# Project 1 - Time Series Forecasting STD Dynamics in San Diego

Denise Gandara[*]

July 20, 2022

## Contents

[*] denisegandara10@gmail.com

# 1 Problem

**The problem is to find which model most accurately predicts/forecasts future cases of Chlamydia and Gonorrhea in San Diego population.** How is data collected? Cases of notifiable diseases such as STDs, are collected by health care providers in every state and reported to their respective reportable disease surveillance programs. This data can be used for analysis in forecasting future cases of STD's.

How will we forecast? Time Series Prediction Analysis is a methodology that can be used to forecast future cases of communicable diseases. The predictions provide insights to implement measures to prevent spread of diseases. We will use different models and evaluate their performance using root mean square error.

# 2 Read Data

## 2.1 Libraries

```
#install.packages("shiny")
library(shiny)
#install.packages("gridExtra")
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
#install.packages("ggplot2")
#install.packages("tidyverse")

#library(dplyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
#install.packages("fpp2")
library(fpp2)
```

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

```
## -- Attaching packages ---------------------------------------------- fpp2 2.4 --
```

```
## v forecast  8.16     v expsmooth 2.3
## v fma       2.4
```

```
##
```

```
library(forecast)
```

### 2.2 Data

```
setwd("~/Documents/Bridge Program/r.analysis")
# 2004 to 2017
SD.df = read.csv("SD.STDs.csv", header = TRUE)
SD.df$Month = as.Date(paste(as.character(SD.df$Month), "-15", sep = ""))

#2004 to 2012
SD.df.train = SD.df[-c(109:168), ]
SD.df.test = SD.df[c(109:168), ]

head(SD.df)
```

```
##        Month Chlamydia Gonorrhea Hepatitis.C Syphilis
## 1 2004-01-15       910       204         180       36
## 2 2004-02-15       779       168         235       14
## 3 2004-03-15       854       170         274       38
## 4 2004-04-15      1063       224         102       27
## 5 2004-05-15       785       165          44       34
## 6 2004-06-15       805       178          55       32
```

## 3 Exploratory Data Analysis

### 3.1 Summary

```
colnames(SD.df)
```

```
## [1] "Month"       "Chlamydia"   "Gonorrhea"   "Hepatitis.C" "Syphilis"
```

```
dim(SD.df)
```

```
## [1] 168   5
```

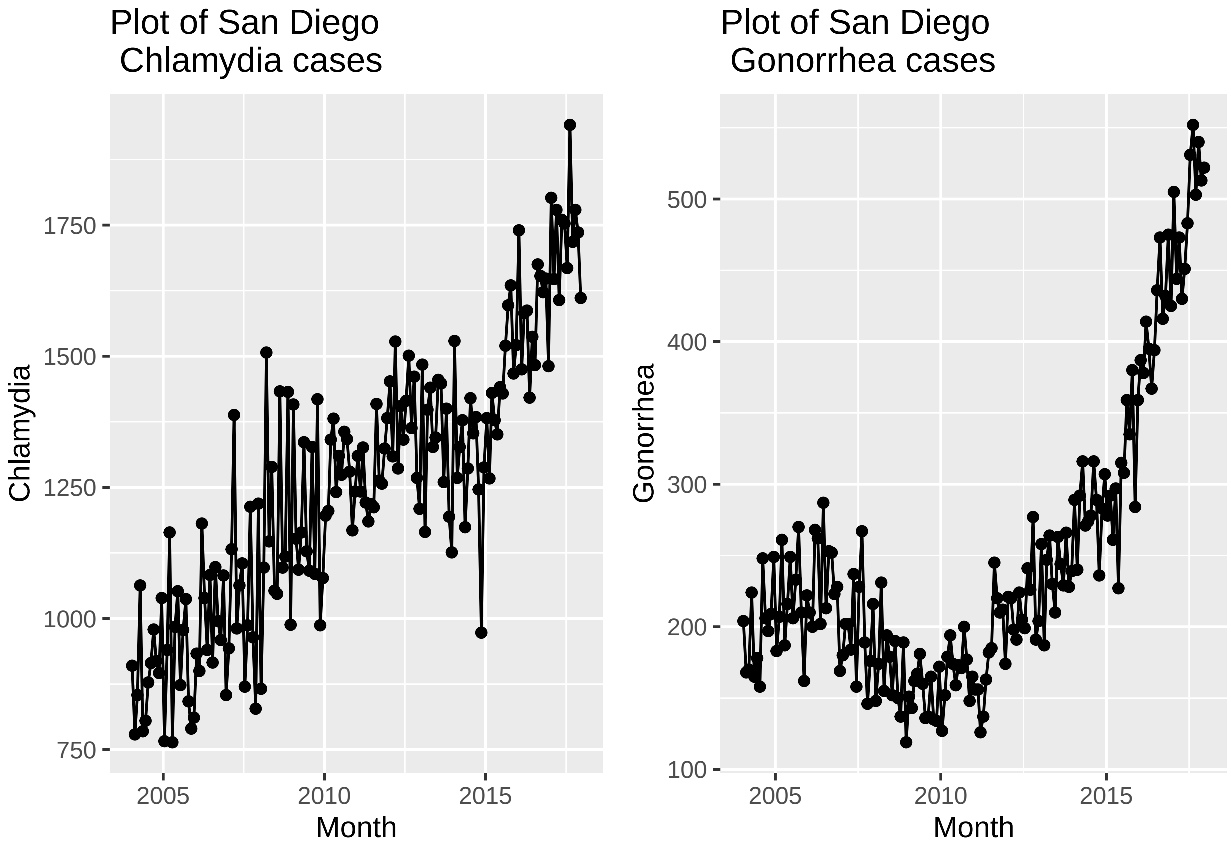
```
summary(SD.df)
```

```
##      Month              Chlamydia      Gonorrhea      Hepatitis.C
##  Min.   :2004-01-15   Min.   : 764   Min.   :119.0   Min.   : 44.0
##  1st Qu.:2007-07-07   1st Qu.:1060   1st Qu.:177.8   1st Qu.:208.5
##  Median :2010-12-30   Median :1268   Median :220.0   Median :275.0
##  Mean   :2010-12-30   Mean   :1255   Mean   :248.0   Mean   :278.2
##  3rd Qu.:2014-06-22   3rd Qu.:1429   3rd Qu.:278.0   3rd Qu.:327.2
##  Max.   :2017-12-15   Max.   :1941   Max.   :552.0   Max.   :643.0
##     Syphilis
##  Min.   : 14.00
##  1st Qu.: 50.00
##  Median : 65.00
##  Mean   : 71.64
##  3rd Qu.: 85.25
##  Max.   :161.00
```

The dataset provides the number of monthly STD cases from January 2004 to December 2017, 14 years of data. It contains 168 observations and 5 variables.

