Quiz2

Quiz2

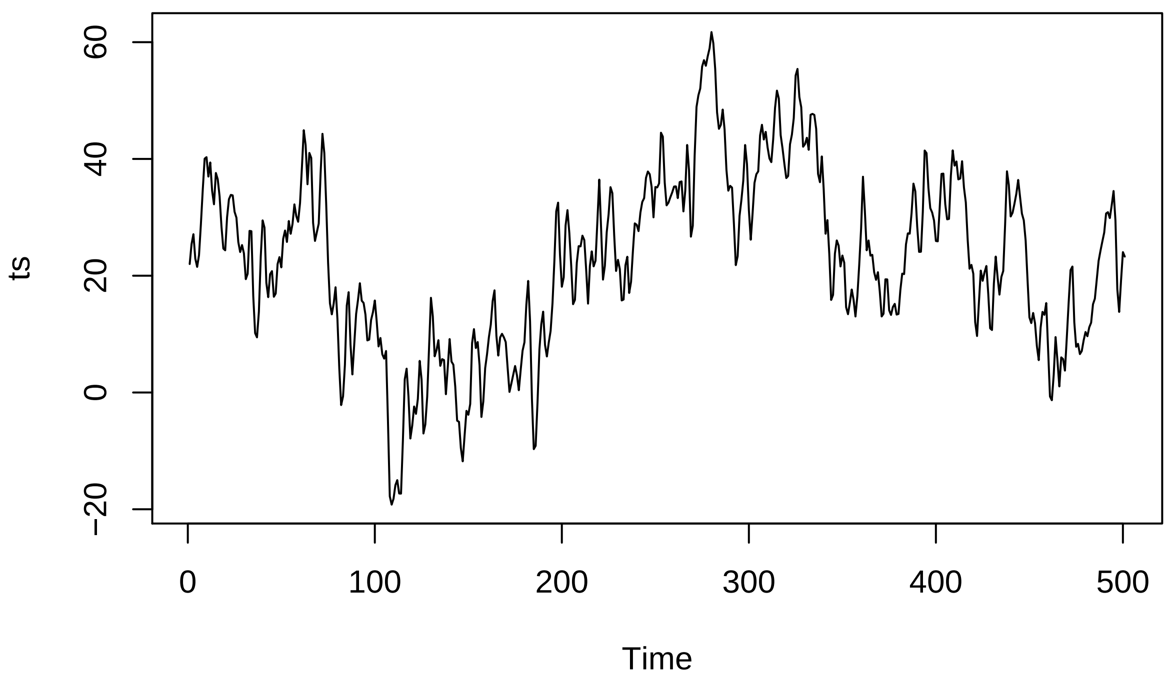
Q1

```r
set.seed(2254)
ts = arima.sim(n=500, model=list(order=c(2,1,2), ar=c(0.6,-0.2), ma=c(-0.7,-01)),sd=sqrt(6)) + 22
head(ts)
```

