Time Series Analysis of the distance between the Moon and the Earth.

Swapnil Bhatta

University of Southern Mississippi

1) Introduction

The Moon is a natural satellite of Earth that orbits it in the prograde direction, which means it moves from west to east relative to the stars. It is in synchronous rotation with the earth, showing only a single side, and revolves around a barycenter within the earth itself. The apogee and perigee, which mean the maximum and minimum distance to the Moon are at an average of 405,400 km and 362,4200 km with the eccentricity of its orbit being around 0.0549006 (Murphy, 2013). The Moon revolves around the earth at a mean orbital velocity of 1.022 km/s and had a sidereal and synodic periods of 27.322 days and 29.530 days respectively. This means a complete period of the moon with respect to background stars is about 27 days while the period with respect to the sun is about 29 days.

Considering just our solar system, the Moon revolves around the Earth, and thus around the Sun. The interaction between these bodies is dominated by the force of gravity. The gravitational force between two bodies is given by Newton's universal law of gravitation which is essentially a force directly proportional to the product of the mass of the two bodies and inversely proportional to the square of the distance separating them.

The time series data used was obtained from NASA's New Horizons program, and contains various features of the moon ranging from its luminosity to its declination. The variable of interest for this project is the distance between the center of the earth and the center of the Moon in AU (Astronomical Units), along with how its variation over time. An astronomical unit is the distance between the Sun and the Earth which is about 1.49×10^8 km. The data starts from January 1, 1951 and continues till this day with the frequency of collection being every day. The analysis of this data can be used to see the seasonal changes in the distance between the earth and the Moon, which could be useful for planned lunar rovers, study of the luminosity gradient of the Moon or even the level of tidal waves on Earth.

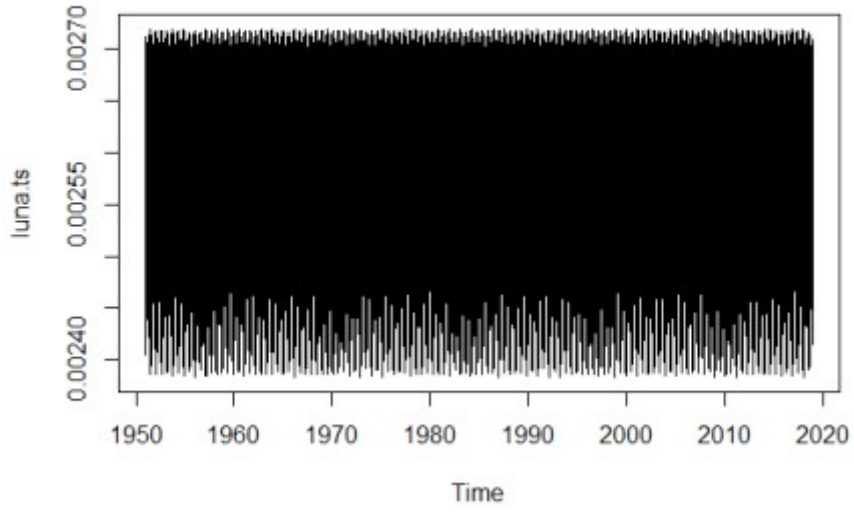

Fig 1: A plot of the time series for lunar distance (in AU) vs time from 1951 to 2018 AD.

Truncated Data:

The plot of the whole time series looks rather ineligible because a lot of data points are congested into a small x-scale. There is no linear trend visible, while minor seasonal variation can be suspected. Truncating the data points to a smaller time period gives a better visualization of the change in distance as a function of time.

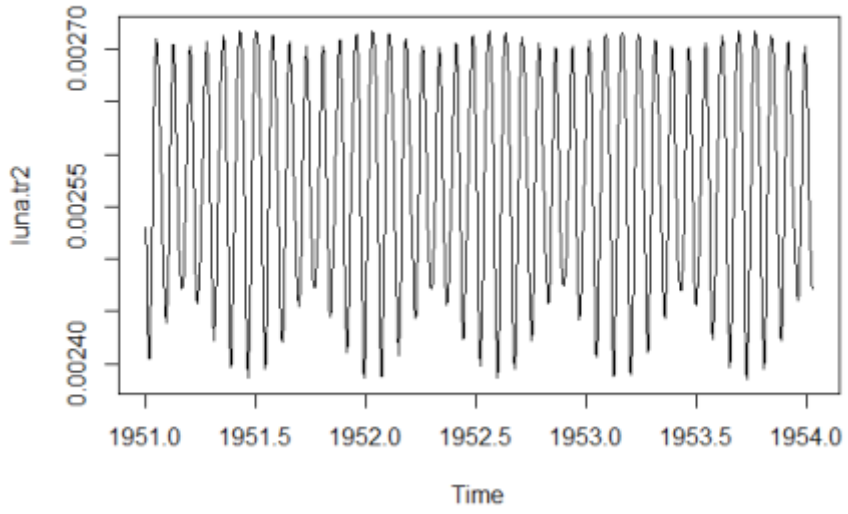

Fig 2: A truncated plot of the time series for lunar distance (in AU) vs time from 1951 to 1954AD.

