Homework 2

Yeshashvi Munagala

1/26/2022

Questions 1 and 3 are attached at the end of the pdf!

2a, 2b)

```
#library(TSstudio)
library(itsmr)
library(forecast)
```

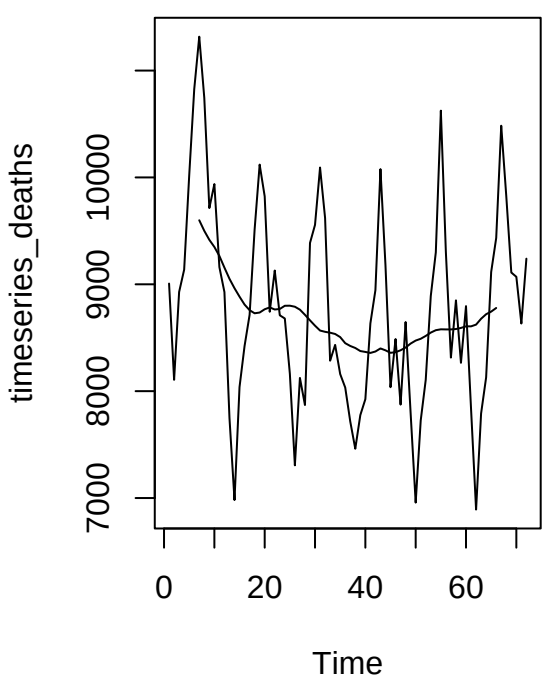
```
## Warning: package 'forecast' was built under R version 4.1.1
```

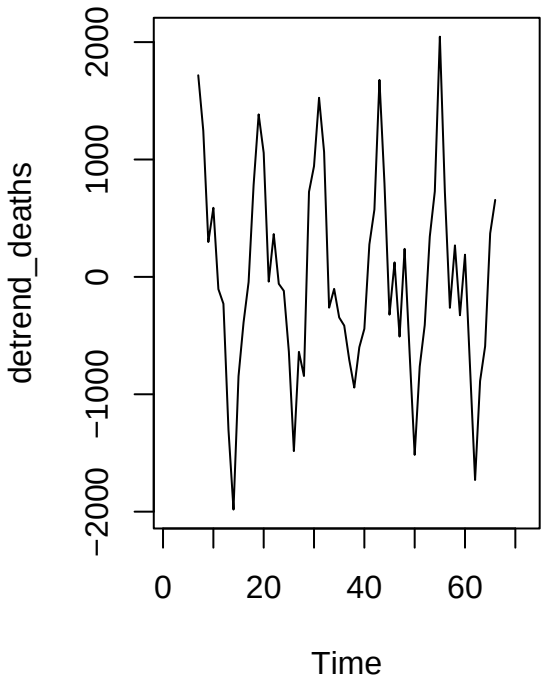
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

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

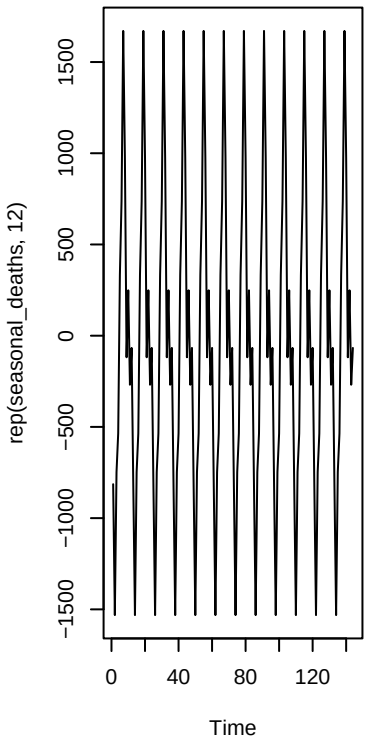
```
## The following object is masked from 'package:itsmr':
##
##     forecast
```