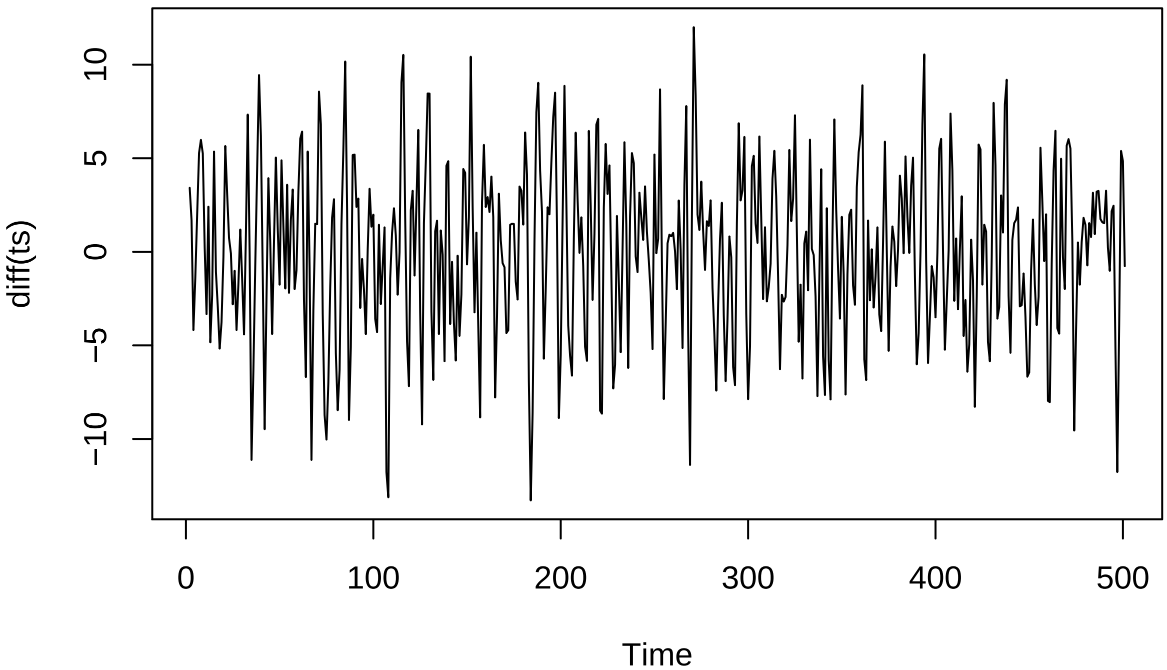
Part a

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
## [1] 22.00000 25.41987 27.10653 22.93072 21.51308 23.52951
```

```r
# part a
plot(ts) #model looks approximately stationary, but mean may move around
```




```r
plot(diff(ts)) #model looks much more stationary here, with constant mean
```




```
ndiffs(ts)# output of ndiffs suggests that model is stationary after differencing once,
```

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

```
#           confirming suspicions from looking at the plots
eacf(diff(ts)) #extended autocorrelation function plot suggests ARIMA(0,1,4)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x x x o o x o o x o  o  o  o
## 1 x x x x o o o o o x o  o  o  o
## 2 x x x o o o o o o x o  o  o  o
## 3 x x x x o o o o o x o  o  o  o
## 4 x x x x o o x o o o o  o  o  o
## 5 x o x o x o o o o o o  o  o  o
## 6 x x x o o x x o o o o  o  o  o
## 7 o x x o o x o x o o o  o  o  o
```

```
auto.arima(ts) #auto arima agrees with our predictions using eacf
```

```
## Series: ts
## ARIMA(0,1,4)
##
## Coefficients:
##          ma1      ma2      ma3      ma4
##       0.6582  -0.3349  -0.2982  -0.0999
## s.e.  0.0444   0.0516   0.0521   0.0433
##
## sigma^2 estimated as 12.38:  log likelihood=-1336.98
## AIC=2683.96   AICc=2684.08   BIC=2705.03
```

```
#ML estimation
mod_ML = auto.arima(ts, method='ML')
```

```r
#CLS estimation
mod_CLS = auto.arima(ts, method='CSS')
#UCLS estimation
mod_UCL = auto.arima(ts, method='CSS-ML')

mod_ML
```

Part b

```
## Series: ts
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2     ma1      ma2
##       0.5665  -0.1719  0.0957  -0.5332
## s.e.  0.0984   0.0659  0.0943   0.0909
##
## sigma^2 estimated as 12.36:  log likelihood=-1336.68
## AIC=2683.36   AICc=2683.49   BIC=2704.44
```

```r
mod_CLS
```

```
## Series: ts
## ARIMA(0,1,4)
##
## Coefficients:
##          ma1      ma2      ma3      ma4
##       0.6596  -0.3360  -0.2994  -0.1005
## s.e.  0.0444   0.0515   0.0522   0.0433
##
## sigma^2 estimated as 12.38:  part log likelihood=-1336.43
```

```r
mod_UCL
```

```
## Series: ts
## ARIMA(2,1,2)
##
## Coefficients:
##          ar1      ar2     ma1      ma2
##       0.5665  -0.1719  0.0957  -0.5332
## s.e.  0.0984   0.0659  0.0943   0.0909
##
## sigma^2 estimated as 12.36:  log likelihood=-1336.68
## AIC=2683.36   AICc=2683.49   BIC=2704.44
```