This plot gives a lot clearer picture of how the distance changes overtime. A look at the lower half of the plot suggests more than one frequency coming into play. The relationship between the variables could be influenced by more than just the gravitational attraction of the Earth. To properly analyze the data and see how accurate the modeling of the phenomena works, the selected truncation is further truncated as seen in figure two.

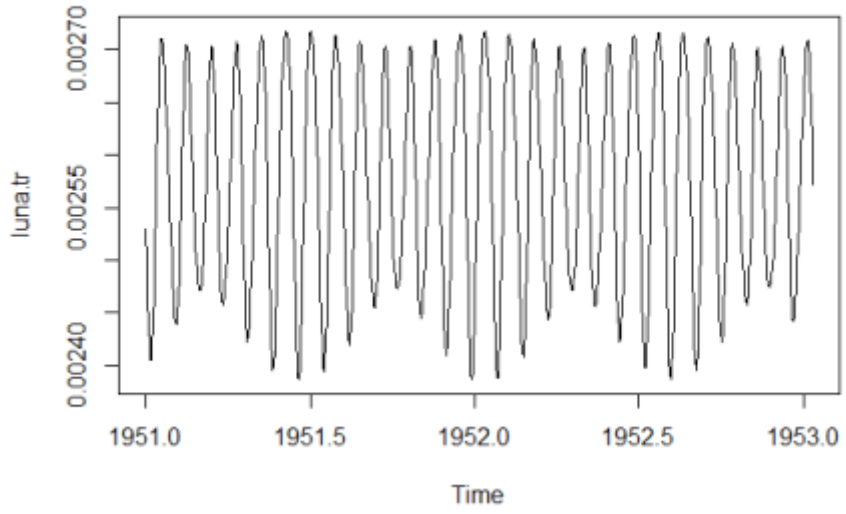

Fig 3: A further truncated plot of the time series for lunar distance (in AU) vs time from 1951 to 1953 AD.

To get an estimate of the present frequencies, spectral analysis is performed on the data by generating a periodogram for the truncated part. The periodogram "is based on the squared correlation between the time series and sine/cosine waves of frequency ω, and conveys exactly the same information as the auto-covariance function" (Crawley). As seen in the plot in figure four, the dominant frequencies within the confidence interval, denoted by the blue line in the top right corner, are around the range of 10-30 cycles per year.

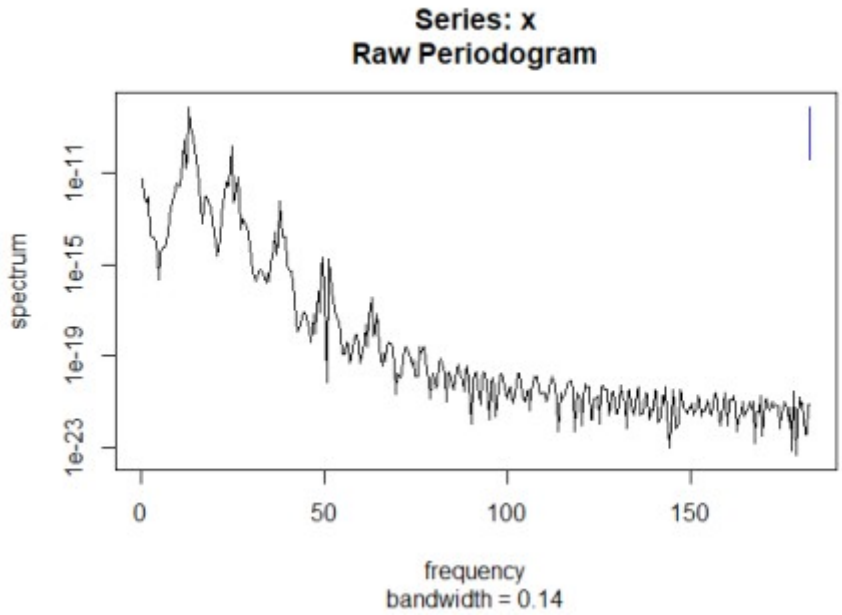

Fig 4: A plot of the raw periodogram for the truncated data, the frequency is plotted against the corresponding spectrum.