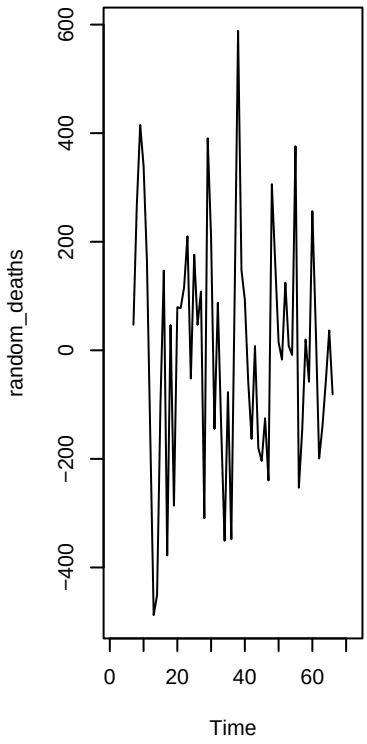
```
timeseries_deaths = tail(head(deaths, 17*12+2),17*12-4)
trend_deaths = ma(timeseries_deaths, order = 12, centre = T)
detrend_deaths = timeseries_deaths - trend_deaths
par(mfrow = c(1, 2))
plot.ts(timeseries_deaths)
lines(trend_deaths)
plot.ts(detrend_deaths)
```

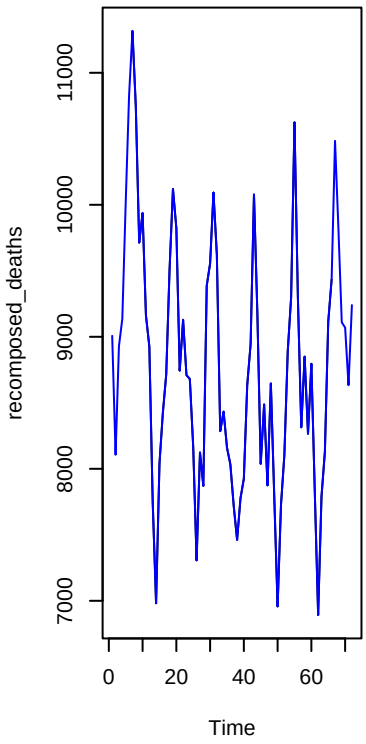




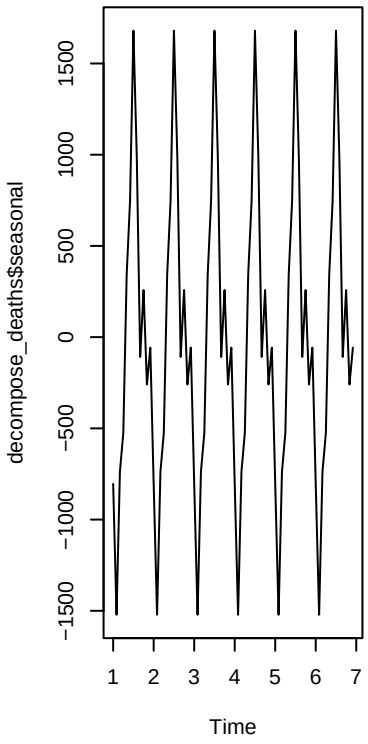

```
m_deaths = t(matrix(data = detrend_deaths, nrow = 12))
#m_deaths

seasonal_deaths = colMeans(m_deaths, na.rm = T)
random_deaths = timeseries_deaths - trend_deaths - seasonal_deaths
recomposed_deaths = trend_deaths + seasonal_deaths + random_deaths
par(mfrow = c(1, 3))
plot.ts(rep(seasonal_deaths,12))
plot.ts(random_deaths)
plot.ts(recomposed_deaths)
lines(timeseries_deaths,col="blue")
```