```r
#Surprisingly, in this case, CLS seems to give the parameters with the best p-values
```

Q4

```r
getSymbols('GOOGL', from='2020-04-29', to='2021-04-29')
```

Part a

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
```

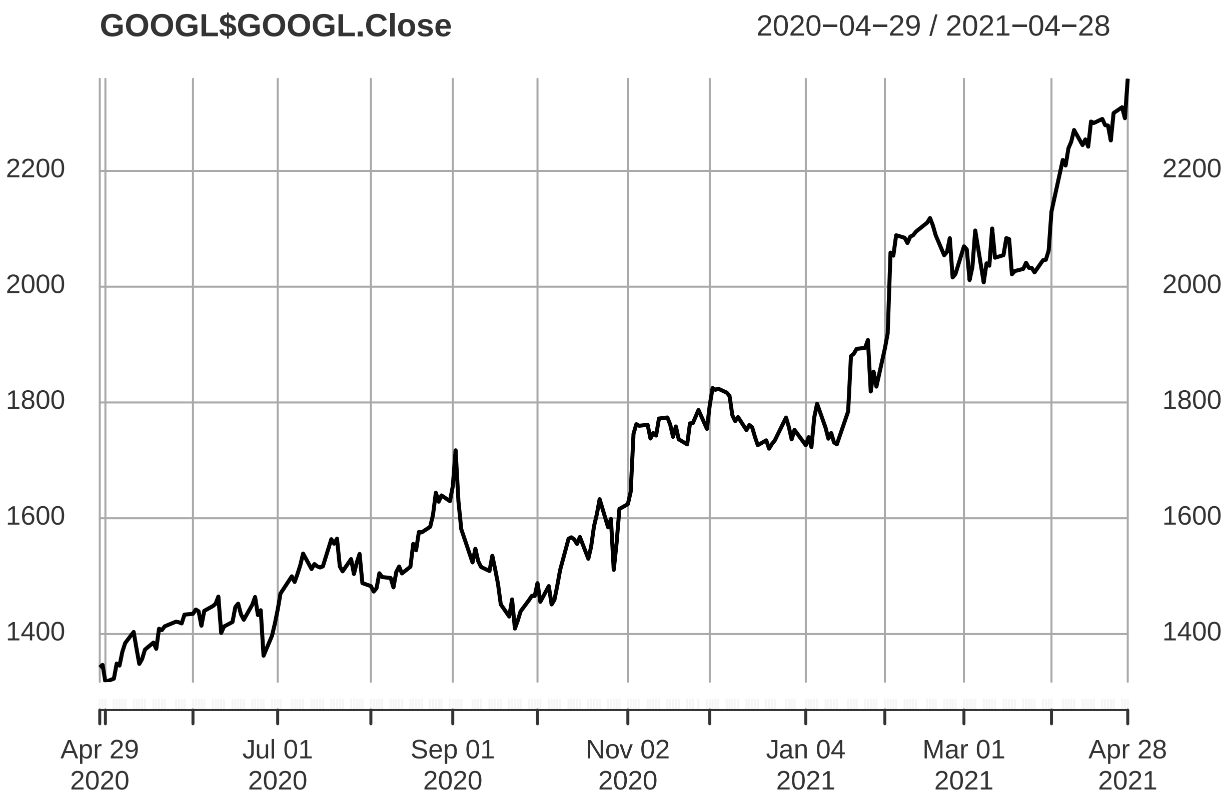
```
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "GOOGL"
```

```
names(GOOGL)
```

```
## [1] "GOOGL.Open"     "GOOGL.High"     "GOOGL.Low"      "GOOGL.Close"
## [5] "GOOGL.Volume"   "GOOGL.Adjusted"
```

```
plot(GOOGL$GOOGL.Close)
```




```
ts = GOOGL$GOOGL.Close
#Data is not stationary, with an increasing mean. Sudden jumps in stock price
#indicate that autocorrelation might not be only dependent on distance between points,
#but the entire trend looks approximately linear, so the only violation of stationarity

adf.test(ts)
```

