San Diego's Ocean Water Monitoring Program

Group 3

11/14/2021

```r
# R Libraries
library(astsa)
library(RCurl)
library(psych)
library(dplyr)
library(RSQLite)
library(naniar)
library(ggplot2)
library(forecast)
```

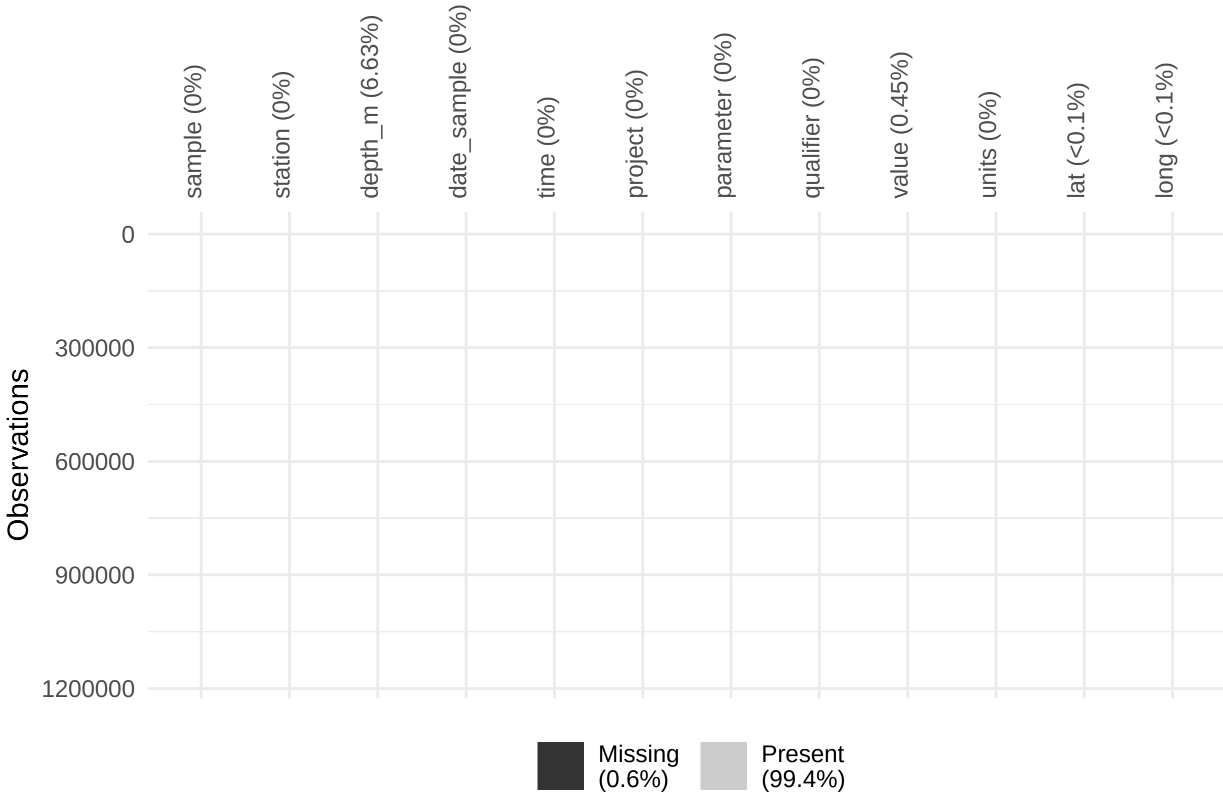
```r
# Datasets (3 CSV files)
csv1 <- getURL("http://seshat.datasd.org/pud/omp/water_quality_2011_2019_datasd.csv")
csv2 <- getURL("http://seshat.datasd.org/pud/omp/water_quality_2000_2010_datasd.csv")
csv3 <- getURL("http://seshat.datasd.org/pud/omp/water_quality_1990_1999_datasd.csv")
stationsCSV <- getURL("http://seshat.datasd.org/pud/omp/reference_stations_water_quality.csv")
csv1dl <- read.csv( text = csv1 )
csv2dl <- read.csv( text = csv2 )
csv3dl <- read.csv( text = csv3 )
stationsdl <- read.csv( text = stationsCSV )
```

```r
# Bind data from 3 CSV files into one dataframe
csvs <- rbind(csv1dl, csv2dl, csv3dl)
# Bind stations data into dataframe
stations <- rbind(stationsdl)
```

```r
conn <- dbConnect(RSQLite::SQLite(), "ADS506.db") # to create a SQL database in memory