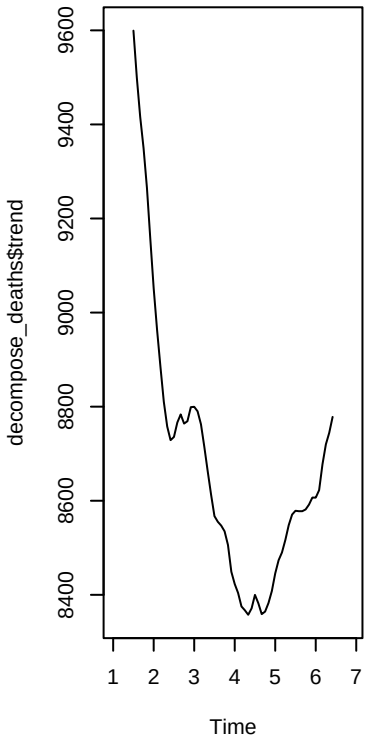


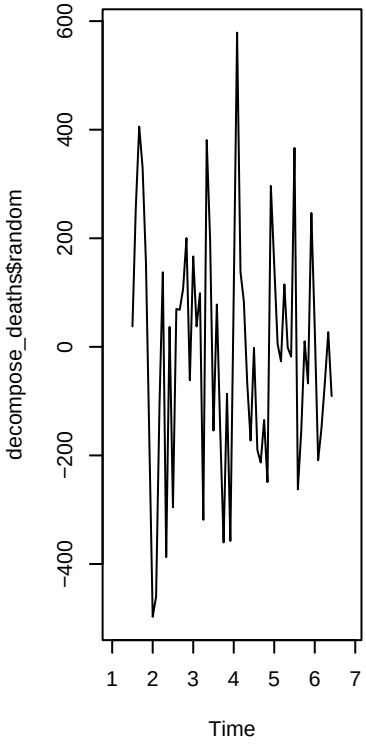




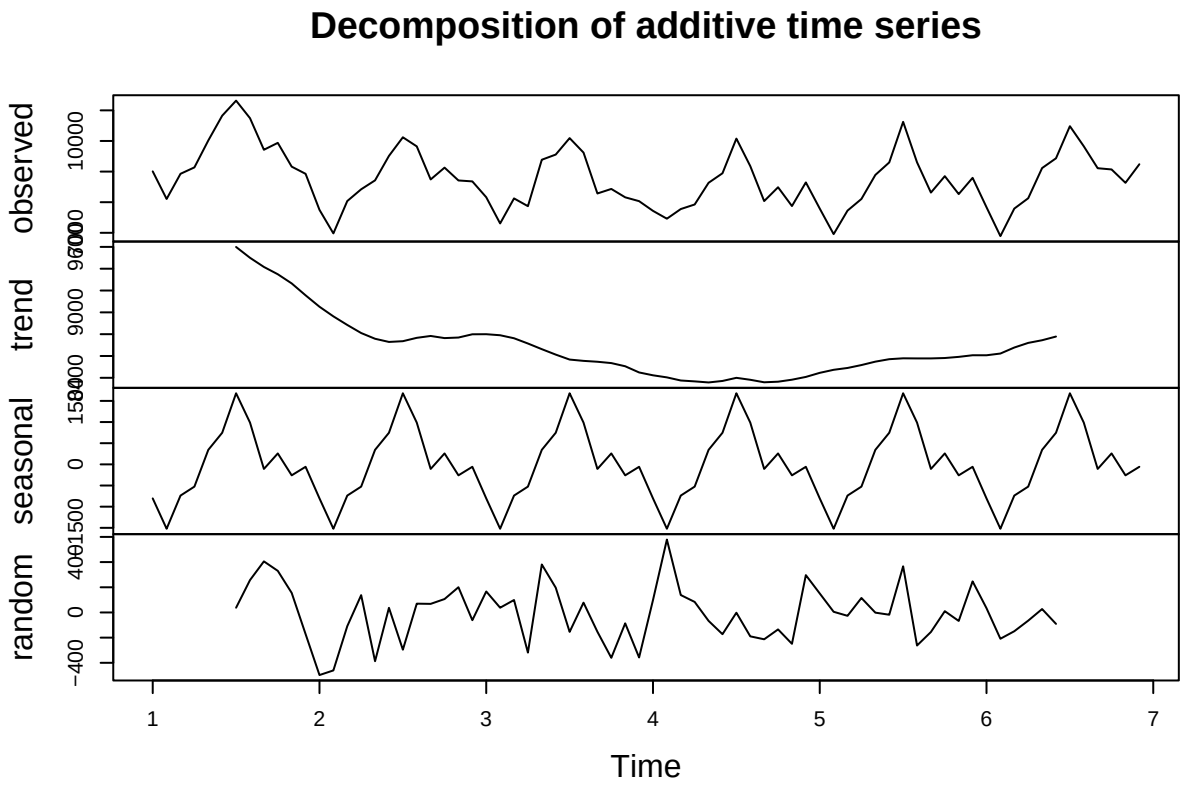

```
ts_deaths = ts(timeseries_deaths, frequency = 12)
decompose_deaths = decompose(ts_deaths, "additive")
plot.ts(decompose_deaths$seasonal)
plot.ts(decompose_deaths$trend)
plot.ts(decompose_deaths$random)
```







```
plot(decompose_deaths)
```