## 3.2 Plot Data

```
p<-ggplot(SD.df, aes(x=Month, y=Chlamydia)) +
  geom_line() +
  geom_point() +
  ggtitle("Plot of San Diego \n Chlamydia cases")

a<-ggplot(SD.df, aes(x=Month, y=Gonorrhea)) +
  geom_line() +
  geom_point() + ggtitle("Plot of San Diego \n Gonorrhea cases")
#install.packages('cowplot')
library(cowplot)
plot_grid(p,a)
```

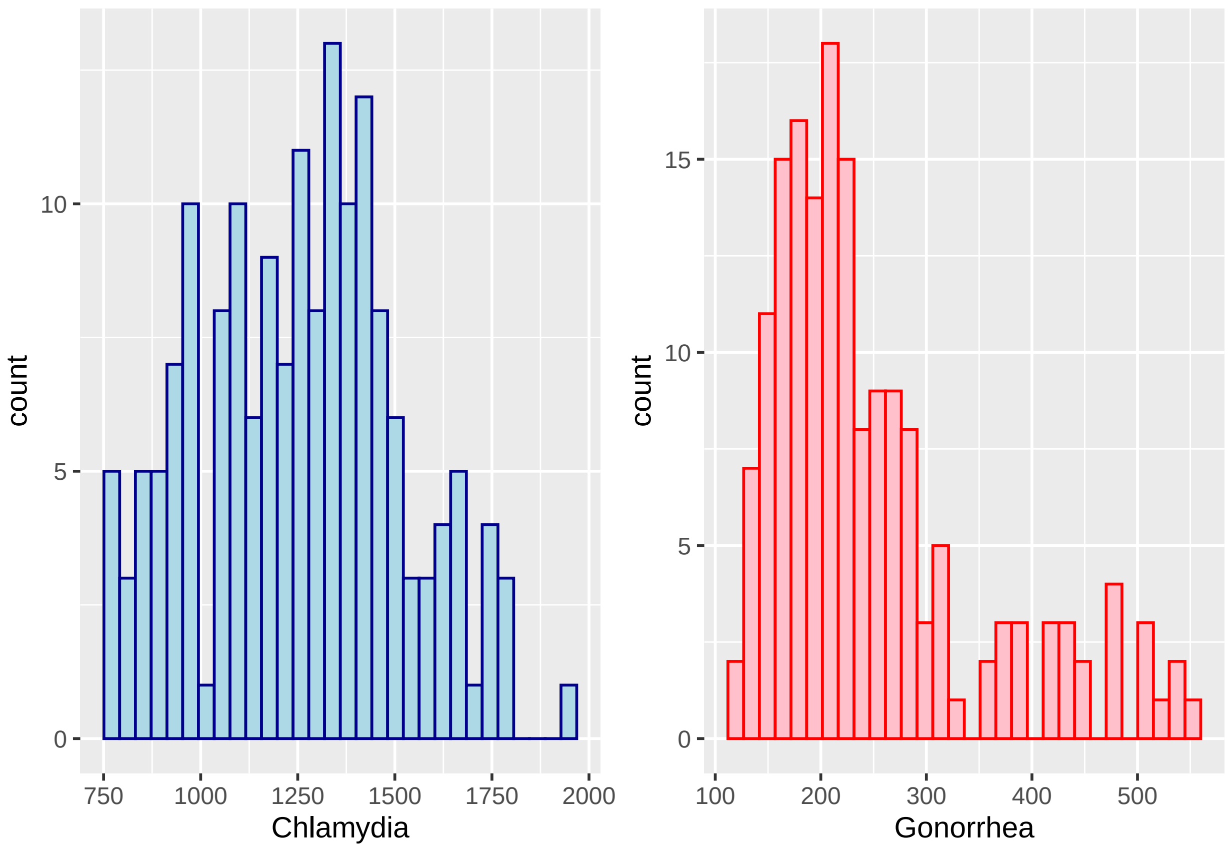


We see positive trends for both graphs. Possibly some exponential increase for Gonorrhea cases.

### 3.3 Distribution

```
p <- ggplot(SD.df, aes(x=Chlamydia))+
  geom_histogram(color="darkblue", fill="lightblue")

a <- ggplot(SD.df, aes(x=Gonorrhea))+
  geom_histogram(color="red", fill="pink")

plot_grid(p,a)
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Both graphs show evidence of right skewness.

## 3.4 Monthly Analysis-Boxplots

Sum of cases per month

```
#group data by year and sum cases
SD.df %>%
    group_by(year = lubridate::floor_date(Month, 'year')) %>%
    summarize(sum_of_Chlamydia_cases = sum(Chlamydia))
```

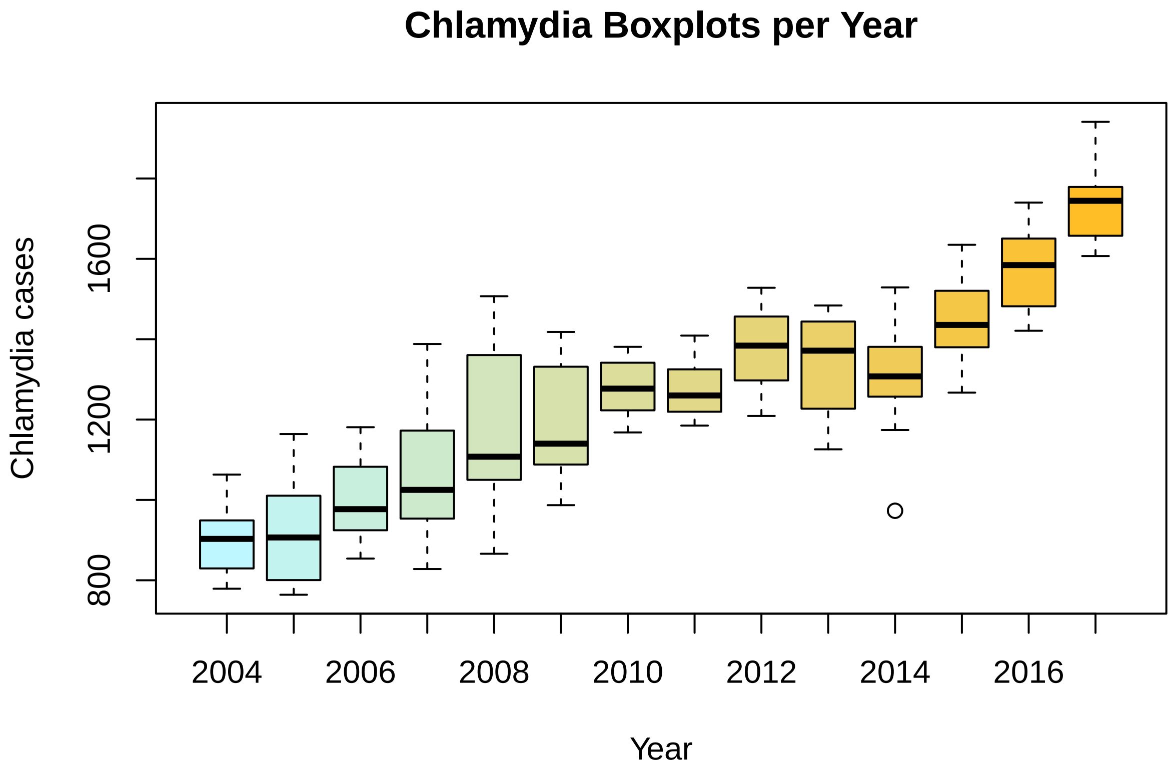
```
## # A tibble: 14 x 2
##    year       sum_of_Chlamydia_cases
##    <date>                      <int>
##  1 2004-01-01                  10822
##  2 2005-01-01                  11001
##  3 2006-01-01                  11980
##  4 2007-01-01                  12693
##  5 2008-01-01                  14074
##  6 2009-01-01                  14266
##  7 2010-01-01                  15336
##  8 2011-01-01                  15349
##  9 2012-01-01                  16538
## 10 2013-01-01                  16042
## 11 2014-01-01                  15626
## 12 2015-01-01                  17418
## 13 2016-01-01                  18904
## 14 2017-01-01                  20801
```

