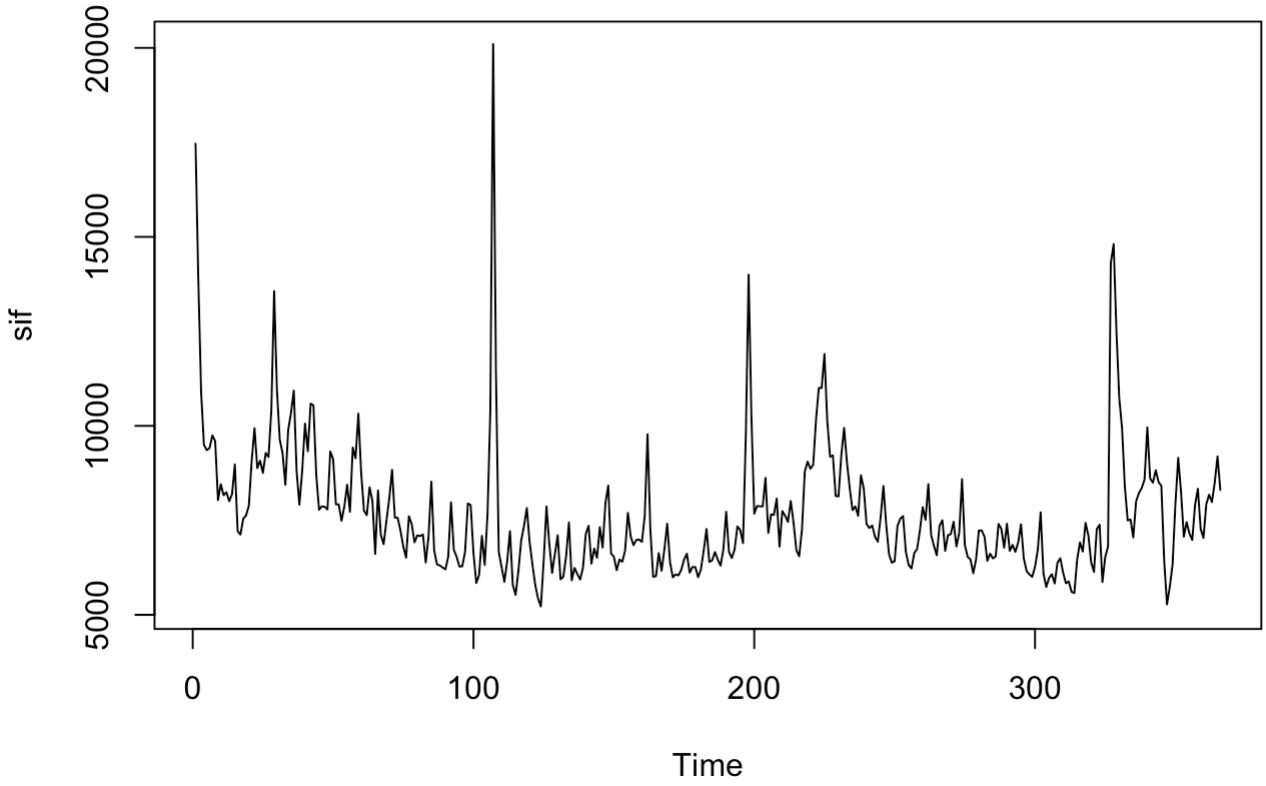
Samarth Patil

```
suppressPackageStartupMessages({
  library(TSA)
  library(ggplot2)
  library(dplyr)
  library(forecast)
  library(tseries)
  library(lmtest)
  library(Metrics)
})
```

Wikipedia web visit (Sessions per day) was counted. Data were collected from 11.29.2015 to 11.28.2016 – 366 days in total.