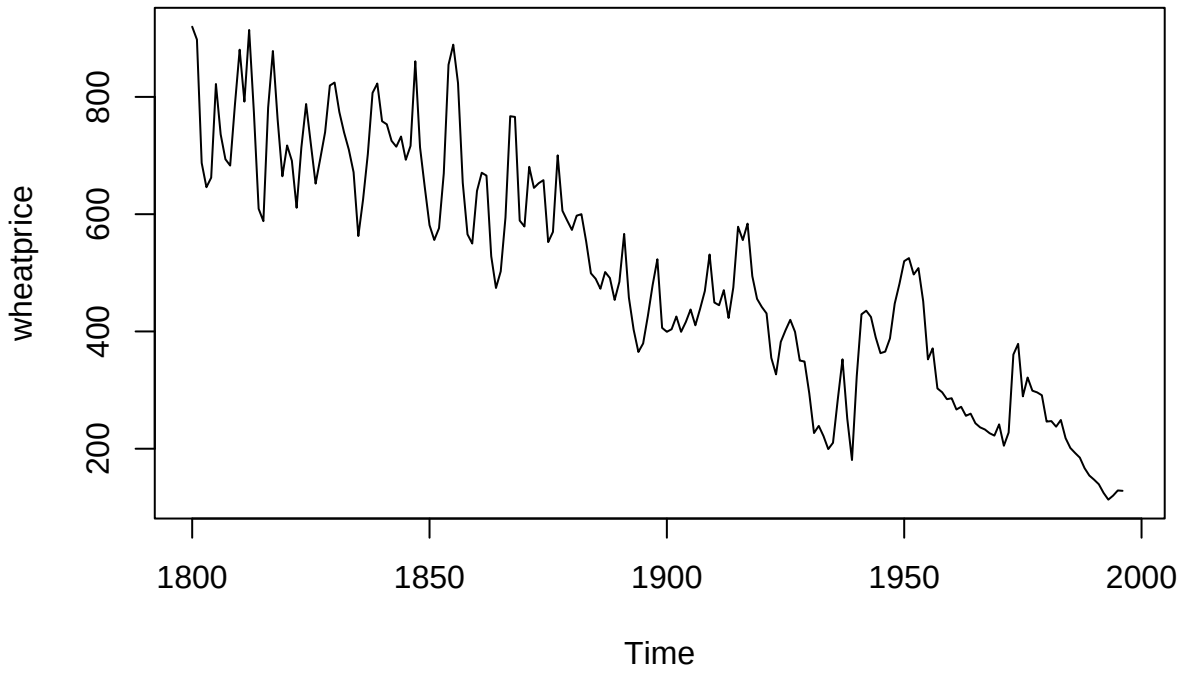
4a)

```
library(fpp3)
```

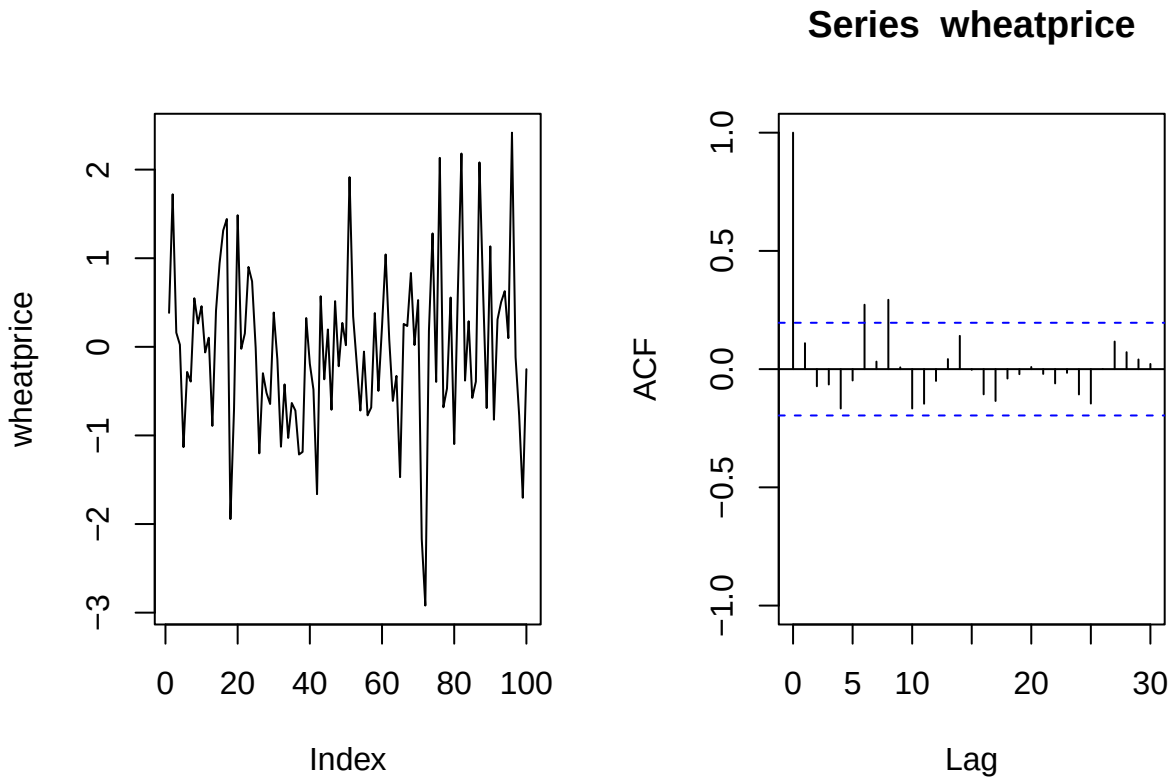
```
## -- Attaching packages --------------------------------------- fpp3 0.4.0 --
## v tibble      3.1.2     v tsibble     1.1.1
## v dplyr       1.0.7     v tsibbledata 0.4.0
## v tidyr       1.1.3     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5
## Warning: package 'lubridate' was built under R version 4.1.1
## Warning: package 'tsibble' was built under R version 4.1.1
## Warning: package 'tsibbledata' was built under R version 4.1.1
## Warning: package 'feasts' was built under R version 4.1.1
## Warning: package 'fabletools' was built under R version 4.1.1
## Warning: package 'fable' was built under R version 4.1.1
## -- Conflicts ------------------------------------------ fpp3_conflicts --
## x lubridate::date()      masks base::date()
## x dplyr::filter()        masks stats::filter()
## x fabletools::forecast() masks forecast::forecast(), itsmr::forecast()
## x tsibble::intersect()   masks base::intersect()
## x tsibble::interval()    masks lubridate::interval()
```

```
## x dplyr::lag()           masks stats::lag()
## x tsibble::setdiff()     masks base::setdiff()
## x tsibble::union()       masks base::union()
```

```
wheatprice <- ts(prices$wheat, start = 1800, end = 1997,frequency = 1)
plot.ts(wheatprice)
```




```
wheatprice <- rnorm(100, mean=0, sd=1)
par(mfrow = c(1, 2))
plot(wheatprice, type="l")
acf(wheatprice, lag.max=30, ylim=c(-1, 1))
```