```
SD.df %>%
    group_by(year = lubridate::floor_date(Month, 'year')) %>%
    summarize(sum_of_Gonorrhea_cases = sum(Gonorrhea))
```

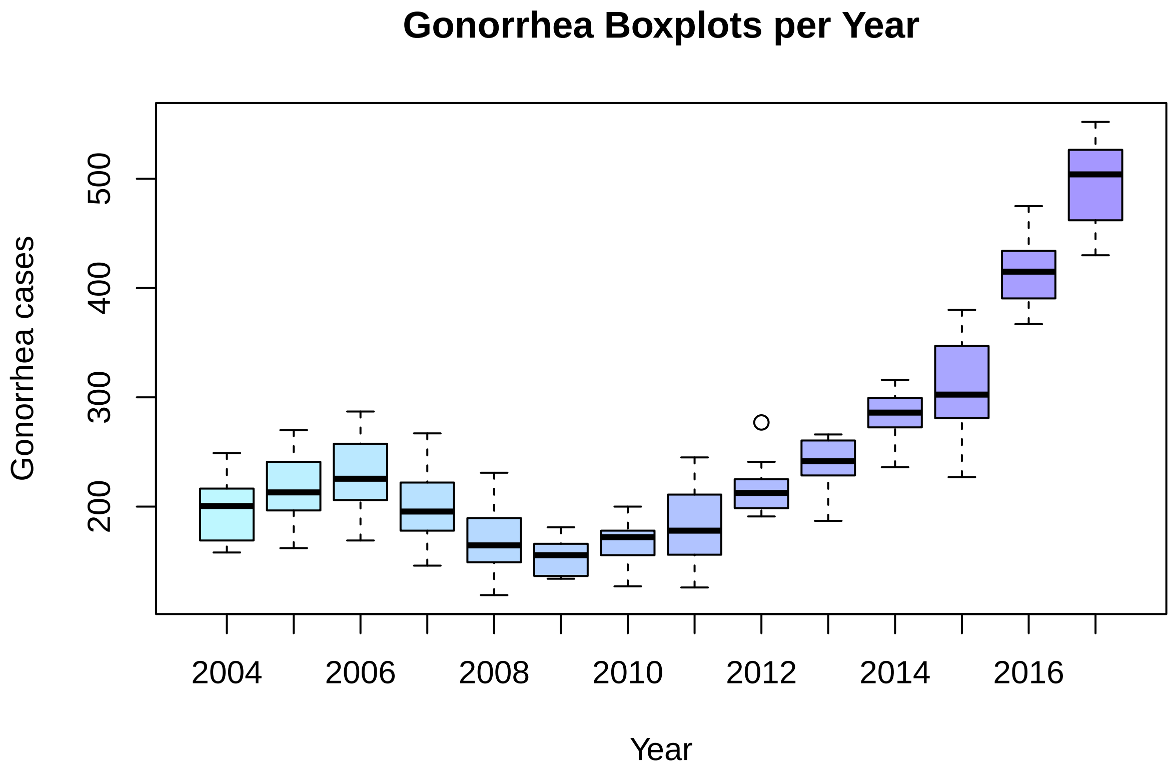
```
## # A tibble: 14 x 2
##    year       sum_of_Gonorrhea_cases
##    <date>                      <int>
##  1 2004-01-01                   2376
##  2 2005-01-01                   2606
##  3 2006-01-01                   2767
##  4 2007-01-01                   2385
##  5 2008-01-01                   2018
##  6 2009-01-01                   1843
##  7 2010-01-01                   2019
##  8 2011-01-01                   2166
##  9 2012-01-01                   2597
## 10 2013-01-01                   2865
## 11 2014-01-01                   3391
## 12 2015-01-01                   3695
## 13 2016-01-01                   4992
## 14 2017-01-01                   5947
```

Calculated the total number of cases for each month.

```
#First Option involves adding a column with only Year, could have replaced Month column as wel
SD.df$Year <-format(SD.df$Month, format="%Y")
#head(SD.df)
boxplot(Chlamydia ~ Year, data = SD.df, xlab = "Year", ylab = "Chlamydia cases", main = "Chlamy
```



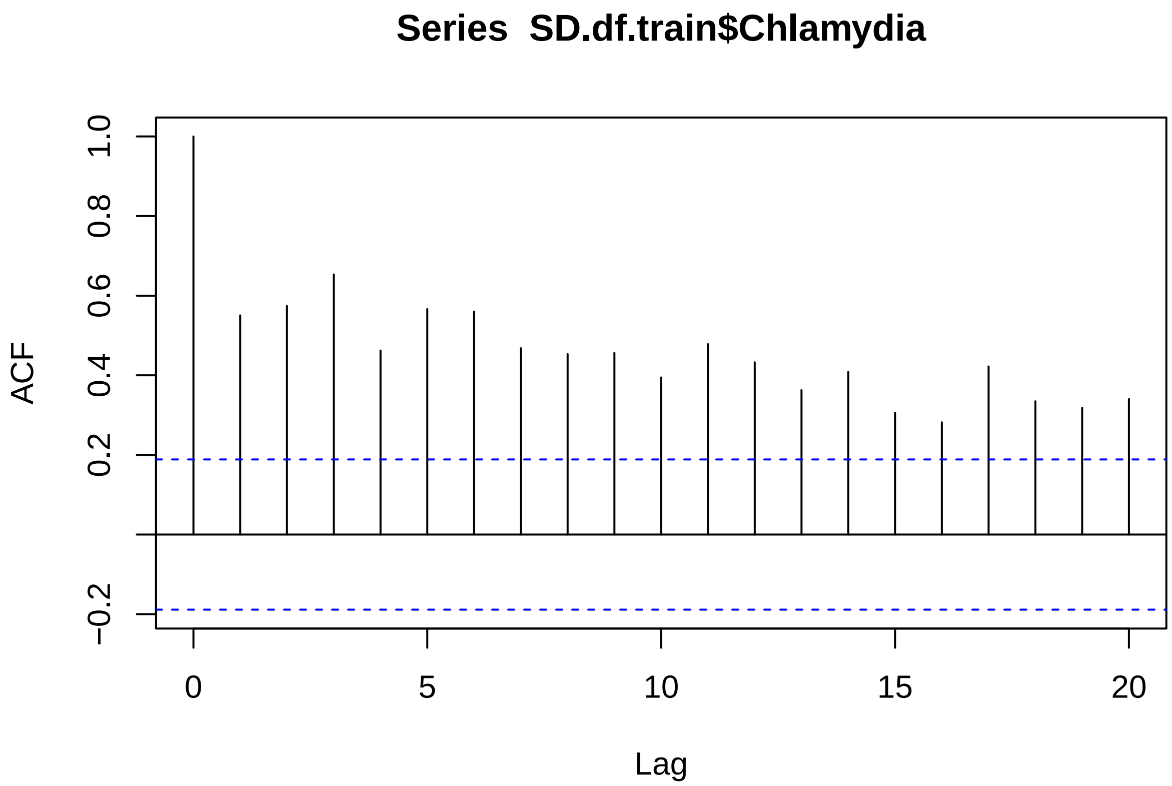
```
#Second Option involves grouping by year, but the x labels will print out year-month-date whic
#boxplot(Chlamydia ~ lubridate::floor_date(Month, 'year'), data = SD.df, xlab = "Year", ylab =