Regression Fit:

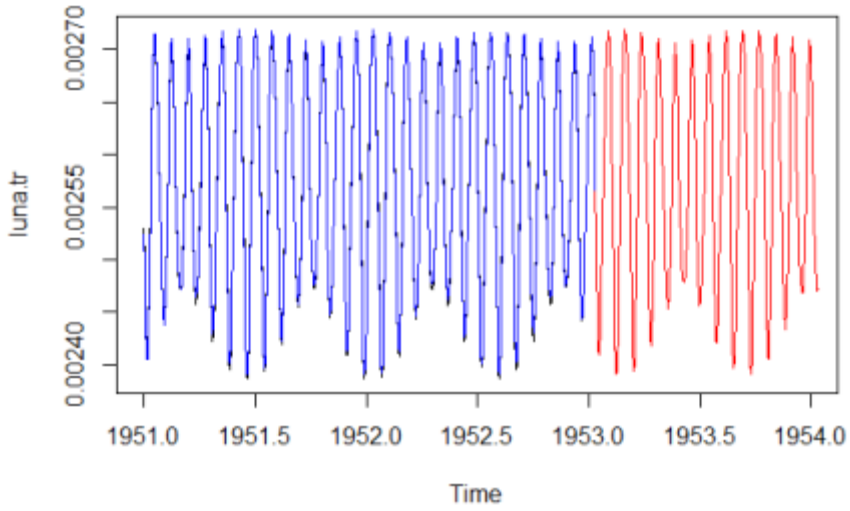

Fig 5: A plot of the time series for the lunar distance (in AU) vs time from 1951 to 1953 in black, with a regression fit for the same period in blue, and a prediction using the fit for 1954 in red.

Fourier analysis is a means of analyzing a periodic function by breaking it down into simple sinusoidal or harmonic components, whose summation would result in the Fourier series being analyzed. The plot suggested that the data was a combination of multiple sinusoidal waves, which is what lead to the implementation of a linear model with a combination of sines and cosines.

The non-linear nature of the data along with varying amplitudes and frequencies made it difficult to get a proper fit. The first frequency obtained from the analysis of the periodogram's critical points and a looped linear module was found to be 13.25 cycles per year. This corresponds to 27.54 days which is approximately the Moons sidereal period. Using this frequency, harmonics were computed but none matched the actual dataset with a sufficient accuracy. Further analysis suggested a model of more than one frequency was needed, and a nested for loop was used to look for appropriate frequencies which gave a better fit and two more frequencies were found at 1.91 cycles per year and 3.53 cycles per year. These correspond to 191 and 103 days but do not have any astronomical significance associated with them. Using these frequencies and their corresponding trigonometric function with a high level of marginal significance, a linear model was obtained with about 99% of the variance accounted for. The fit was still a little off at certain points as seen by the contrast black dots on the blue regression in certain sections of figure five.

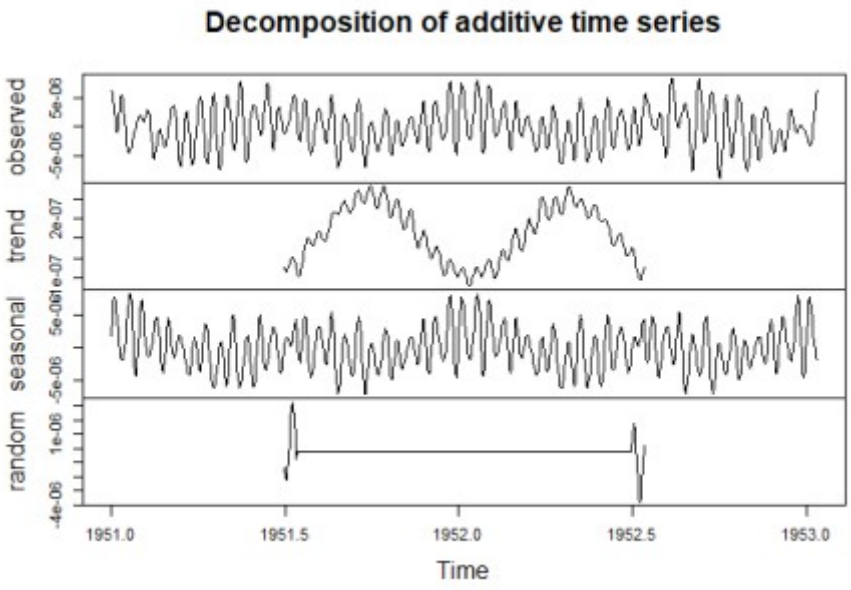

Fig 6: A plot of the decomposition of the residuals remaining from the regression fit.

Computing and analyzing Residuals

Residuals are essentially the difference between the observed value and the predicted value. From the fit obtained in figure five, the residuals were calculated by subtracting the model from the fit and plotting the decomposition of it in figure six. As seen in the subplot titled seasonal, the residuals still show certain seasonality. This seasonality is not random as the random subplot is of a much smaller magnitude.

The autocorrelation function of the residuals of the regression fit is plotted in figure seven. Most of the lags present are above the confidence interval which means further analysis of the residuals is necessary.