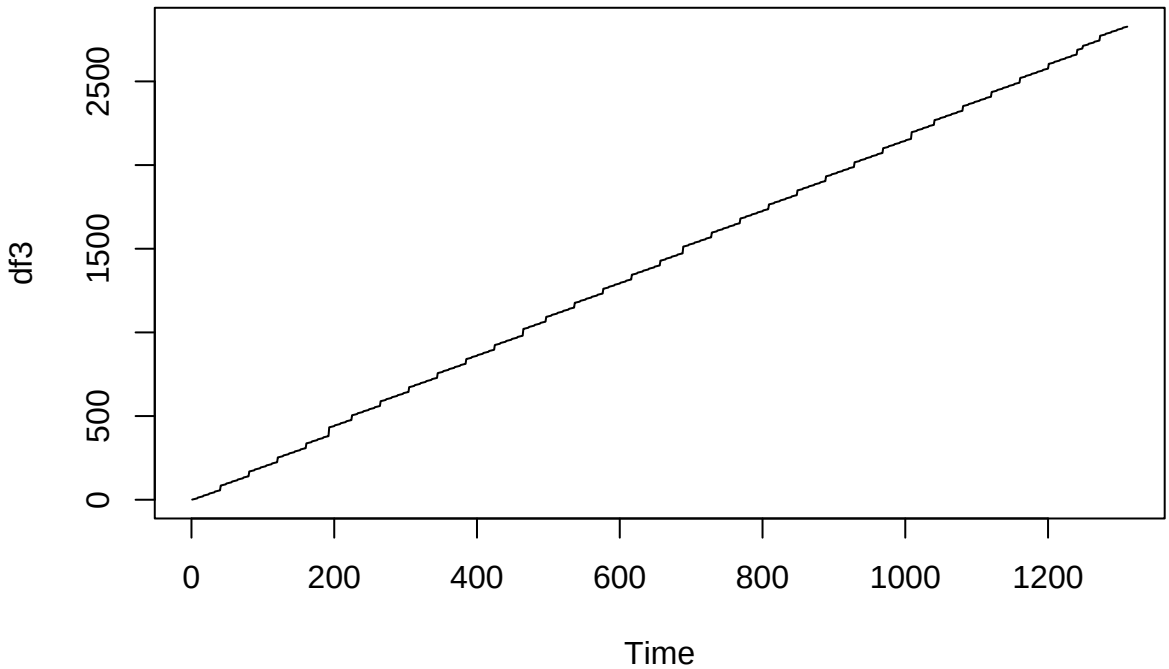

```
# There is clearly a decreasing trend in this time series. The model does a good job af representing th
```

4b)

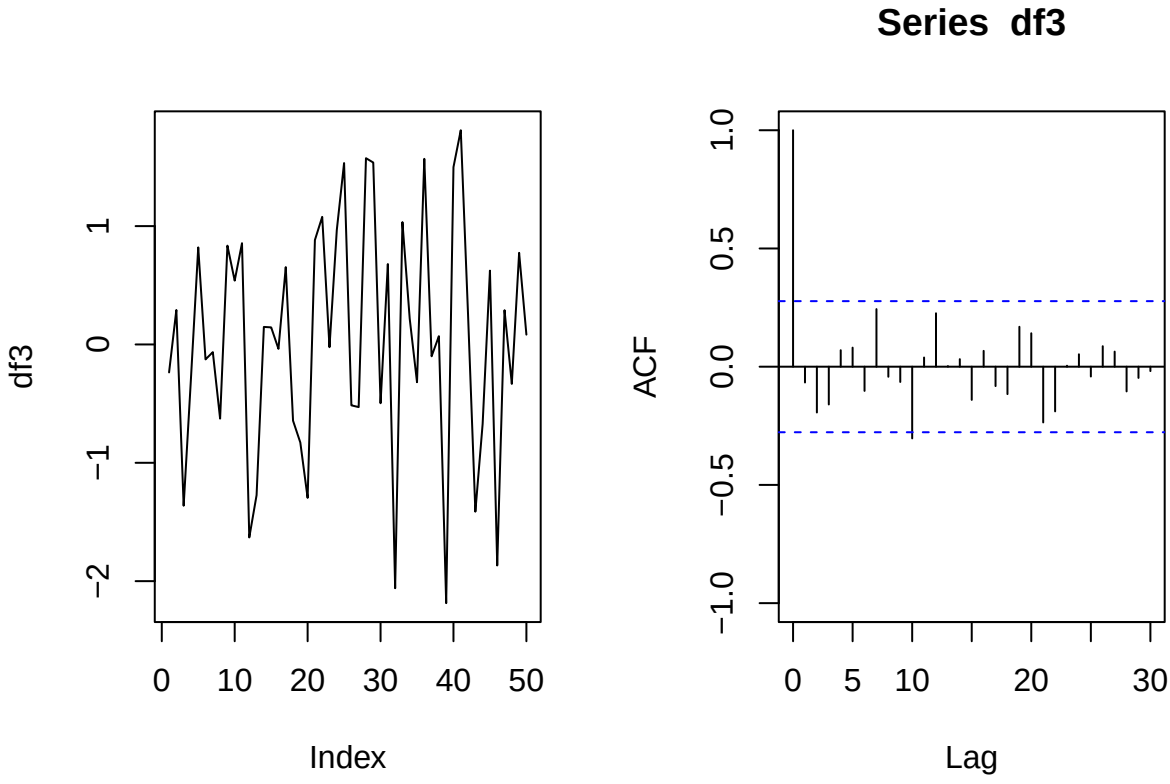
```
df <- data.frame(bank_calls)

dt <- seq(
  from = as.POSIXct("2003-03-03 07:00:00", tz = "GMT"),
  to = as.POSIXct("2003-10-24 21:00:00", tz = "GMT"),
  by = "7200 sec")

binned_dataset = bank_calls %>% mutate(BinnedTime = cut(DateTime, breaks=dt)) %>%
  group_by(BinnedTime,add=TRUE)
```

```
## Warning: The `add` argument of `group_by()` is deprecated as of dplyr 1.0.0.
## Please use the `.add` argument instead.
```

```
df2 <- data.frame(binned_dataset$Calls, binned_dataset$BinnedTime)
colnames(df2) <- c('Calls','BinnedTime')

df2 <- aggregate(df2$Calls,
                 by = list(df2$BinnedTime),
                 FUN = sum)
colnames(df2) <- c('DateTime','Aggregate Calls')

df3 <- df2[c(1:50),]
df3 <- ts(df2$DateTime)
plot.ts(df3)
```