boxplot(Gonorrhea ~ Year, data = SD.df, xlab = "Year", ylab = "Gonorrhea cases", main = "Gonor
```



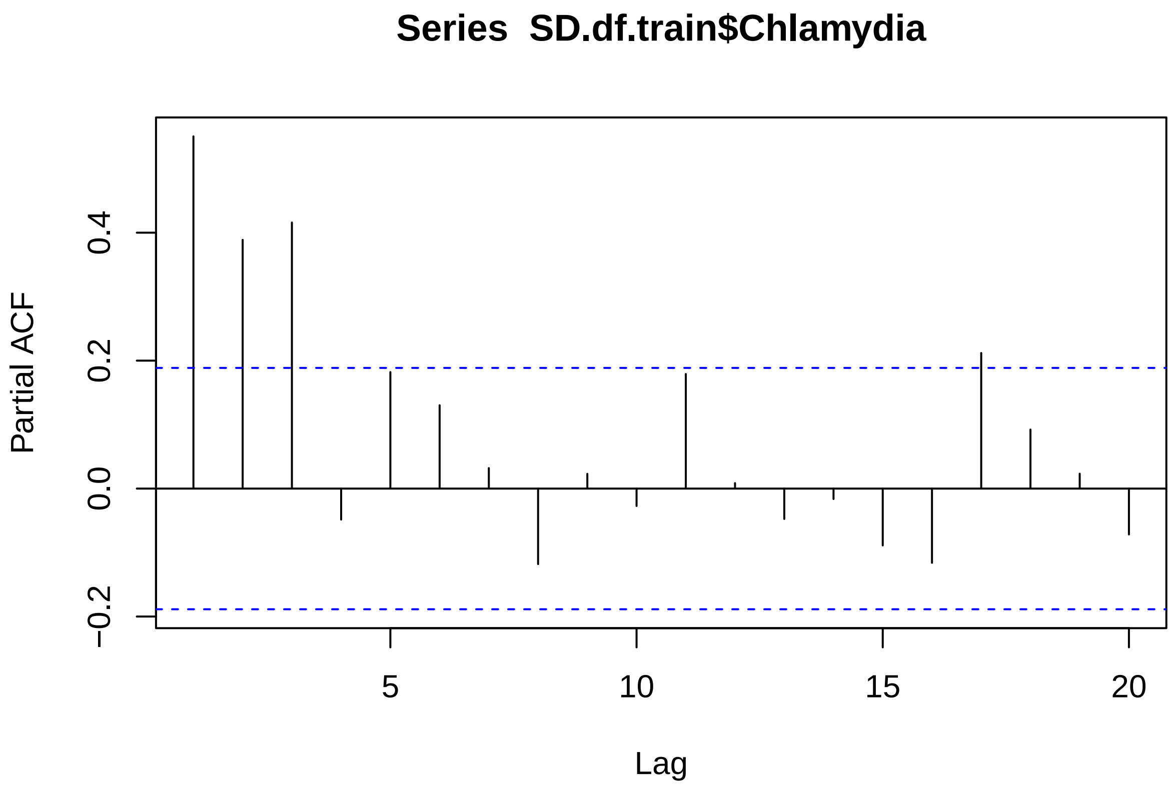
We see mostly increasing trend of cases for both STD's. Gonorrhea cases between 2006 and 2009 show evidence of a downward trend.

### 3.5 Autocorrelation plots

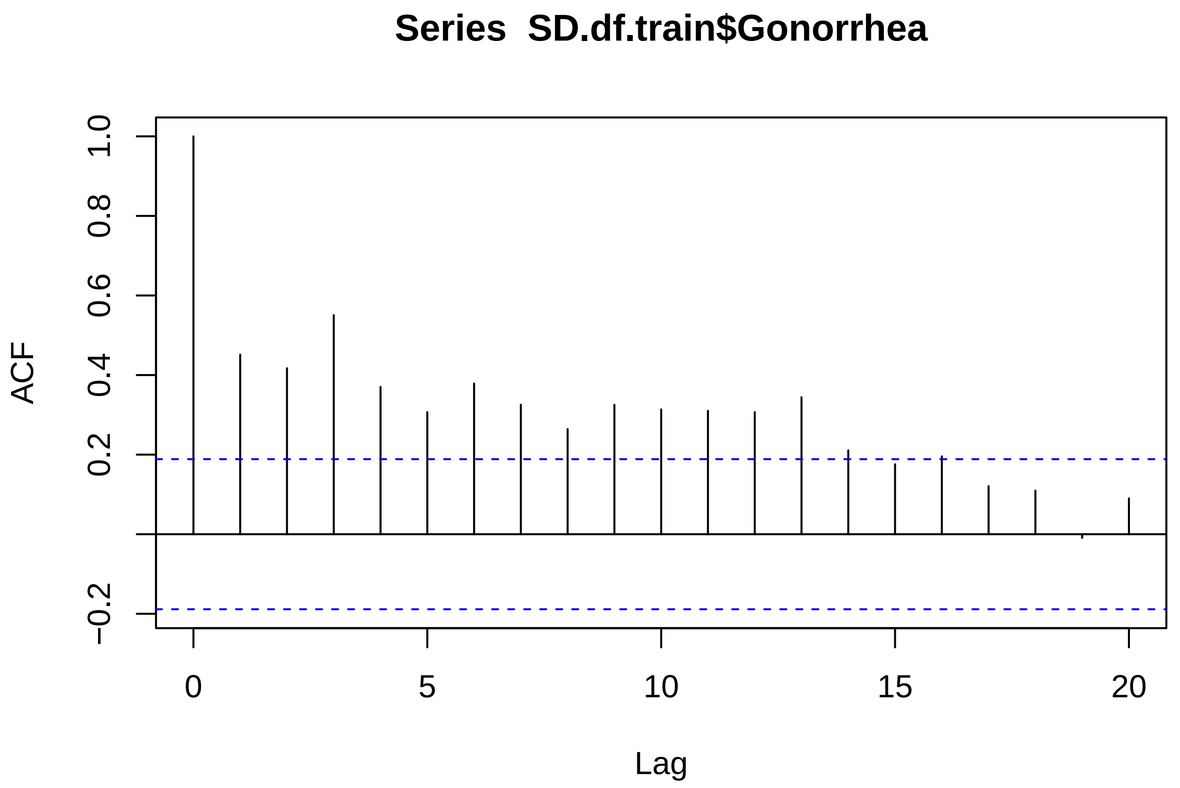
```
acf(SD.df.train$Chlamydia)  # it looks exponential. Looks like S=3
```



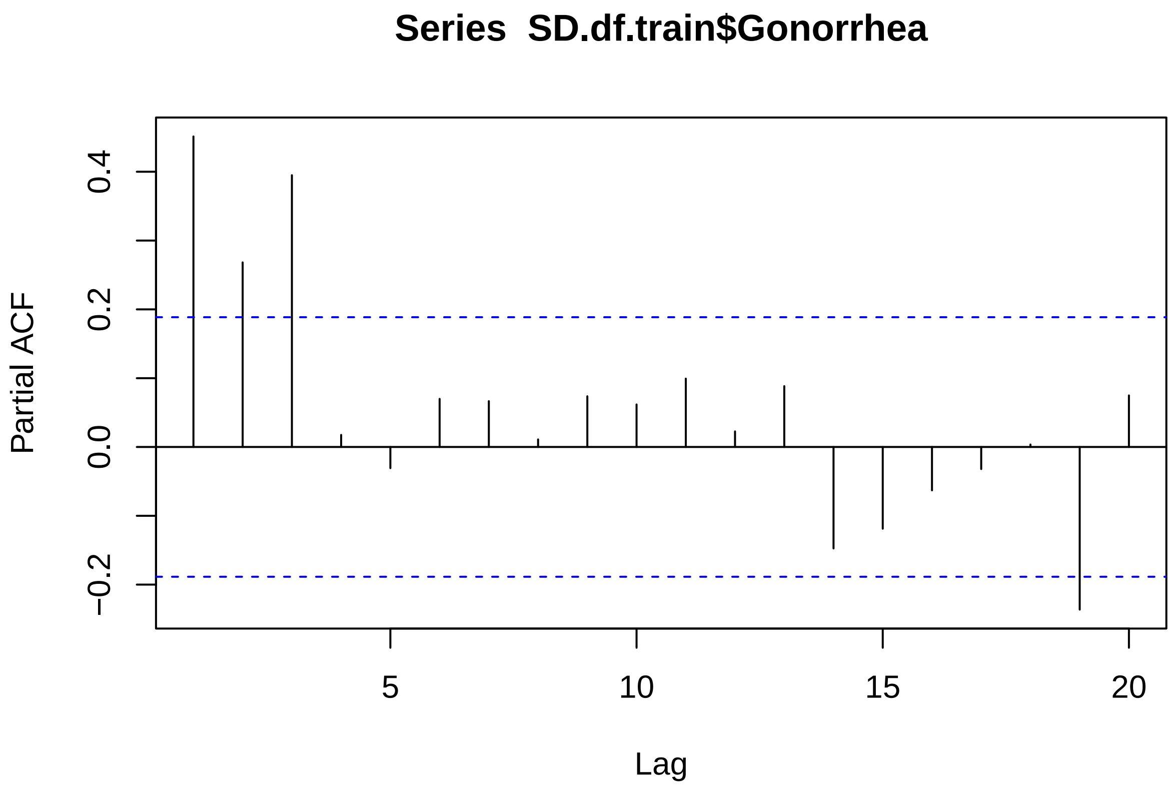
```
                                  #ACF doesn't go to zero, it needs differencing, remove trend. Use
pacf(SD.df.train$Chlamydia) #try AR(3) S=3
```