Part b

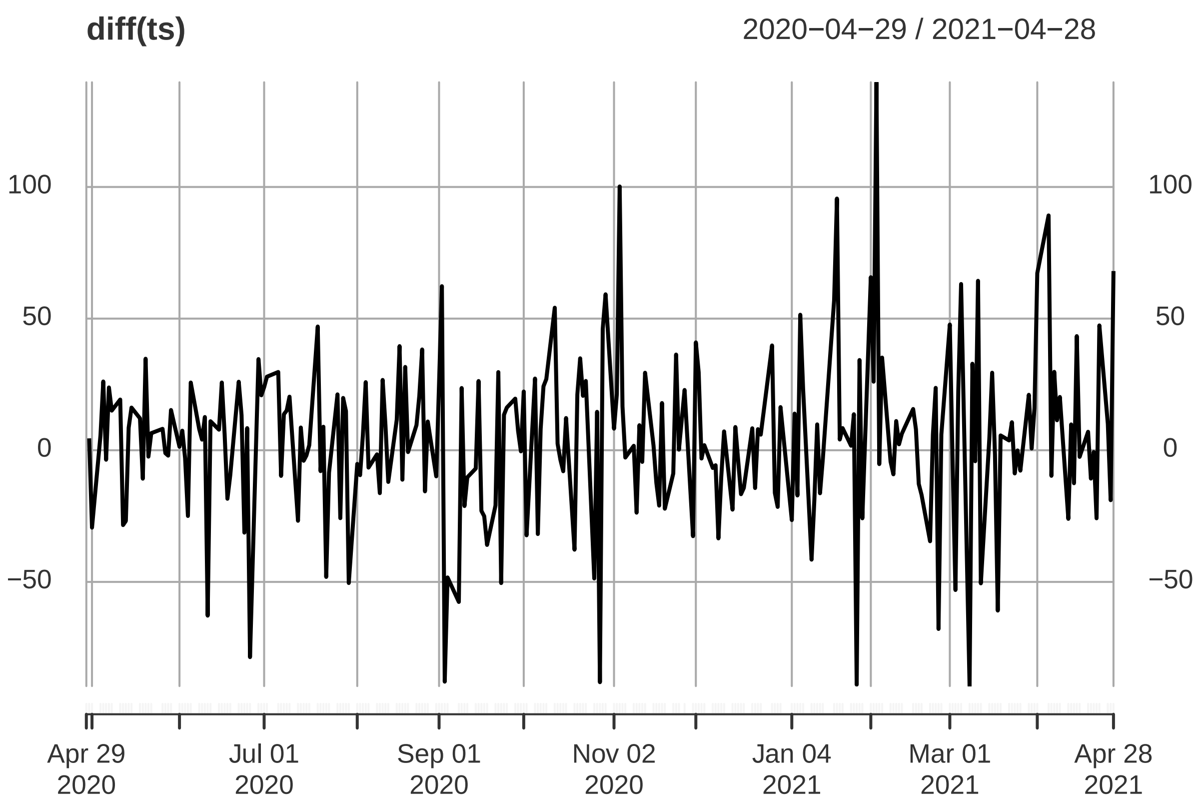
```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 2.38    0.99
```

```
## [2,]   1 2.54    0.99
## [3,]   2 2.46    0.99
## [4,]   3 2.57    0.99
## [5,]   4 2.35    0.99
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 2.01    0.99
## [2,]   1 2.16    0.99
## [3,]   2 2.00    0.99
## [4,]   3 2.11    0.99
## [5,]   4 2.07    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 3.03    0.99
## [2,]   1 3.28    0.99
## [3,]   2 3.23    0.99
## [4,]   3 3.43    0.99
## [5,]   4 3.25    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
#adf test strongly indicates that the data is NOT stationary.
ndiffs(ts)
```