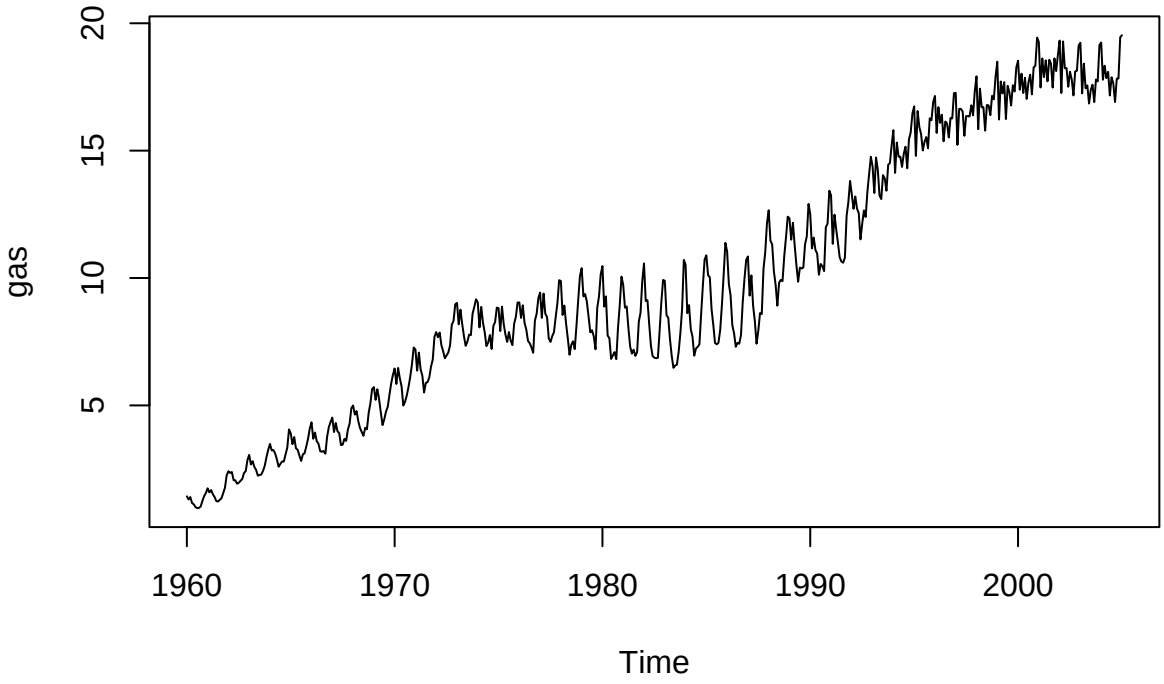



```
#df2$DateTime <- as.datetime(df2$DateTime,'%Y-%m-%d %H:%M:%S')

df3 <- rnorm(50, mean=0, sd=1)
par(mfrow = c(1, 2))
plot(df3, type="l")
acf(df3, lag.max=30, ylim=c(-1, 1))
```