```
acf(SD.df.train$Gonorrhea) #looks exponential, S=3 maybe
```



```
pacf(SD.df.train$Gonorrhea) #try AR(3) S=3
```



We check autocorrelation which measures the linear relationship between lagged values. The dashed blue lines indicate whether the correlations are significantly different from zero. A slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due to the seasonality. https://otexts.com/fpp3/acf.html

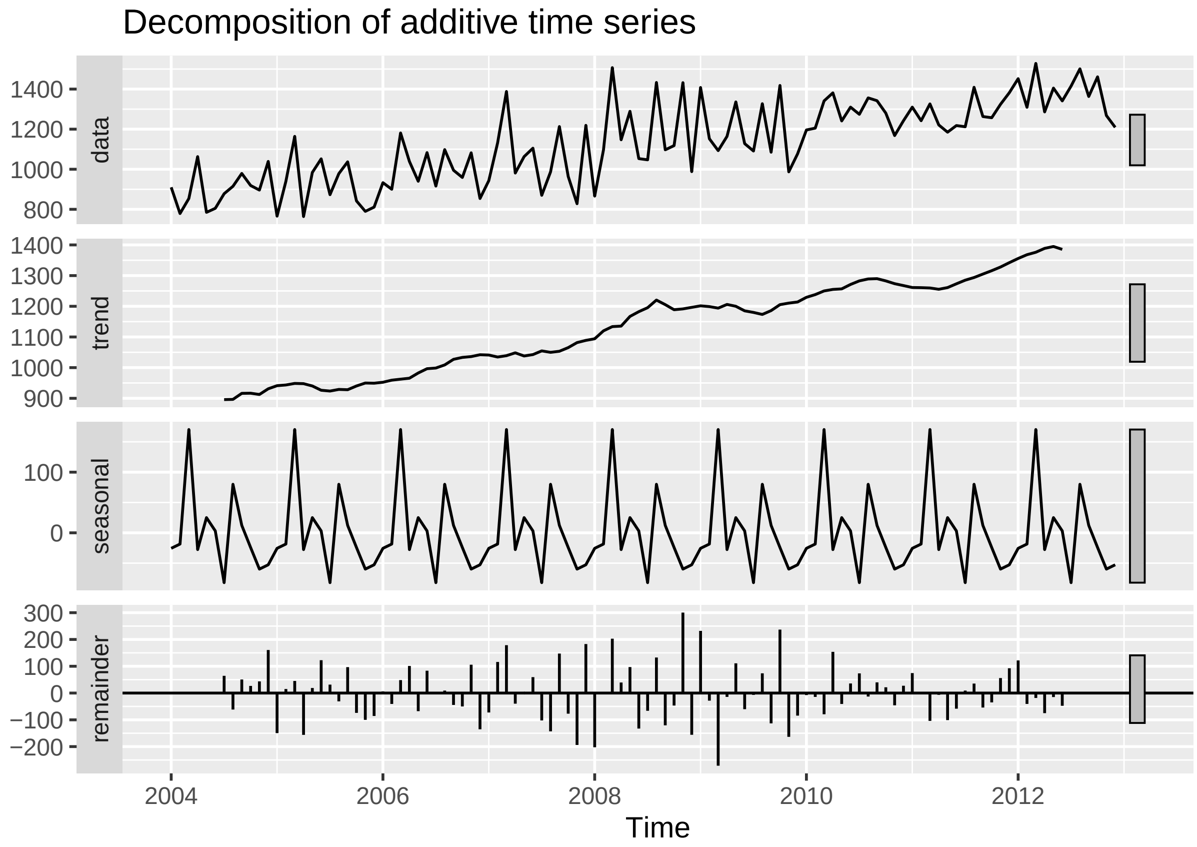
We see evidence of trend and possibly seasonality for both STD cases