copy_to(conn,
        csvs, # load csvs dataframe into SQL
        overwrite = TRUE) # if exists, overwrite
copy_to(conn,
        stations, # load stations dataframe into SQL
        overwrite = TRUE) # if exists, overwrite
```

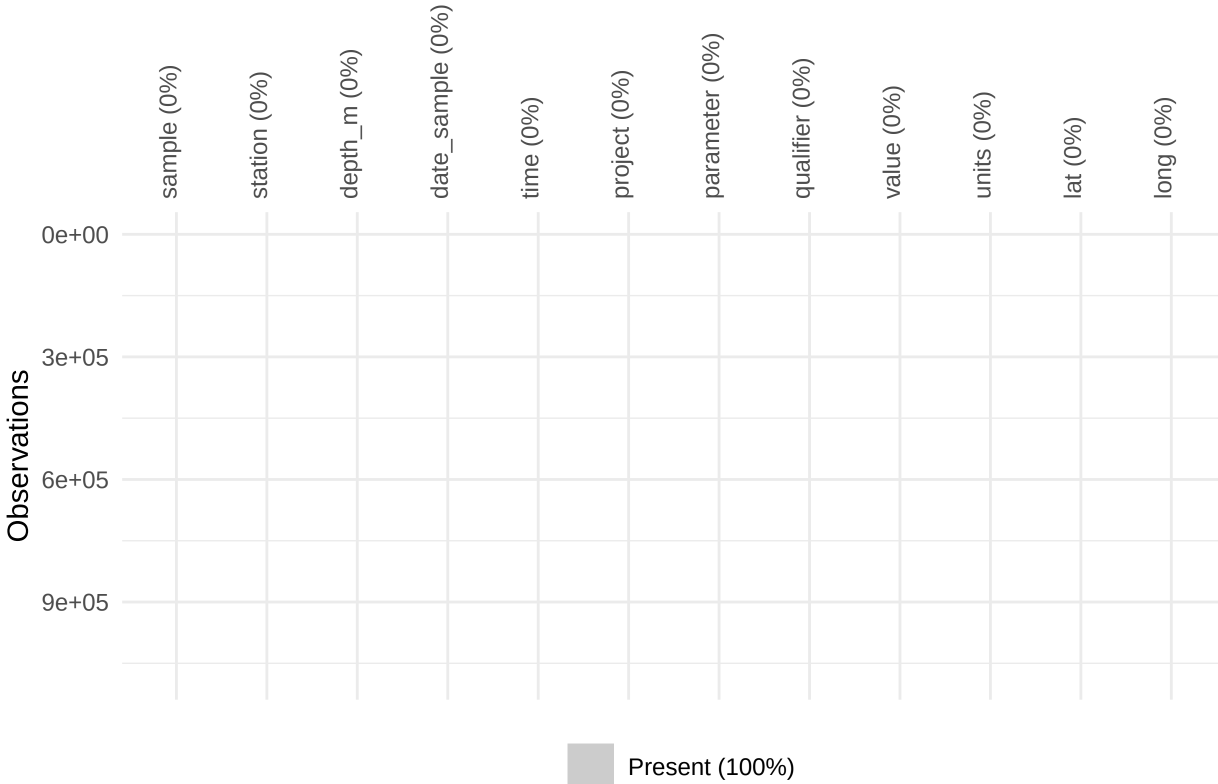
```r
# combine information from both dataframes into one
df <- dbGetQuery(conn, sql("
SELECT csvs.*, s.lat, s.long
FROM csvs
left join stations as s
on csvs.station = s.[ï..station]
"))
# disconnect from database
dbDisconnect(conn)
# clean up data that is no longer needed
rm(conn, csvs, csv1dl, csv2dl , csv3dl, csv1, csv2, csv3, stations, stationsdl, stationsCSV)
```

```r
# Plot missing data
vis_miss((df), warn_large_data = FALSE) +
  theme(axis.text.x = element_text(angle = 90))
```




```r
# remove all columns with missing data
df <- na.omit(df)
```

```r
# Plot missing data
vis_miss((df), warn_large_data = FALSE) +
  theme(axis.text.x = element_text(angle = 90))
```




```
# We are now left with 1,084,859 observations after dropping columns with missing data
str(df)
```