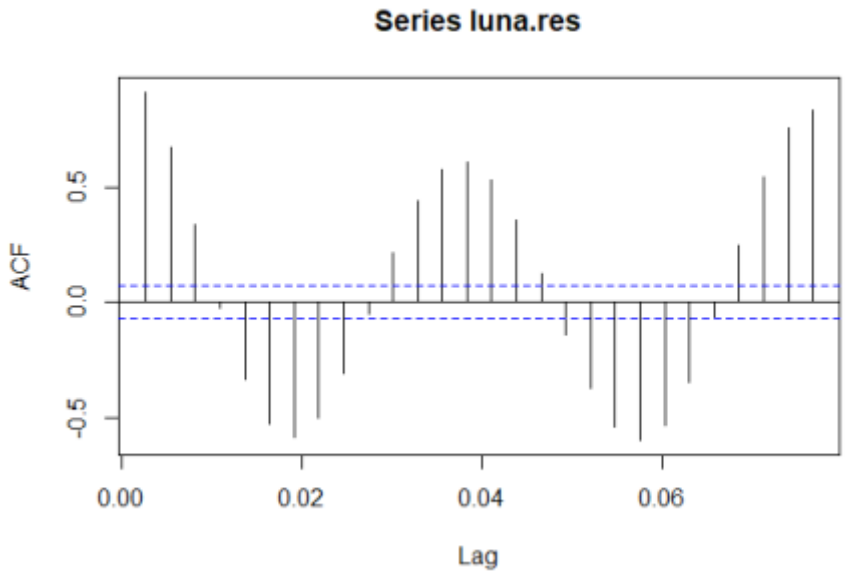

Fig 7: Autocorrelation function plot of the residuals of the regression fit.

Attempting to model these residuals with an ARIMA fit gave almost the identical wave itself. Auto-Regressive Integrated Moving Average, known more commonly as ARIMA is a model that regresses significant variables into their own lags to get a better fit for the data. Fitting an ARIMA model to the regression fits residuals gave a best fit at an order of c(7,0,10). The regression model after removing the residuals accounted for by the ARIMA model is plotted in figure eight.

As seen in both plots, the model with the added ARIMA data does a great job of explaining almost all variations. The ACF for the remaining residual is plotted in figure nine. The significance of the lags in the plot still reject the null hypothesis that the residuals are just white noise, suggesting the need of further analysis. Comparing it with the ACF of the initial residuals however, shows improvement.