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

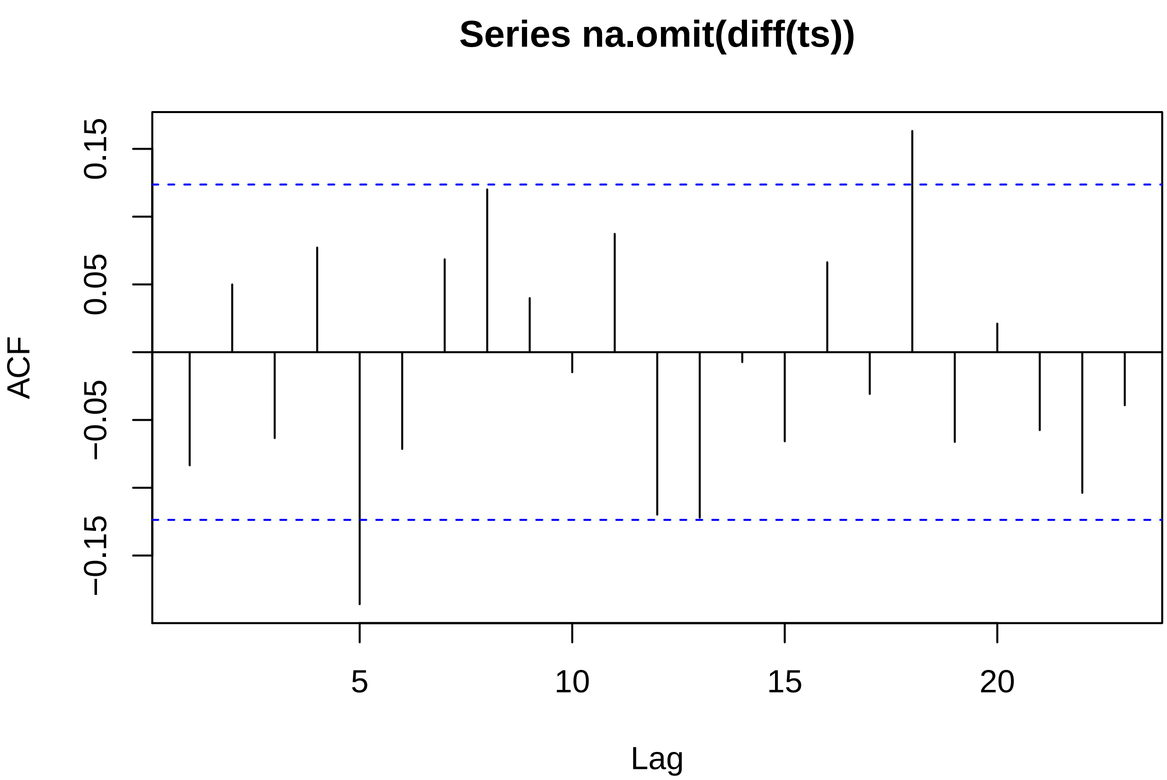
```
plot(diff(ts))
```



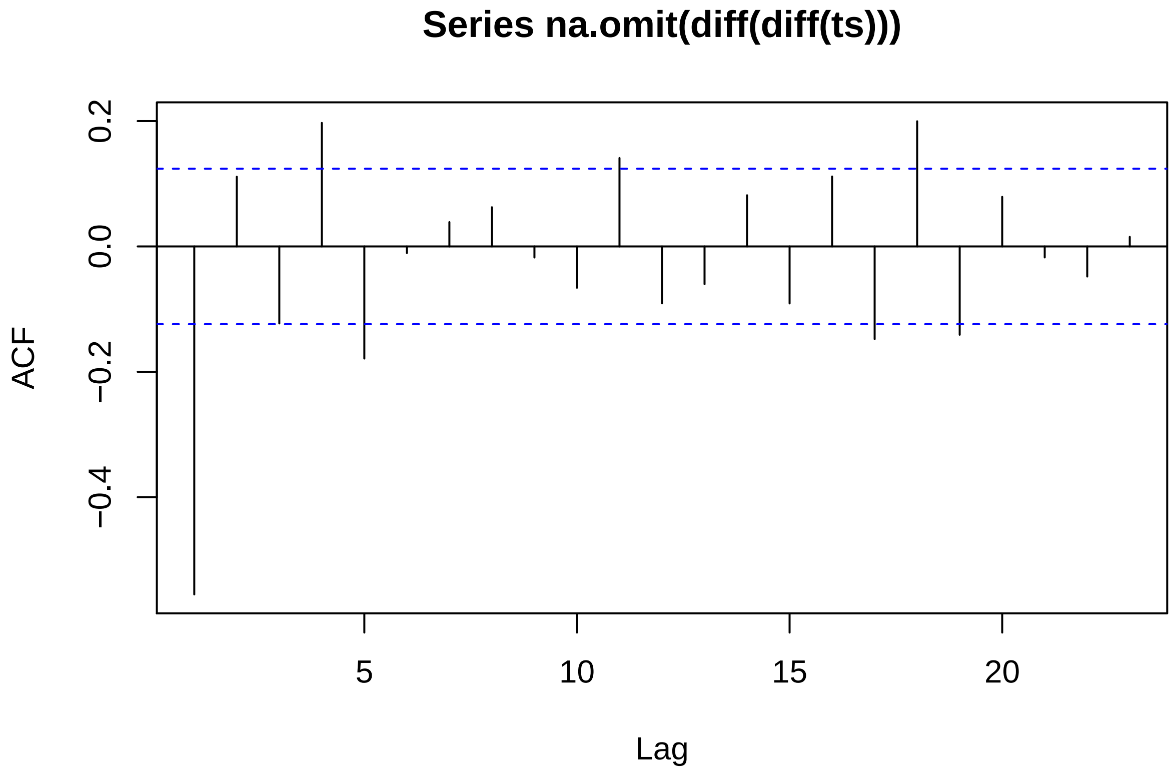

```
#data now looks approximatley stationary, but still has large jumps
adf.test(na.omit(diff(diff(ts))))
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag  ADF p.value
## [1,]   0 29.8    0.99
## [2,]   1 45.7    0.99
## [3,]   2 66.8    0.99
## [4,]   3 78.2    0.99
## [5,]   4 93.8    0.99
## Type 2: with drift no trend
##      