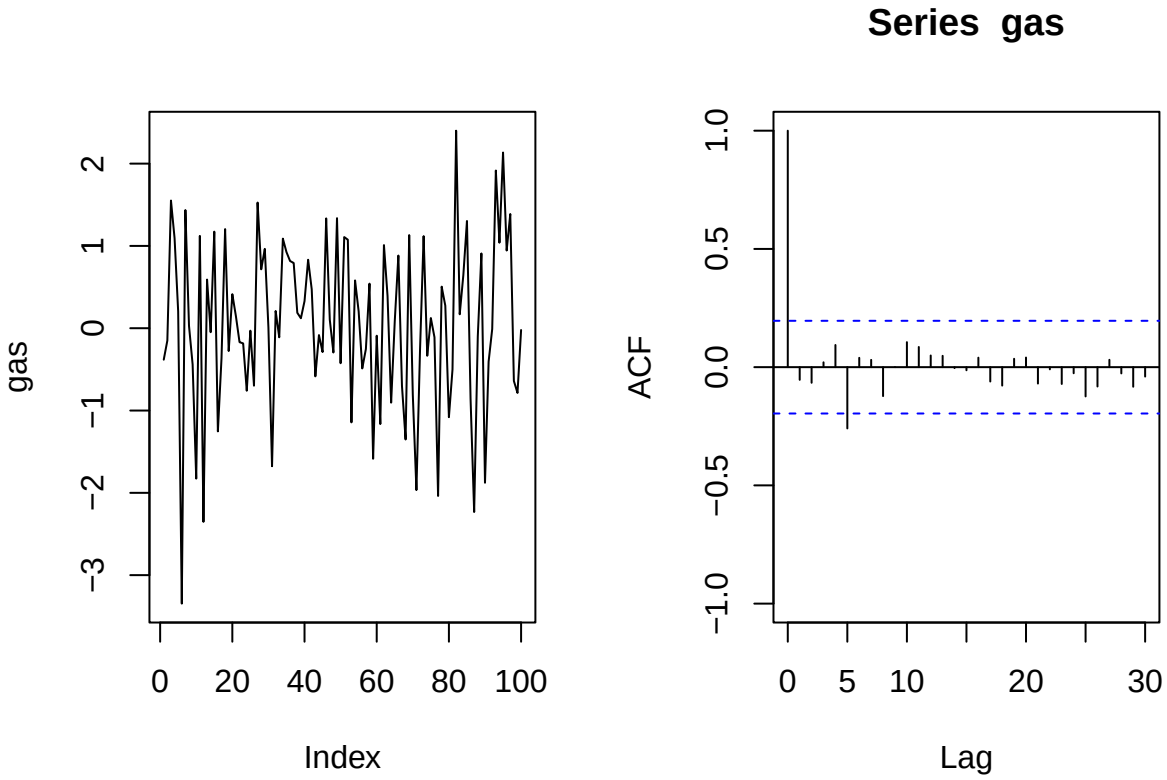


4c)

```
canadian_gas$Month <- as.Date(canadian_gas$Month)
gas <- ts(canadian_gas$Volume, start = 1960, end = 2005, frequency = 12)
plot.ts(gas)
```




```
gas <- rnorm(100, mean=0, sd=1)
par(mfrow = c(1, 2))
plot(gas, type="l")
acf(gas, lag.max=30, ylim=c(-1, 1))
```




```
# Here, I expected the correlogram to exhibits peaks at multiples of 12 because
#this data contains monthly data; however, that is not the case. Based on
#autocorrelation, it seems as though the model does a good job representing the
#data set, as the lines in the ACF plot are mostly between the dashed blue lines.
```