	1 0

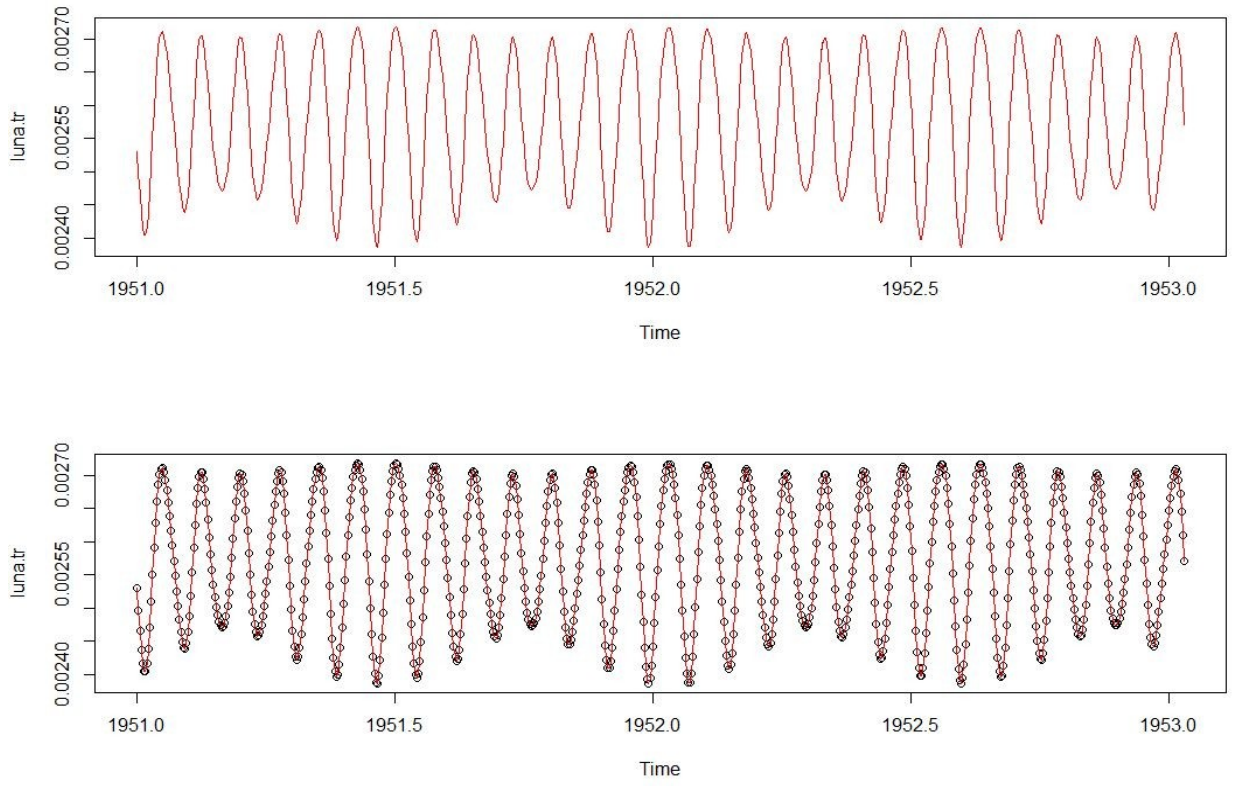

Fig 8: Plot for the regression fit with ARIMA data of the lunar distance (in AU) vs time from 1951 to 1953. The first plot is the actual data in black and the model plotted together over it in red, the second plot is the model plotted in red over the data points in black.

	1 1

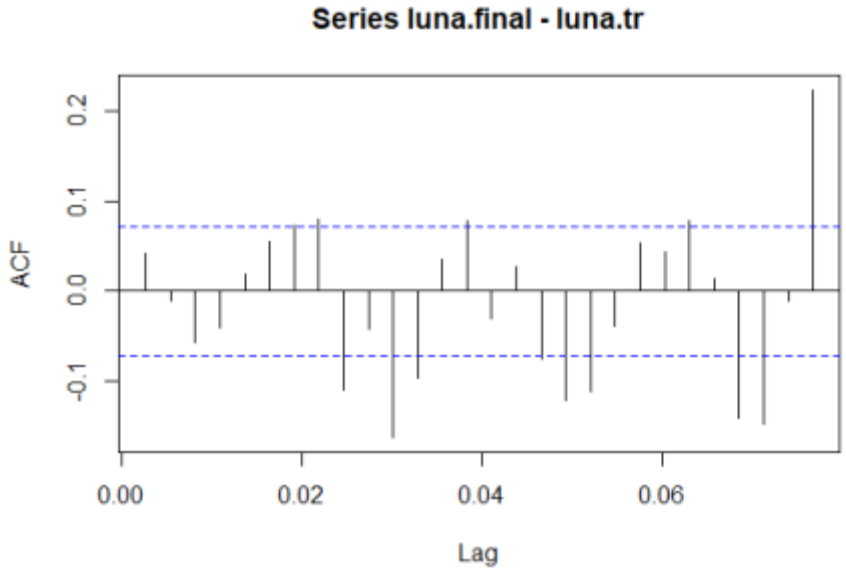

Fig 9: ACF plot of the residuals of the regression fit with the ARIMA data.

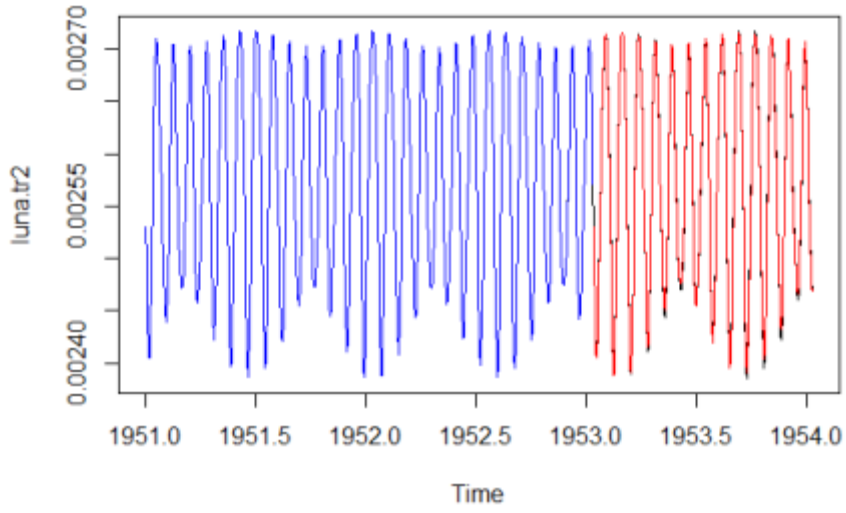

Fig 10: A plot of the time series for the lunar distance (in AU) vs time from 1951 to 1954 in black, with a regression fit with ARIMA data for 1951 to 1953 in blue, and the predicted fit with ARIMA data for 1954 in red.

	1 2

The prediction portion from the ARIMA data did not work out as expected and the fit for the future data was not as on point as the regression and ARIMA model for the initial plot. The increase in the explained variation was a mere .1% comparing between the actual prediction and the prediction with the ARIMA data.