```
tsdata.C = ts(SD.df.train$Chlamydia, frequency = 12, start = (2004)) # Chlamydia
tsdata.G = ts(SD.df.train$Gonorrhea, frequency = 12, start = (2004)) # Gonorrhea
```

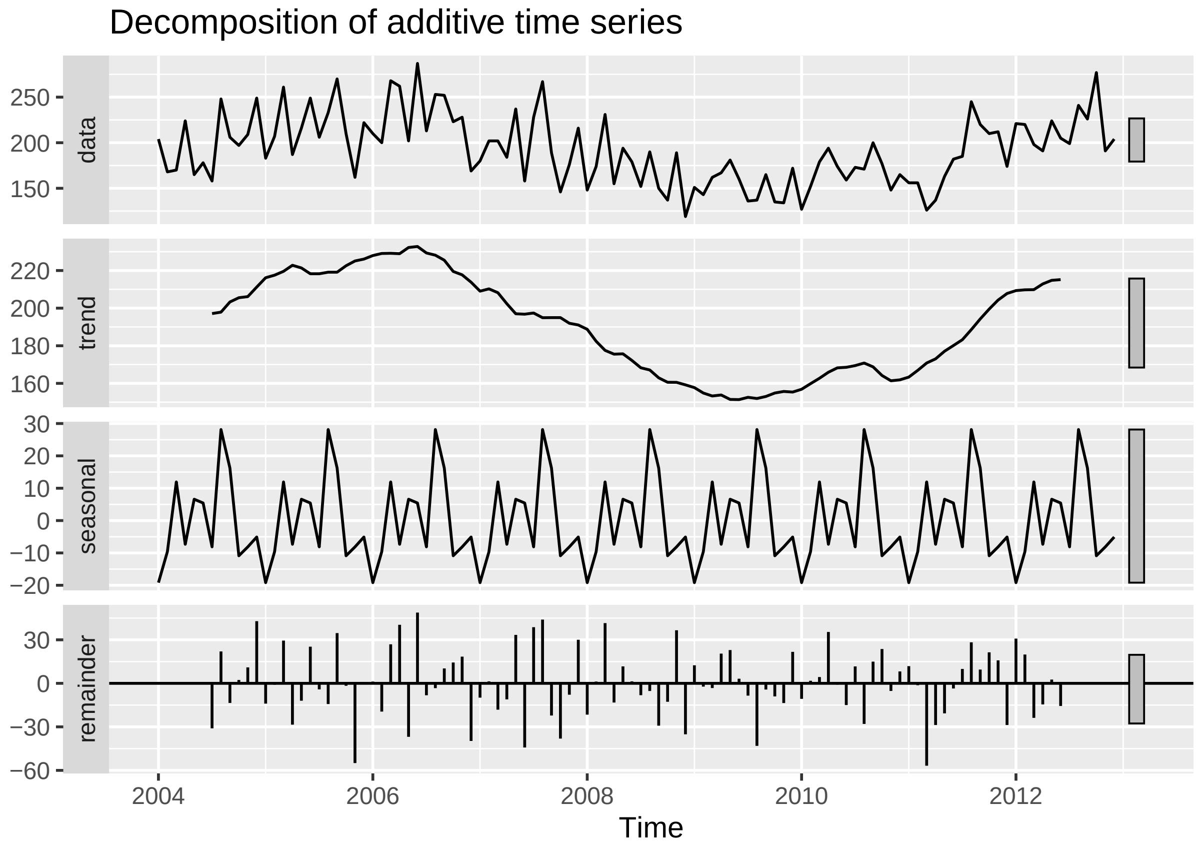
Now we convert data to timeseries data

### 3.6 Decomposition

```
#Chlamydia
autoplot(decompose(tsdata.C))
```

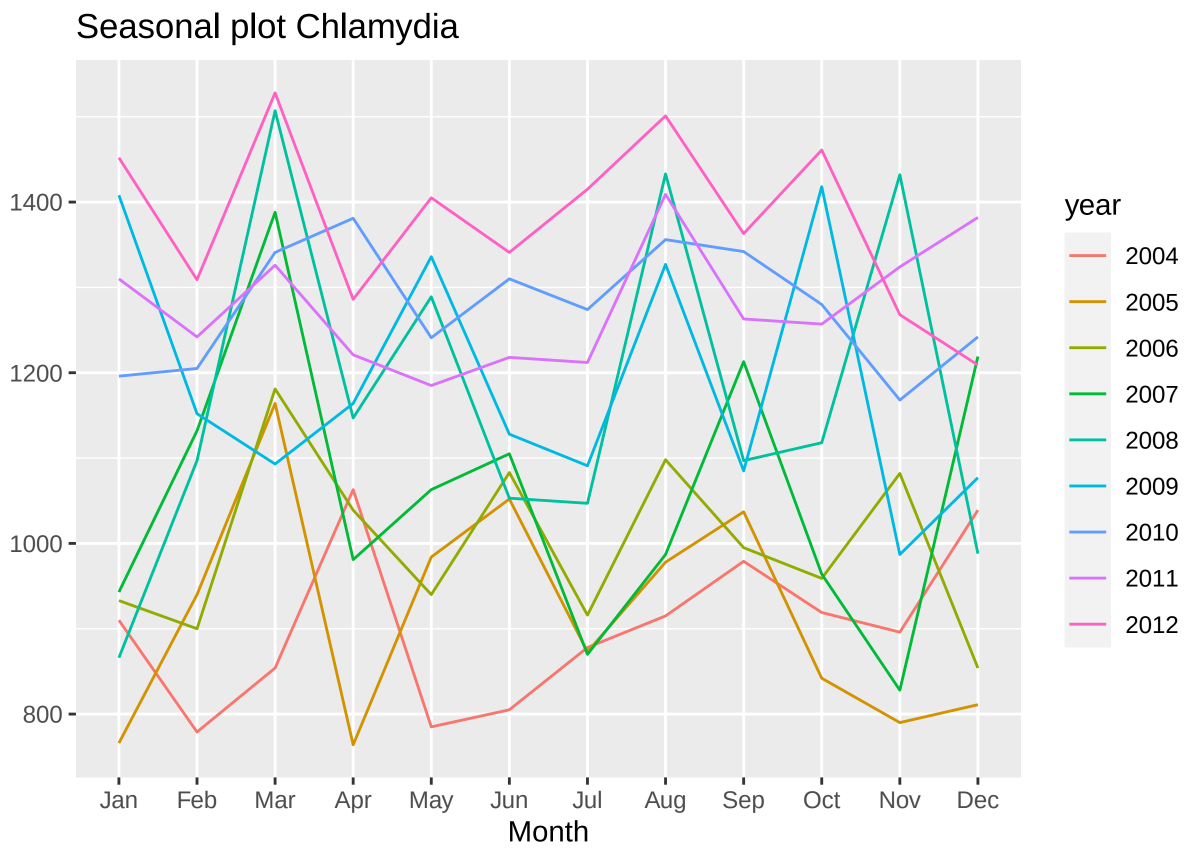