Loading the data and plotting it.

```
wiki_data <- read.csv('./data/Wiki_A_Song_of_Ice_and_Fire_web_visit-3.txt', header=F)
sif=ts(wiki_data$V1)
ts.plot(sif)
```



Split the time series into a training set and a test set. We are interested in forecasting the web visit volume in the next 7 days. So define the training set as the first 359 data points, and the test set as the last 7 data points.

```
sif_train=ts(sif[1:359], start=1, end = 359)
sif_test=ts(sif[360:366], start=360, end=366)
```

Run the auto arima to see how it performs. Roughly a coefficient is significant if its magnitude is at least twice as large as its standard error.

```
arima0=auto.arima(sif_train)
arima0
```

```
## Series: sif_train
## ARIMA(2,1,1)
##
## Coefficients:
##          ar1      ar2      ma1
##       0.7736  -0.2261  -0.9163
## s.e.  0.0647   0.0580   0.0447
##
## sigma^2 = 1331249:  log likelihood = -3031.15
## AIC=6070.31   AICc=6070.42   BIC=6085.83
```

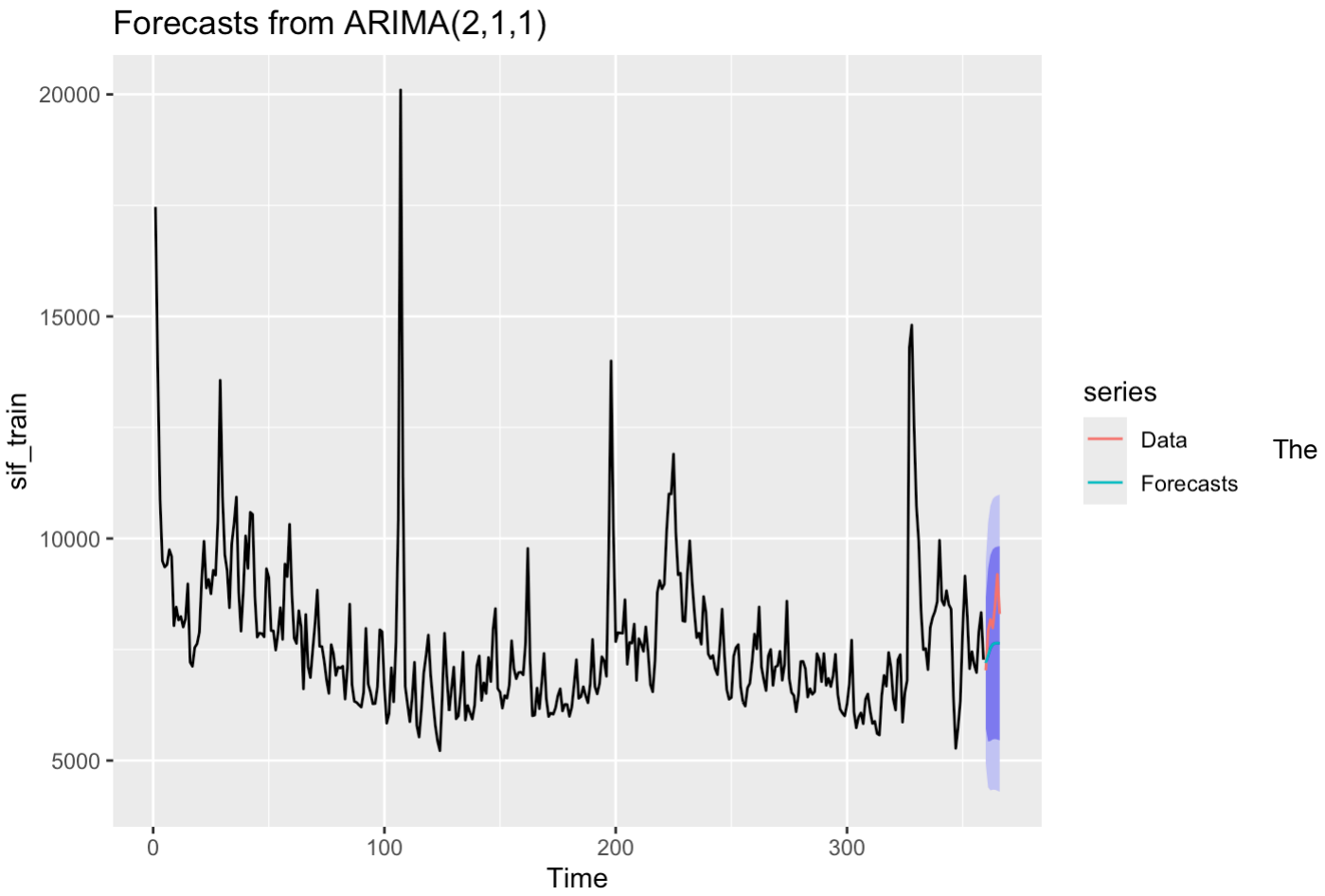
```
coeftest(arima0)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.773571   0.064697  11.9569 < 2.2e-16 ***
## ar2 -0.226077   0.058048  -3.8947 9.833e-05 ***
## ma1 -0.916277   0.044721 -20.4888 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All three coefficients are significant

Forecast the web visit volume in the next 7 days. Plot the forecasts, the raw data, and the 80% and 95% prediction intervals. Our lowest expectation is that at least the 95% prediction interval should cover the true data. Our highest expectation is that the forecasts highly align with the true data.