Linear significance

A phenomena know as tidal acceleration between the Earth and the Moon is causing the gradual recession of the Moon and it is slowly but steadily moving away from the Earth. For time series of the lunar orbit and its distance from the earth, there was neither a true nor a pseudo replicate linear significance observed. This is expected as the rate for recession is 3.8cm/year which converted to astronomical units would be $2.54x10^{-13}$ AU.

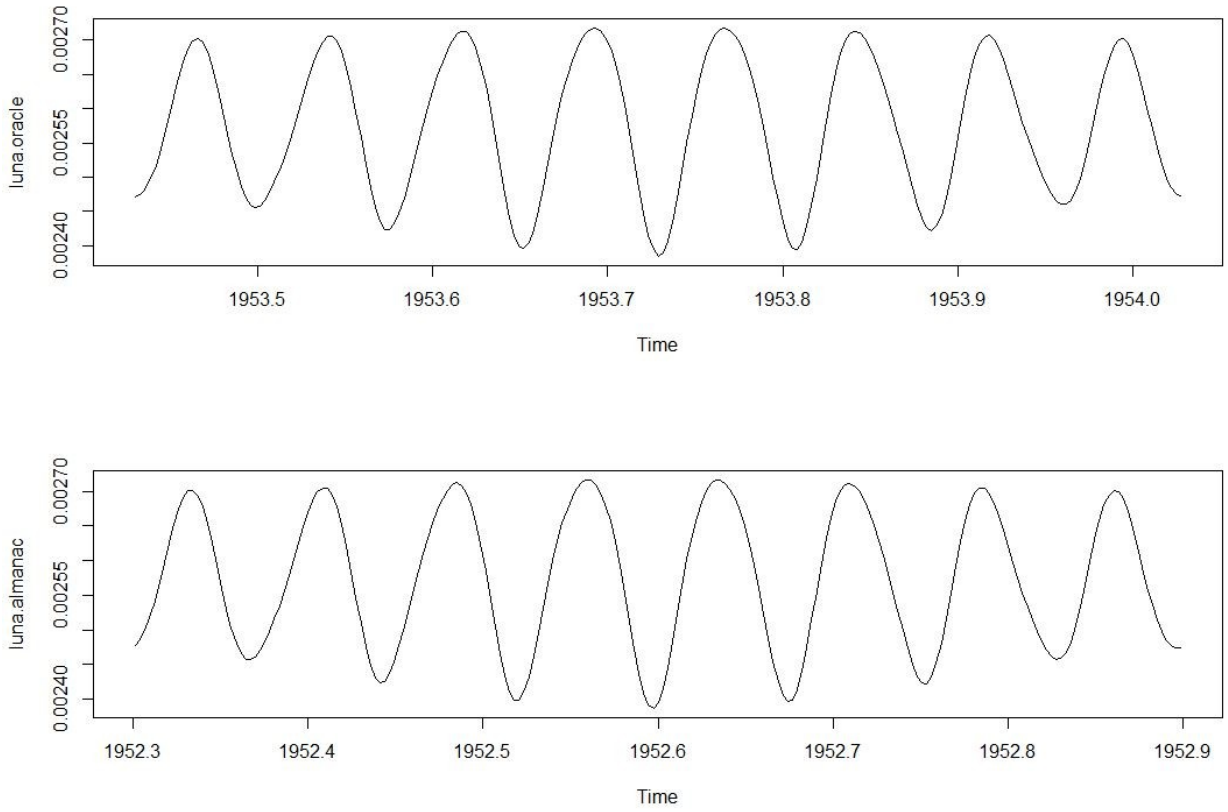

Fig 11: Oracle prediction for a period in 1953 and Almanac prediction for a period in 1952.

The observed lunar distance and its change over time is a stationary process, which is simply a stochastic process that does not change when shifted in time. Given this, future predictions of a period would essentially be a replication of the past period itself. This makes almanac and oracle data sets redundant in analyzing the quality of the prediction. Figure five shows different time periods in the time series with very similar, if not same data points.