lag  ADF p.value
## [1,]   0 29.7    0.99
## [2,]   1 45.6    0.99
## [3,]   2 66.7    0.99
## [4,]   3 78.1    0.99
## [5,]   4 93.6    0.99
## Type 3: with drift and trend
##      lag  ADF p.value
## [1,]   0 29.7    0.99
## [2,]   1 45.5    0.99
## [3,]   2 66.5    0.99
## [4,]   3 77.9    0.99
## [5,]   4 93.4    0.99
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

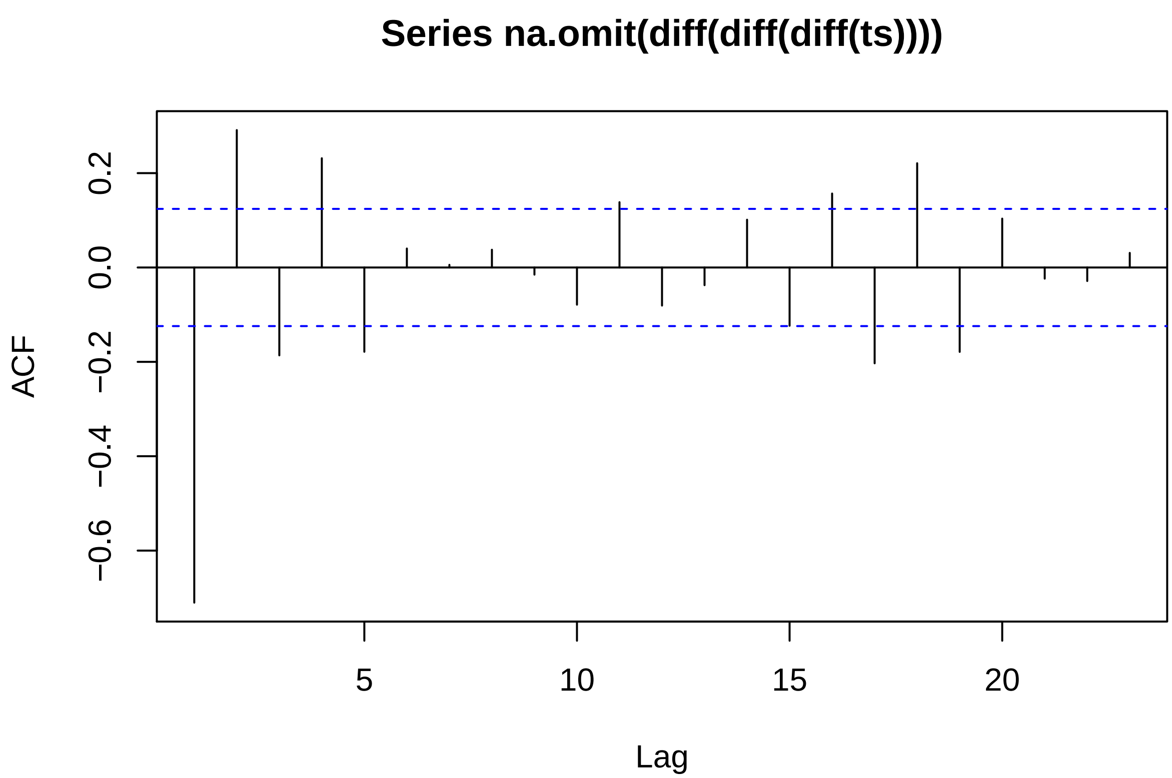
```
#adf still strongly suggests data is not stationary
acf(na.omit(diff(ts)))
```