```
arima0_forecast=forecast(arima0, h=7)
autoplot(arima0_forecast)+autolayer(sif_test, series="Data") +
  autolayer(arima0_forecast$mean, series="Forecasts")
```



The forecasts are acceptable, but not perfect, as we can see that it does not capture the trend. However, even the 80% confidence contains the true values.

Calculate the RMSE of the forecasts. This RMSE will be used as a benchmark for comparison later.

```
rmse_arima=rmse(arima0_forecast$mean,sif_test)
```

Create regressors.

```
t1=1:359
t2=t1^2
t1_test=360:366
t2_test=t1_test^2
```

On the training set, run a linear regression using `sif` against both `t1` and `t2`.

```
lm0=lm(sif_train~t1+t2)
summary(lm0)
```

```
##
## Call:
## lm(formula = sif_train ~ t1 + t2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2710.7  -922.0  -356.4   462.6 12495.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.439e+03  2.540e+02  37.157  < 2e-16 ***
## t1          -2.290e+01  3.259e+00  -7.027 1.08e-11 ***
## t2           5.392e-02  8.766e-03   6.151 2.07e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1595 on 356 degrees of freedom
## Multiple R-squared:  0.1365, Adjusted R-squared:  0.1316
## F-statistic: 28.13 on 2 and 356 DF,  p-value: 4.555e-12
```

All coefficients are significant

Extrapolate the results to the test set and plot them.