References

HORIZONS System. (n.d.). Retrieved March 26, 2019, from https://ssd.jpl.nasa.gov/?horizons

Micheal J. Crawley. (n.d.). *The R Book, 2nd Edition | Statistical Software / R | Computational & Graphical Statistics | General & Introductory Statistics | Subjects | Wiley* (2nd ed.). Retrieved from https://www.cs.upc.edu/~robert/teaching/estadistica/TheRBook.pdf

Murphy, T. W. (2013). Lunar laser ranging: the millimeter challenge. *Reports on Progress in Physics*, *76*(7), 076901. https://doi.org/10.1088/0034-4885/76/7/076901

APPENDIX

```
## Clear console
rm(list=ls())

##Check added packages

library(features)
library(TSA)


##Loading the Lunar data into R.
luna<-read.csv("luna.csv")

##Setting the numbers up as a vector.
luna.nrs<-as.vector(na.omit(luna[,2]))

##Creating a time series out of the given data.
luna.ts<-ts(luna.nrs, start=c(1951,1,1),freq=365)

##Checking and plotting time series
luna.ts
plot(luna.ts)

##Truncating a window in the time series for better analysis and plotting it.

luna.tr<-window(luna.ts, end=c(1953,12))
plot(luna.tr)
luna.tr2<-window(luna.ts, end=c(1954,12)) ## For the future.
plot(luna.tr2)

##Getting the time part of luna.tr
t<-time(luna.tr)

##Generating a spectrum of the time series to observe relevant frequencies and get the
corresponding frequencies and getting a summary of it. (See text)
spec<-spectrum(luna.ts)
spec<-spectrum(luna.tr)
```

```
##Using the features package to extract all the critcial points from the raw peridogram of the series.
luna.feat<-features(spec$freq, spec$spec)
luna.fget<-fget(luna.feat)


##For critical points; luna.fget$crit.pts[i]

###### Regression fit

##Attempt at regression:(frequencies through luna.fget)
luna.fget$crit.pts[1]


## Fitting the data

##best fit around the first critical point, corresponds with the period of the moon.
# for (i in 1200:1500){
#   w2=i/100
#   w=w2*2*pi
#   luna.sinemod <- lm(luna.tr ~  cos(t*w)+sin(t*w))
#   if ((summary(luna.sinemod)$r.squared) > .9799){
#     print(w2)
#     print(summary(luna.sinemod)$r.squared)
#
#   }
# }

w=13.25*pi*2

##Looking for the best second frequency [Long computation, do not execute]
# for (i in 1:36000){
#   w2=i/100
#   w=w2*2*pi
#   luna.sinemod <- lm(luna.tr ~  cos(t*w1)+sin(t*w1)+ cos(t*w) +sin(t*w) + cos(2*t*w) + sin(2*t*w) + cos(3*t*w) + sin(3*t*w) + cos(4*t*w) + sin(4*t*w) + cos(5*t*w) + sin(5*t*w) + cos(6*t*w) + sin(6*t*w) + cos(7*t*w) + sin(7*t*w) +
#                       cos(8*t*w) + sin(8*t*w))
#   if ((summary(luna.sinemod)$r.squared) > .9799){
#     print(w2)
#     print(summary(luna.sinemod)$r.squared)
#
#   }
# }
```

```
##Choosing 1.91 and the harmonics with the highest P-values (Directly or using a for loop with [if (summary(model1)$coef[,"Pr(>|t|)"][i]<.01) ]

w1=1.91*pi*2

##Further analysis shows a third frequency
w2=3.53*pi*2

luna.sinemod <- lm(luna.tr ~  cos(t*w)+sin(t*w)+ cos(6*t*w1) + sin(6*t*w1) + sin(7*t*w1)+cos(3*t*w2) + sin(4*t*w2) +  cos(7*t*w2) + sin(7*t*w2))
summary(luna.sinemod)

## R^2=0.99

plot(t, luna.sinemod$fitted.values, type="l")


## Getting a line
luna.sinemodline <- luna.sinemodline <- luna.sinemod$coefficients[1]+luna.sinemod$coefficients[2]*cos(t*w) +luna.sinemod$coefficients[3]*sin(t*w) +luna.sinemod$coefficients[4]*cos(6*t*w1)+luna.sinemod$coefficients[5]*sin(6*t*w1)+ luna.sinemod$coefficients[6]*sin(7*t*w1)+luna.sinemod$coefficients[7]*cos(3*t*w2) + luna.sinemod$coefficients[8]*sin(4*t*w2) +  luna.sinemod$coefficients[9]*cos(7*t*w2) + luna.sinemod$coefficients[10]*sin(7*t*w2)

##Checking regression
plot(luna.sinemodline)
plot(luna.tr)
lines(luna.sinemodline, col=2)

##Future Prediction for a different window (The year of 1953)

##Plotting with increased window size

plot(luna.tr, xlim=c(1951, 1954))

##LM model:

lines(luna.sinemodline, col=4)

luna.predts <- window(luna.ts, start=c(1953,12), end=c(1954,12))
t2 <- time(luna.predts)
```

```
luna.predline <-  luna.sinemod$coefficients[1]+luna.sinemod$coefficients[2]*cos(t2*w)