```
#Gonorrhea
autoplot(decompose(tsdata.G))
```

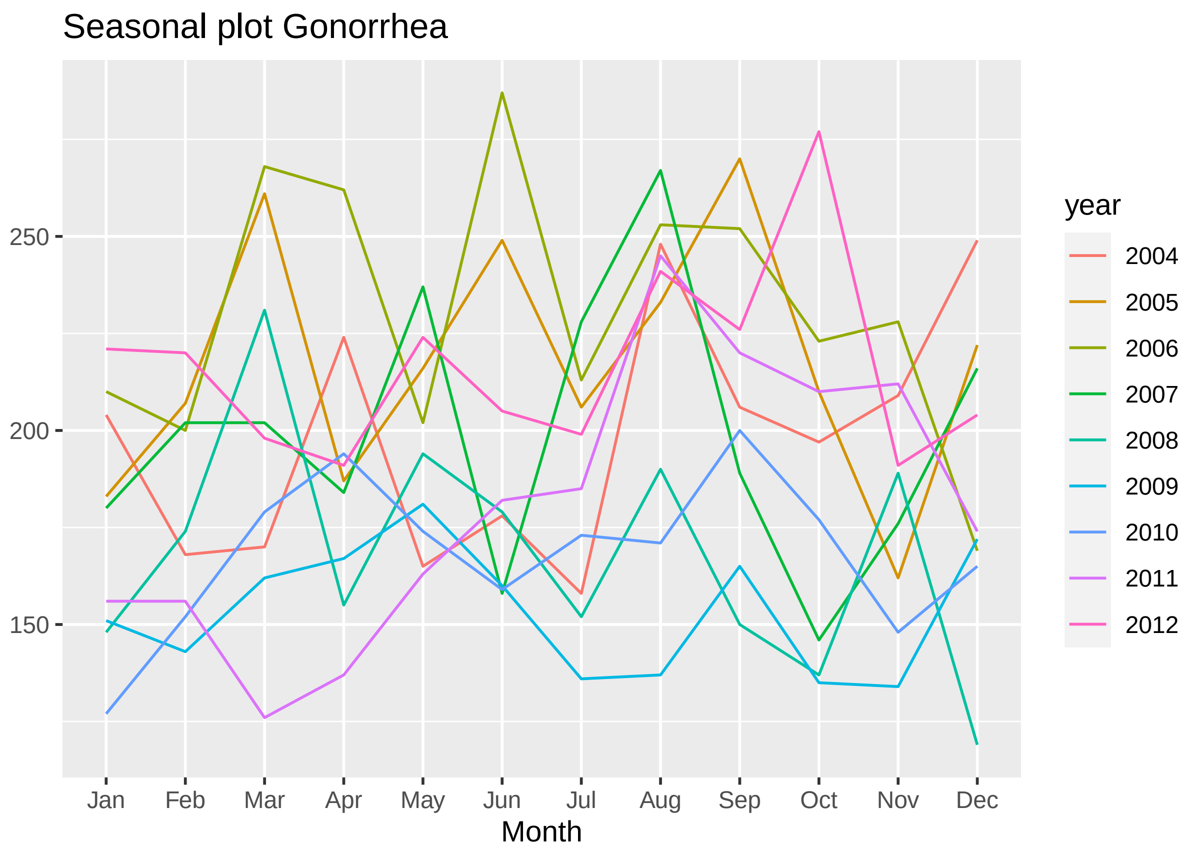


### 3.7 Seasonality

```
ggseasonplot(tsdata.C) +
    ggtitle("Seasonal plot Chlamydia")
```



```
ggseasonplot(tsdata.G) +
    ggtitle("Seasonal plot Gonorrhea")
```



```
#another method
# a <- ggsubseriesplot(tsdata.C)
# b <- ggsubseriesplot(tsdata.G)
# plot_grid(a,b)
```

It is difficult to tell whether there is evidence of seasonality.

# 4 Modelling

## 4.1 ARIMA model

ARIMA is the abbreviation for AutoRegressive Integrated Moving Average and is the most widely used approach for forecasting time series data. Auto Regressive (AR) terms refer to the lags of the differenced series, Moving Average (MA) terms refer to the lags of errors and I is the number of difference used to make the time series stationary. https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/

```
C.arima <- auto.arima(tsdata.C, stepwise = FALSE, approximation = FALSE)
G.arima <- auto.arima(tsdata.G, stepwise = FALSE, approximation = FALSE)
```

```
summary(C.arima)
```

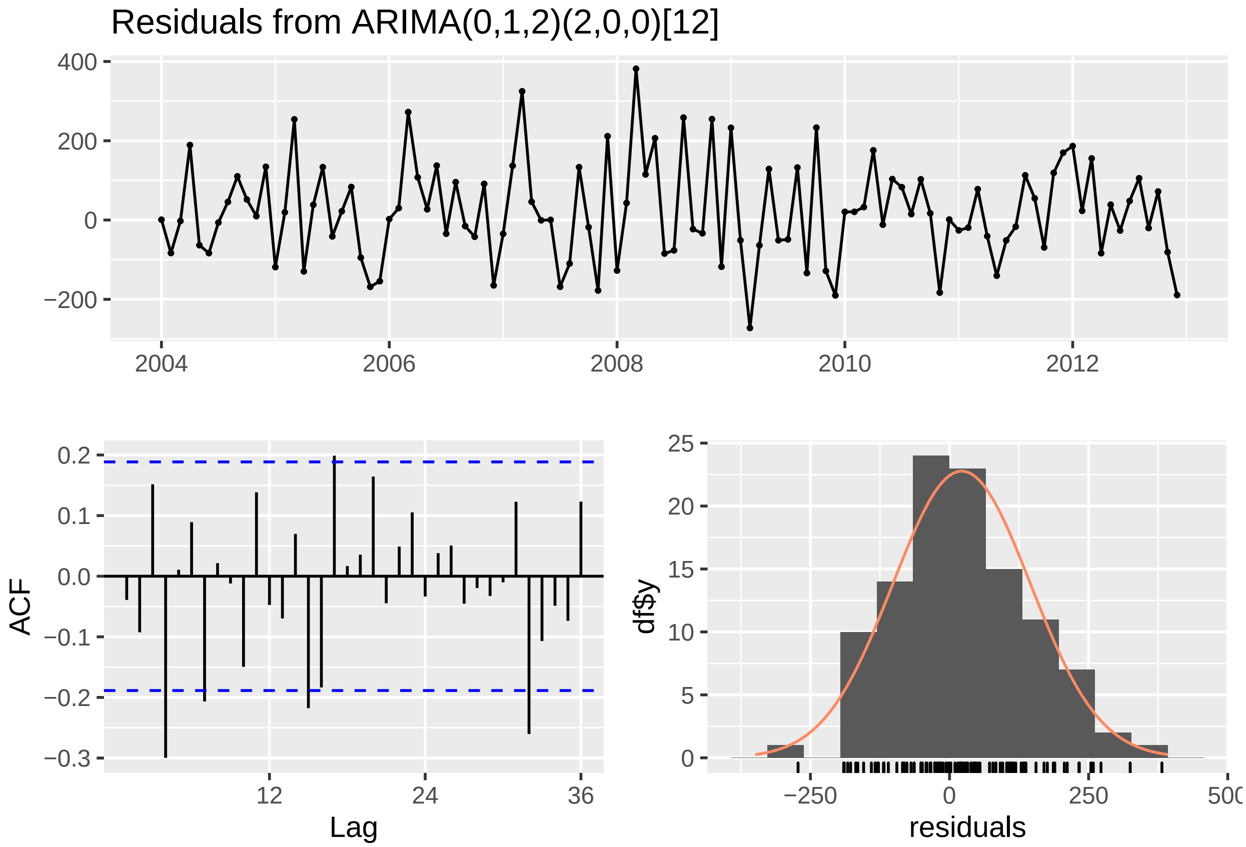
```
## Series: tsdata.C
## ARIMA(0,1,2)(2,0,0)[12]
##
## Coefficients:
##           ma1     ma2    sar1    sar2
##       -1.0834  0.2225  0.0680  0.3052
## s.e.   0.0976  0.0922  0.0896  0.0989
##
## sigma^2 = 16402:  log likelihood = -671.06
## AIC=1352.12   AICc=1352.71   BIC=1365.48
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 21.81811 125.0719 97.27085 0.8719199 8.611593 0.6707371
##                     ACF1
## Training set -0.03905842
```