```
X_test=data.frame(t1=t1_test, t2=t2_test)
lm0_forecast=forecast(lm0,newdata = X_test)
lm0_forecast=ts(lm0_forecast$mean,start = 360,end = 366)
lm0_forecast
```

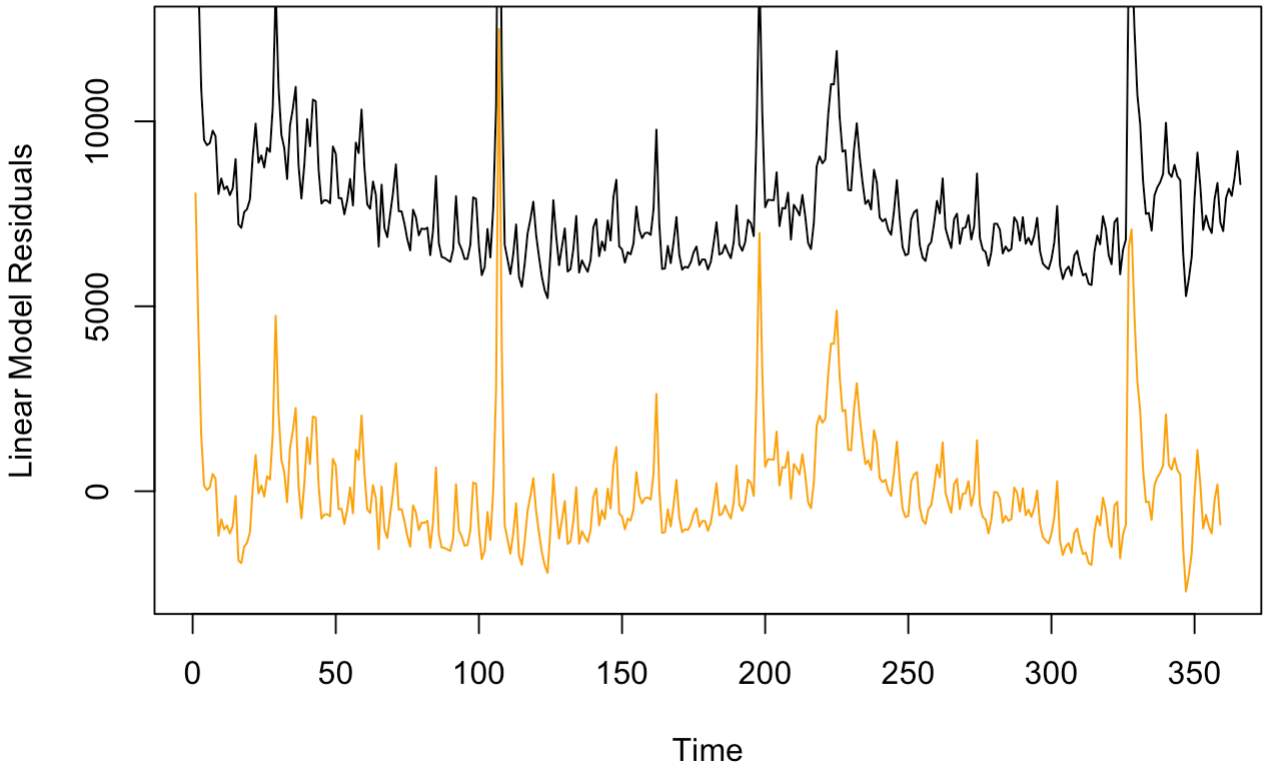
```
## Time Series:
## Start = 360
## End = 366
## Frequency = 1
##        1        2        3        4        5        6        7
## 8183.575 8199.554 8215.641 8231.836 8248.138 8264.548 8281.067
```

Calculate the RMSE on the test set for the linear model above.

```
rmse_lm=rmse(lm0_forecast,sif_test)
```

Extract the residuals of the linear model above. Consider it as a new time series for Arima. Plot the residuals and compare the curve with the one above.