+luna.sinemod$coefficients[3]*sin(t2*w)
+luna.sinemod$coefficients[4]*cos(6*t2*w1)+luna.sinemod$coefficients[5]*sin(6*t2*w1)+
luna.sinemod$coefficients[6]*sin(7*t2*w1)+luna.sinemod$coefficients[7]*cos(3*t2*w2) +
luna.sinemod$coefficients[8]*sin(4*t2*w2) +  luna.sinemod$coefficients[9]*cos(7*t2*w2) +
luna.sinemod$coefficients[10]*sin(7*t2*w2)
lines(luna.predline, col = 2)


##Residual
luna.res = luna.tr-luna.sinemodline

plot(luna.res)
plot(decompose(luna.res))

acf(luna.res)


##ARIMA

get.best.arima <- function(x.ts, maxord = c(1,1,1,1,1,1))
{
 best.aic <- 1e8
 n <- length(x.ts)

 for(p in 0:maxord[1])
  for (d in 0:maxord[2])
   for(q in 0:maxord[3])
    for(P in 0:maxord[4])
     for(D in 0:maxord[5])
      for(Q in 0:maxord[6])
      {
       try(
        {
         fit <- arima(x.ts, order=c(p,d,q), seas = list(order=c(P,D,Q), frequency(x.ts) ),
method = "CSS")
         fit.aic <- -2*fit$loglik + (log(n)+1) * length(fit$coef) #suppressing code after
the + gives better fits (using loglik as the only criterion that way)
         if (fit.aic < best.aic)
         {
          best.aic <- fit.aic
          best.fit <- fit
          best.model <- c(p,d,q,P,D,Q)
         } #end if
        } # end first argument of try
        , FALSE) #end try
```

```
            print(c(p,d,q,P,D,Q))
            flush.console()
          } # end for


        dev.new()


        over <- paste("Process Fit with ARIMA(", toString(best.model[1]), ",",
toString(best.model[2]), ",", toString(best.model[3]), ") Process. \n Coefficients:",
toString(round(best.fit$coef, digits=3)))
        under <- paste("Periodic Coefficients:", toString(best.model[4]), ",",
toString(best.model[5]), ",", toString(best.model[6]))

        acf(na.omit(best.fit$resid), lag.max=100, main=over, xlab=under)



        list(akaike=best.aic, data=best.fit, orders=best.model)
      } # end get.best.arima



      luna.arima <- get.best.arima(luna.res, c(10,2,10,0,0,0) )

      ## Fit with ARIMA (7,0,10)

      luna.arma <- luna.res - luna.arima$data$residuals
      ##Checking ARIMA progression plot(luna.arma, xlim=c(1951, 1954))

      luna.final <- (luna.sinemodline + luna.arma)

      plot(luna.tr)
      lines(luna.final, col=4)

      ##Divide the window into two parts
      par(mfrow=c(2,1))

      ## Plot over the line luna.tr
      plot(luna.tr, type="l")
      lines(luna.final, col =2)

      ##plot over the points in luna.tr
      plot(luna.tr, type="p")
```

```
lines(luna.final, col =2)

##Reset the window ratio
par(mfrow=c(1,1))


plot(luna.tr2)
lines(luna.final, col =4)

##Adding the ARIMA approximations

luna.predarima <- predict(luna.arima$data, n.ahead = length(t2))

lines(luna.predline+luna.predarima$pred, col=2)




##ACF residual
acf(luna.final-luna.tr)
acf(luna.predline+luna.predarima$pred-luna.predts)


################################

plot(luna.tr2)
##Orcale is the last period of the data (a larger underlying period)
luna.oracle <- window(luna.ts, start=c(1953.43), end=c(1954.03))
plot(luna.oracle)

plot(luna.tr)
##Almanac would be the year 1952
luna.almanac <- window(luna.ts, start=c(1952.3), end=c(1952.9))

plot(luna.almanac)

##Seeing them side by side
par(mfrow=c(2,1))

plot(luna.oracle)
plot(luna.almanac)

par(mfrow=c(1,1))

##Function
```

```
a=c(1,2,3,4,5)
b=c(3,4,5,6,7)

quality <- function(x, y) {
  x <- as.vector(x)
  y <- as.vector(y)
  i=0
  j=0
    for(n in 1:length(x)){
      i=i+abs(x[n]-y[n])
      j=j+abs(y[n])
    }
  print((i/j)*100)
}

quality(a, b)


## Testing linear significance
tx <- time(luna.ts)
lunatslm <- lm (luna.ts ~ cos(tx*2*w)+sin(tx*2*w))
summary(lunatslm)
lunatslm$coef[1]
#
#
# model1 <-  lm(luna.ts~cos(tx*2*w)+sin(tx*2*w))
# model2 <-  lm(luna.ts~ (0.002573513+cos(tx*2*w)+sin(tx*2*w)))
# anova(model1,model2)

## The temporal pseudoreplication command (ANOVA) did not work out in terms of my
code
print(mean(luna.ts)-lunatslm$coef[1])

##The intercept (average distance to the moon at 0 AD (x=0)?) is too small to consider)
```