```
## 'data.frame':    1084859 obs. of  12 variables:
##  $ sample     : chr  "101111769" "101111770" "101111771" "101111772" ...
##  $ station    : chr  "I25" "I25" "I25" "I26" ...
##  $ depth_m    : num  2 6 9 6 9 6 9 2 2 2 ...
##  $ date_sample: chr  "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
##  $ time       : chr  "11:54:00 PST" "11:54:00 PST" "11:54:00 PST" "12:04:00 PST" ...
##  $ project    : chr  "SBOO" "SBOO" "SBOO" "SBOO" ...
##  $ parameter  : chr  "ENTERO" "ENTERO" "ENTERO" "ENTERO" ...
##  $ qualifier  : chr  "e" "" "" "" ...
##  $ value      : num  24 110 100 94 400 ...
##  $ units      : chr  "CFU/100 mL" "CFU/100 mL" "CFU/100 mL" "CFU/100 mL" ...
##  $ lat        : num  32.6 32.6 32.6 32.6 32.6 ...
##  $ long       : num  -117 -117 -117 -117 -117 ...
##  - attr(*, "na.action")= 'omit' Named int [1:81747] 6 7 8 9 10 11 12 13 14 15 ...
##   ..- attr(*, "names")= chr [1:81747] "6" "7" "8" "9" ...
```

```
# convert data_sample variable from "character" to "date" data type
df$date_sample <- as.Date(df$date_sample)
```

```
# confirm date_sample is now Date format
df %>% select(date_sample) %>% str()
```

```
## 'data.frame':    1084859 obs. of  1 variable:
##  $ date_sample: Date, format: "2011-01-01" "2011-01-01" ...
```

```
# Add new variable from date_sample variable, comprised of "Month_Yr"
df$sample_month_yr <- format(as.Date(df$date_sample), "%Y-%m")
```

```
# Sample 5 rows of date_sample and new variable side by side, to confirm new variable creation
df %>% select(date_sample, sample_month_yr) %>% sample_n(5)
```

```
##   date_sample sample_month_yr
## 1  2006-12-07         2006-12
## 2  2012-01-03         2012-01
## 3  2018-02-13         2018-02
## 4  2009-06-11         2009-06
## 5  2005-07-26         2005-07
```