```
z=ts(lm0$residuals, start = 1, end = 359)
ts.plot(z, ylab = "Linear Model Residuals", col='orange')
lines(sif)
```



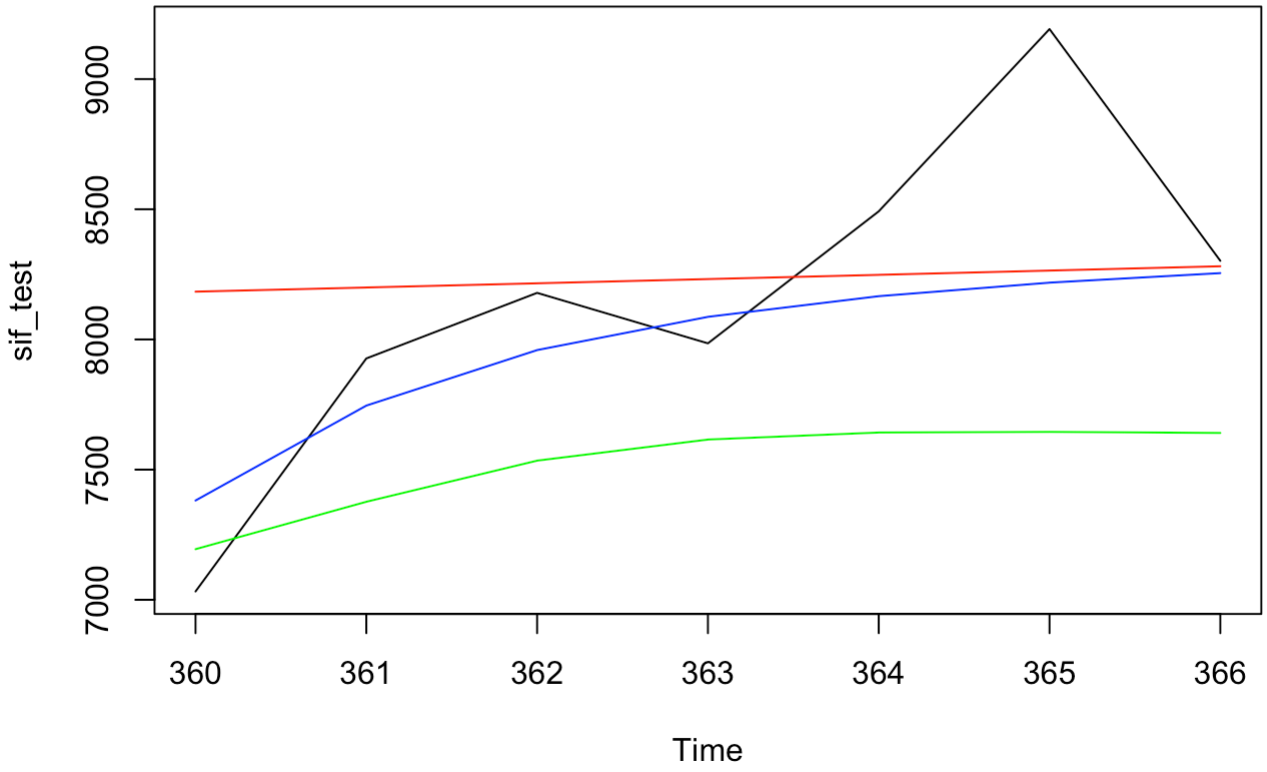
Run an `auto.arima()` on the residuals, and calculate the forecasts on the test set.

```
arima1=auto.arima(z)
z_forecast=forecast(arima1, h=7)
coeftest(arima1)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 0.565440   0.062928  8.9856 < 2.2e-16 ***
## ma1 0.281578   0.070390  4.0003 6.327e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now calculate the forecasts combining both the linear model and the ARIMA, and plot the results from all three models (i.e., arima alone, linear model alone, linear model plus arima)

```
y_forecast=lm0_forecast+z_forecast$mean
ts.plot(sif_test)
lines(y_forecast, col='blue')
lines(arima0_forecast$mean, col='green')
lines(lm0_forecast, col='red')
```



Now calculate the RMSE on the test set for the linear model plus arima above.

```
rmse_lm_plus_arima=rmse(y_forecast,sif_test)
```

rmse_lm_plus_arima is lower than both rmse_lm and rmse_arima

Fine-tune the model

```
for(i in 1:10){
  for( j in 1:10){
    arima2=Arima(z,order=c(i,1,j), method="ML")
    z_forecast1=forecast(arima2, h=7)
    y_forecast1=lm0_forecast+z_forecast1$mean
    rmse_lm_plus_your_arima=rmse(y_forecast1,sif_test)
    if(rmse_lm_plus_your_arima < 400){
      print(paste0(rmse_lm_plus_your_arima,'-',i,'-',j))
    }
  }
}
```

```
## [1] "396.576026380795-7-8"
```

arima(7,1,8) and arima(8,1,10) both beat the result above by over 10% in terms of RMSE