```
acf(na.omit(diff(diff(ts))))
```




```
acf(na.omit(diff(diff(diff(ts)))))
```



```
#taking increasing differences shows that data continues to have non-zero acf
#values for large lags. Again, this is a sign of an explosivley non-stationary dataset
#however, after taking one difference, data appears nearly stationary, and might
#be suitably approximated by a ARIMA(p,1,q) model
```

```
eacf(na.omit(diff(ts)))
```

Part C

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 o o o o x o o o o o o  o  o  o
## 1 x o o o x x o o o o o  o  o  o
## 2 x o o o x o o o o o o  o  x  o
## 3 x o o o x o o o o o o  o  x  o
## 4 x o o x x o o o o o o  o  x  o
## 5 x x o x o o o o o o o  o  o  o
## 6 x x o x o x o o o o o  o  o  o
## 7 x x x x o x o o o o o  o  o  o
```

```
#extended autocorrelation function suggests ARIMA(0,1,1)
auto.arima(ts)
```

```
## Series: ts
## ARIMA(0,1,0) with drift
##
## Coefficients:
##        drift
##       4.0512
## s.e.  1.9415
##
## sigma^2 estimated as 949.9:  log likelihood=-1216.13
## AIC=2436.26   AICc=2436.31   BIC=2443.31
```

```
#auto arima finds a model ARIMA(0,1,0) with drift
# , which matches the appearance of the plot for a single difference

names(arima(ts, order=c(0,1,1)))
```

```
##  [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
##  [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
## [13] "nobs"      "model"
```

```
mod1 = arima(ts, order=c(0,1,1))
mod2 = auto.arima(ts)
mean(mod1$residuals)
```

```
## [1] 4.280061
```

```
mean(mod2$residuals)
```

```
## [1] 0.005310032
```

```
#the auto arima model seems to be best
mod = auto.arima(ts)
```

```r
mod = auto.arima(ts, method='ML')
```

Part D

```r
Box.test(mod$residuals, type=c('Box-Pierce'))
```

Part e

```
##
##  Box-Pierce test
##
## data:  mod$residuals
## X-squared = 0.0096482, df = 1, p-value = 0.9218
```

```r
Box.test(mod$residuals, type=c('Ljung-Box'))
```

```
##
##  Box-Ljung test
##
## data:  mod$residuals
## X-squared = 0.0097635, df = 1, p-value = 0.9213
```

```r
#Both tests fail to reject, thus residuals are IID and the model is an acceptable fit.
```