```
# Variable "project" contains two values, PLOO (Point Loma) and SBOO (South Bay).
# There is a disproportionate split in the data between both "project"s.
df %>% group_by(project) %>% summarise(n=n()) %>% mutate(freq = n / sum(n))
```

```
## # A tibble: 2 x 3
##   project      n  freq
##   <chr>    <int> <dbl>
## 1 PLOO    727012 0.670
## 2 SBOO    357847 0.330
```

```
# there are several "unit" types in the data.  We are going to select the "unit" type with
# the highest representation
df %>% group_by(units) %>% summarise(n=n()) %>% mutate(freq = n / sum(n))
```

```
## # A tibble: 8 x 3
##   units           n   freq
##   <chr>       <int>  <dbl>
## 1 %          132276 0.122
## 2 C          132401 0.122
## 3 CFU/100 mL 315481 0.291
## 4 mg/L       138229 0.127
## 5 pH         100919 0.0930
## 6 ppt        102675 0.0946
## 7 sigma-t     81612 0.0752
## 8 ug/L        81266 0.0749
```

```
# there are several "parameter" types in the data.  We are going to select the "parameter" type with
# the highest representation
df %>% group_by(parameter) %>% summarise(n=n()) %>% mutate(freq = n / sum(n))
```

```
## # A tibble: 12 x 3
##    parameter        n    freq
##    <chr>        <int>   <dbl>
##  1 CHLOROPHYLL  81266 0.0749
##  2 DENSITY      81612 0.0752
##  3 DO          102792 0.0948
##  4 ENTERO      108837 0.100
##  5 FECAL       103413 0.0953
##  6 OG            7922 0.00730
##  7 PH          100919 0.0930
##  8 SALINITY    102675 0.0946
##  9 SUSO         27515 0.0254
## 10 TEMP        132401 0.122
## 11 TOTAL       103231 0.0952
## 12 XMS         132276 0.122
```

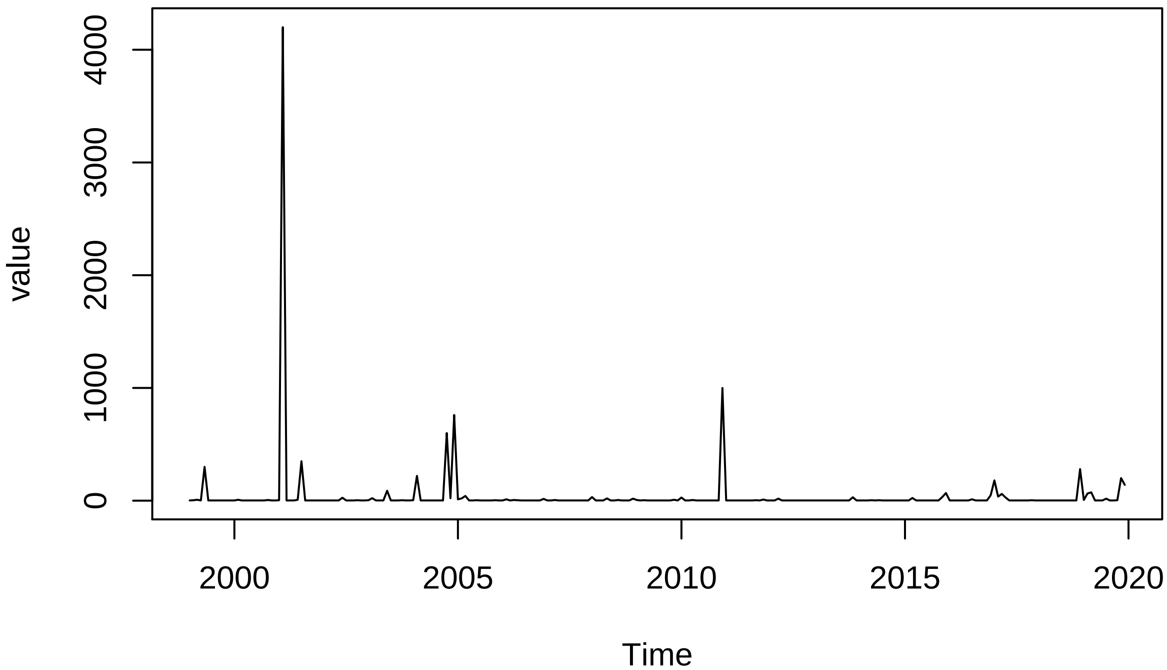
```r
# We will place each "project" into its respective dataframe, filtered for a specific unit and paramete
pl <- df %>% filter(project == "PLOO") %>% filter(units == "CFU/100 mL") %>% filter(parameter == "ENTERO
sb <- df %>% filter(project == "SBOO") %>% filter(units == "CFU/100 mL") %>% filter(parameter == "ENTERO
```

```r
# get month end value
sb_mth_end <- sb %>% group_by(sample_month_yr) %>%  do(tail(., n=1))
```

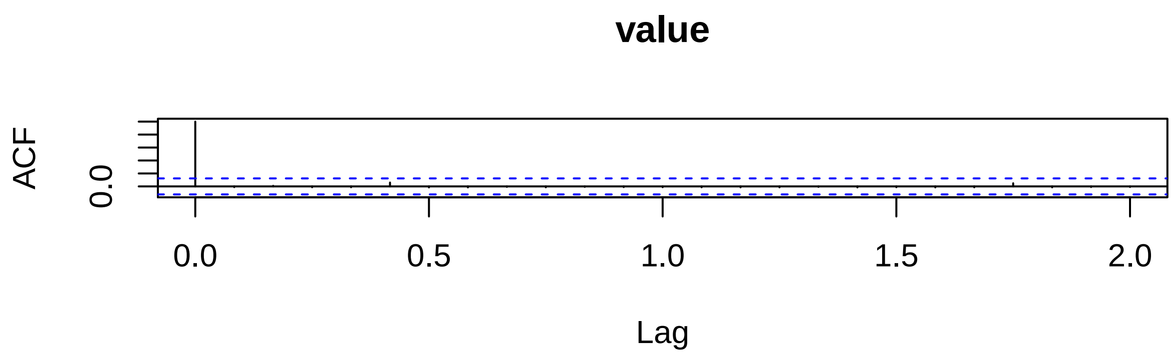
```r
# convert to TS
sb_mth_end.val <- sb_mth_end[c('value')]
df.ts <- ts(sb_mth_end.val, frequency=12, start=c(1999))
df.ts
```

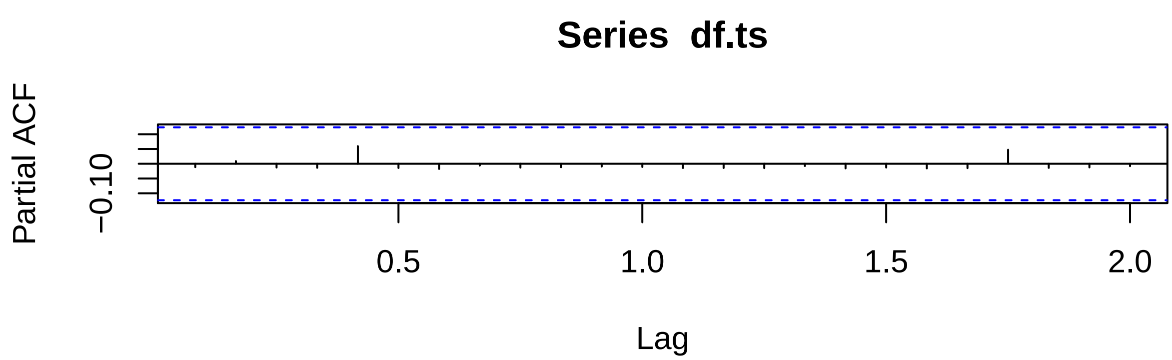
```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1999    2    4    8    2  300    2    2    2    2    2    2    2
## 2000    2    8    2    2    2    2    2    2    2    6    2    2
## 2001    4 4200    2    2    2    8  350    2    2    2    2    2
## 2002    2    2    2    2    2   26    2    2    2    4    2    2
## 2003    4   22    2    2    2   88    2    2    2    4    2    2
## 2004    4  220    2    2    2    2    2    2    2  600   22  760
## 2005   12   20   42    2    2    4    2    2    2    2    4    2
## 2006    2   12    2    6    4    2    2    2    2    2    2   16
## 2007    2    2    6    2    2    2    2    2    2    2    2    2
## 2008   32    2    2    2   20    2    2    6    2    2    2   18
## 2009    6    2    4    2    2    2    2    2    2    2    8    2
## 2010   28    2    2    6    2    2    2    2    2    2    2 1000
## 2011    2    2    2    2    2    2    2    2    4    2   10    2
## 2012    2    2   18    2    2    2    2    2    2    2    2    2
## 2013    2    2    2    2    2    2    2    2    2    2   30    2
## 2014    2    2    2    4    2    4    2    2    2    2    2    2
## 2015    2    2   24    2    2    2    2    2    2    2   32   68
## 2016    2    2    2    2    2    2   12    2    2    2    2   48
## 2017  180   36   60   28    2    2    2    2    2    2    4    2
## 2018    2    2    2    2    2    2    2    2    2    2    2  280
## 2019    6   64   74    2    2    2   18    2    2    4  200  140
```

```r
plot(df.ts)
```

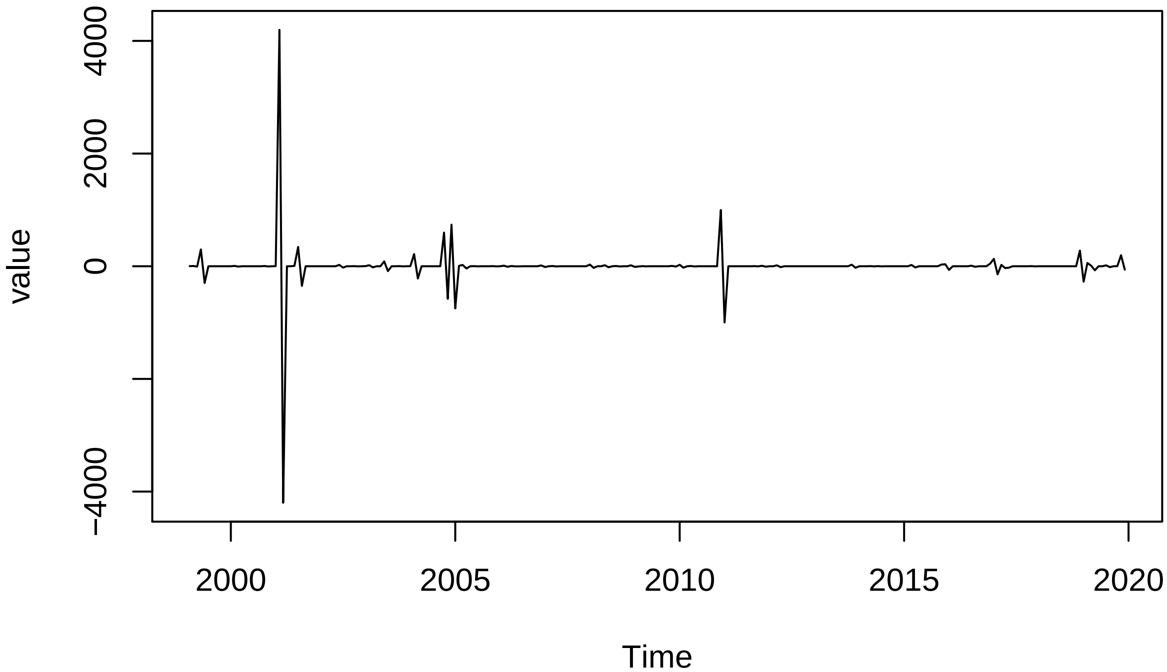



```
# ACF shows significant autocorrelation. observations are not independent.
par(mfrow = c(2, 1))
acf(df.ts)
pacf(df.ts)
```

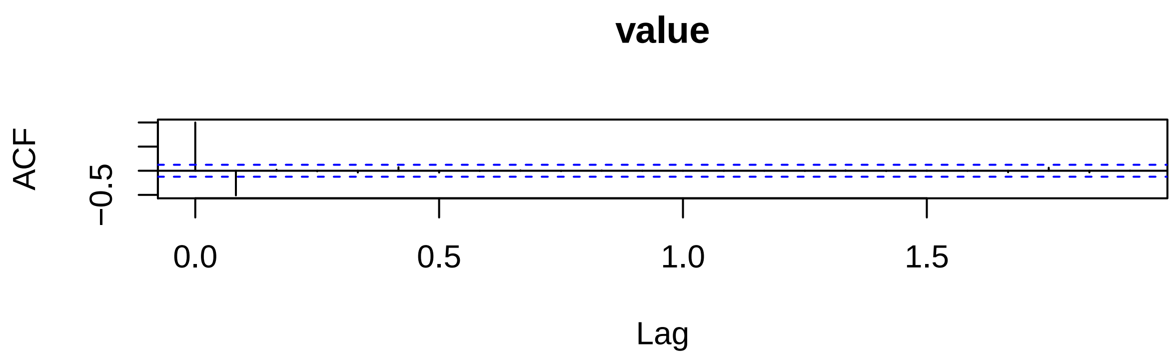


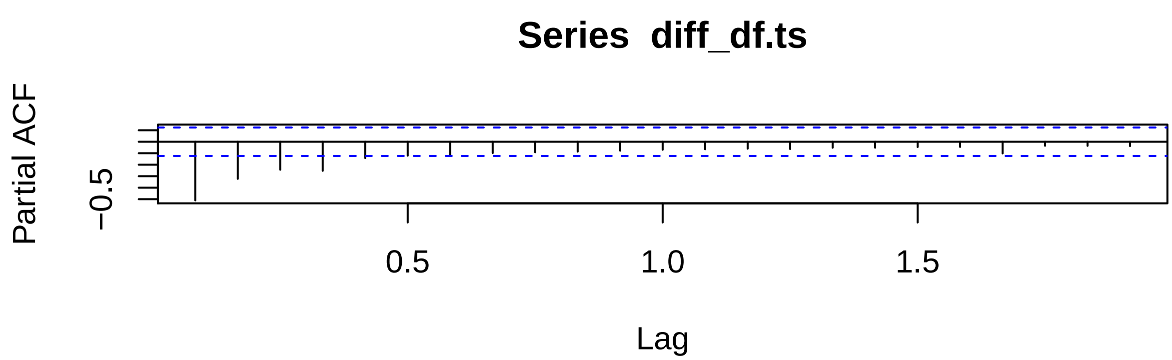



```
diff_df.ts <- diff(df.ts)
plot(diff_df.ts)
```




```r
# ACF shows AR(1). PACF MA(2)?
par(mfrow = c(2, 1))
acf(diff_df.ts)
pacf(diff_df.ts)
```



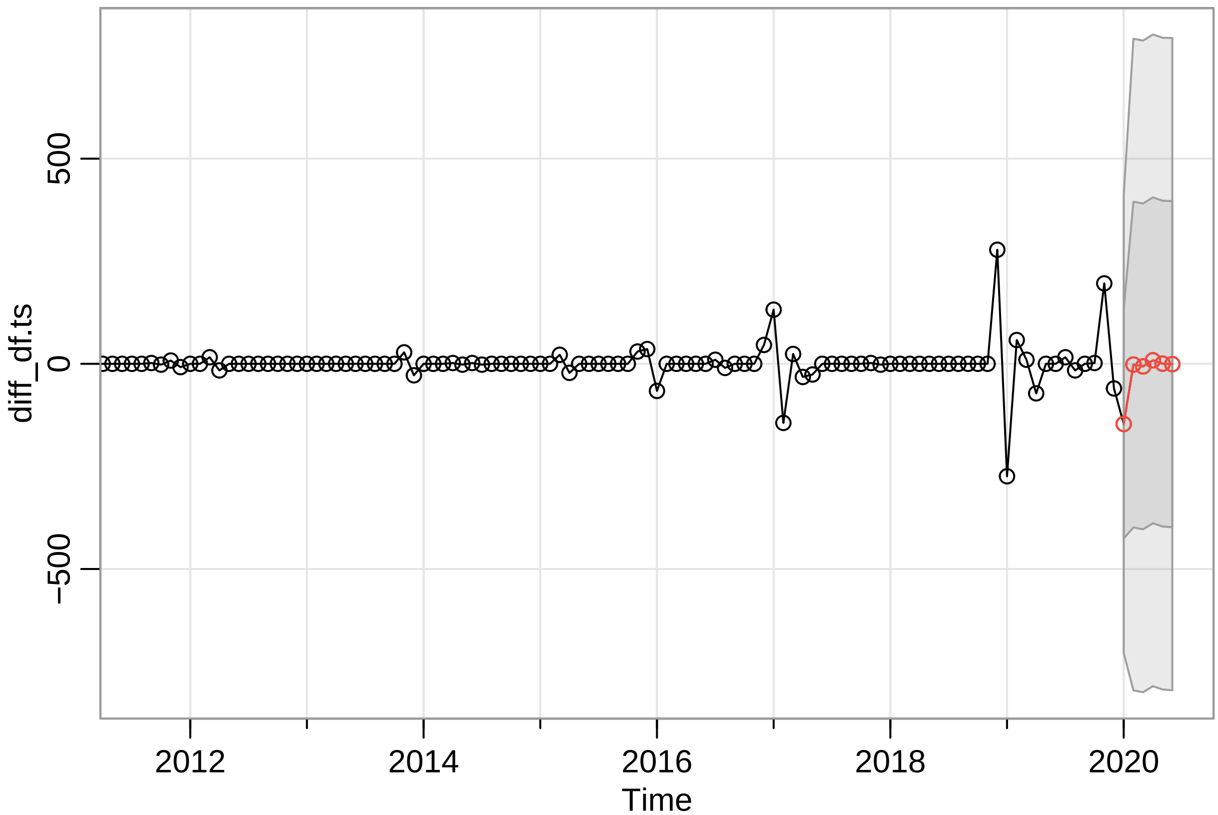



```r
auto.arima(diff_df.ts)
```

```
## Series: diff_df.ts
```

```
## ARIMA(4,0,2) with zero mean
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ma1      ma2
##       -0.6729  -0.0010  -0.0100  -0.0402  -0.3283  -0.6535
## s.e.   0.6476   0.0783   0.0781   0.0662   0.6453   0.6396
##
## sigma^2 estimated as 80628:  log likelihood=-1772.92
## AIC=3559.84   AICc=3560.3   BIC=3584.51
```

```
sarima.for(diff_df.ts,6,4,0,2)
```




```
## $pred
##                Jan          Feb          Mar          Apr          May
## 2020 -146.5583212   -1.7609390   -6.2928448    8.5912431    0.4789333
##                Jun
## 2020   -0.7646626
##
## $se
##           Jan      Feb      Mar      Apr      May      Jun
## 2020 279.1124 396.9769 397.0116 397.0593 397.0787 397.3672
```

```
fit <- sarima(diff_df.ts, 4,0,2)
```

```
## initial  value 5.993974
## iter   2 value 5.795794
## iter   3 value 5.749163
## iter   4 value 5.713440
## iter   5 value 5.707285
## iter   6 value 5.692694
```

```
## iter   7 value 5.679441
## iter   8 value 5.675923
## iter   9 value 5.667771
## iter  10 value 5.658786
## iter  11 value 5.656696
## iter  12 value 5.655303
## iter  13 value 5.653807
## iter  14 value 5.652265
## iter  15 value 5.651887
## iter  16 value 5.651163
## iter  17 value 5.650686
## iter  18 value 5.650174
## iter  19 value 5.649183
## iter  20 value 5.647775
## iter  21 value 5.647392
## iter  22 value 5.646311
## iter  23 value 5.646258
## iter  24 value 5.646205
## iter  25 value 5.646138
## iter  26 value 5.646103
## iter  27 value 5.646052
## iter  28 value 5.646028
## iter  29 value 5.646027
## iter  30 value 5.646026
## iter  31 value 5.646026
## iter  32 value 5.646024
## iter  33 value 5.646024
## iter  34 value 5.646024
## iter  34 value 5.646024
## iter  34 value 5.646024
## final  value 5.646024
## converged
## initial  value 5.644270
## iter   2 value 5.642510
## iter   3 value 5.641851
## iter   4 value 5.641410
## iter   5 value 5.641022
## iter   6 value 5.641017
## iter   7 value 5.641017
## iter   8 value 5.641015
## iter   9 value 5.641010
## iter  10 value 5.640999
## iter  11 value 5.640981
## iter  12 value 5.640955
## iter  13 value 5.640895
## iter  14 value 5.640877
## iter  15 value 5.640872
## iter  16 value 5.640866
## iter  17 value 5.640866
## iter  18 value 5.640865
## iter  19 value 5.640863
## iter  20 value 5.640862
## iter  20 value 5.640862
## iter  20 value 5.640862
```

```
## final  value 5.640862
## converged
```