```
summary(G.arima)
```

```
## Series: tsdata.G
## ARIMA(2,1,0)
##
## Coefficients:
##           ar1      ar2
##       -0.7003  -0.5008
## s.e.   0.0843   0.0852
##
## sigma^2 = 938.7:  log likelihood = -517.41
## AIC=1040.82   AICc=1041.05   BIC=1048.84
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE    MAPE      MASE       ACF1
## Training set 0.4305478 30.20989 24.13111 -1.674696 12.8099 0.6744065 -0.0375223
```

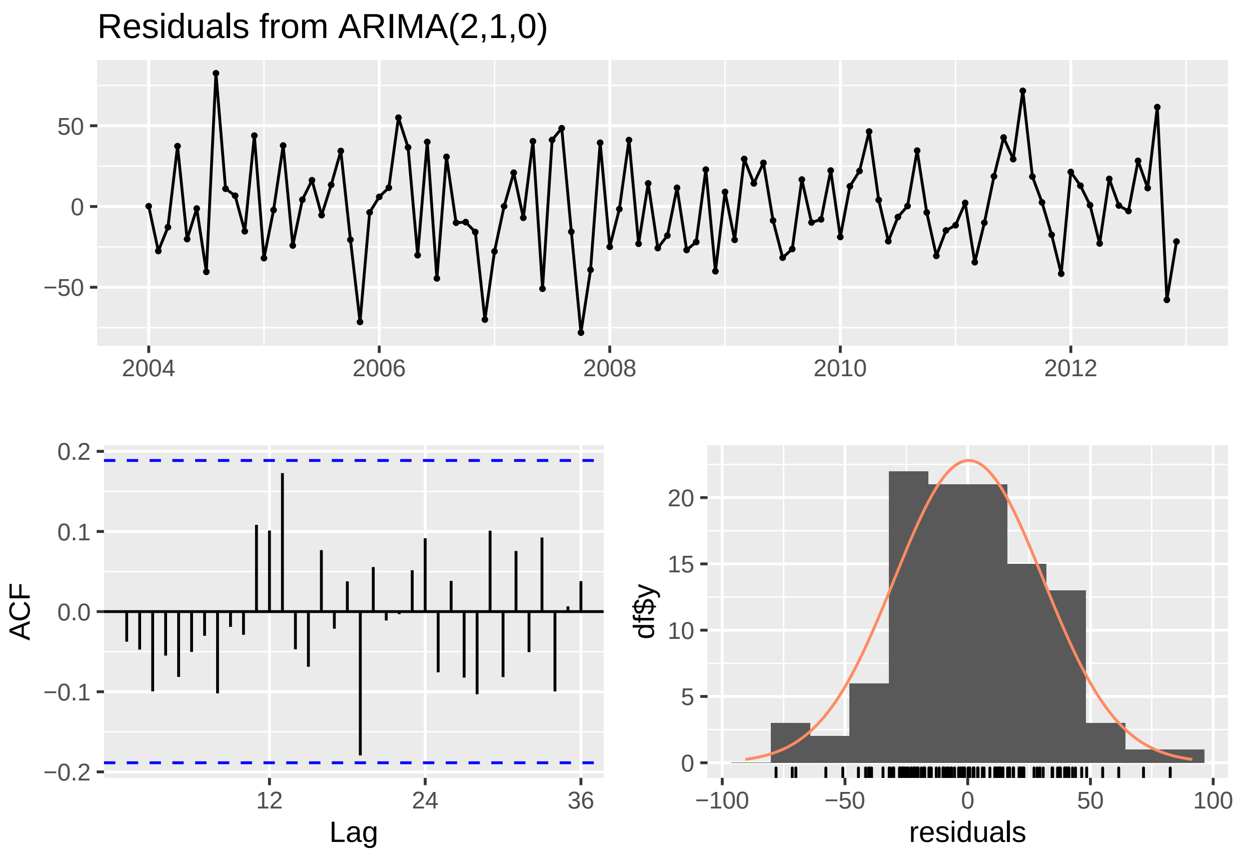
Our chlamydia model created a Seasonal ARIMA model because it detected seasonality. Our gonorrhea model created a ARIMA model without seasonality.

```
checkresiduals(C.arima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(2,0,0)[12]
## Q* = 46.591, df = 18, p-value = 0.0002431
##
## Model df: 4.   Total lags used: 22
```

```
checkresiduals(G.arima)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)
## Q* = 17.482, df = 20, p-value = 0.6215
##
## Model df: 2.   Total lags used: 22
```

Chlamydia model seems to show normality with zero mean (bottom right plot). We can observe that the residuals may still be correlated (bottom left plot).

Gonorrhea model seems to be performing well. We see normality and we can observe that the residuals are uncorrelated (bottom left plot) and do not exhibit any obvious seasonality (the top plot). Also, the residuals are roughly normally distributed with zero mean (bottom right plot). Again, this is a strong indication that the residuals are normally distributed which is what we want.

```
C.forecast <- forecast(C.arima, h = 60)
#autoplot(C.forecast, main = "ARIMA Forecast Chlamydia")
#summary(C.forecast)  prints point forecast and intervals


G.forecast <- forecast(G.arima, h = 60)
```

```
#autoplot(G.forecast, main = "ARIMA Forecast Gonorrhea")
#summary(G.forecast)  prints point forecast and intervals
```

Obtained the forecast for ARIMA models

### 4.2 Exponential Smoothing Method model

This method produces forecasts that are weighted averages of past observations where the weights of older observations exponentially decrease.

```
fit_ets <- ets(tsdata.C)
fit_ets2 <- ets(tsdata.G)
```

```
summary(fit_ets)
```

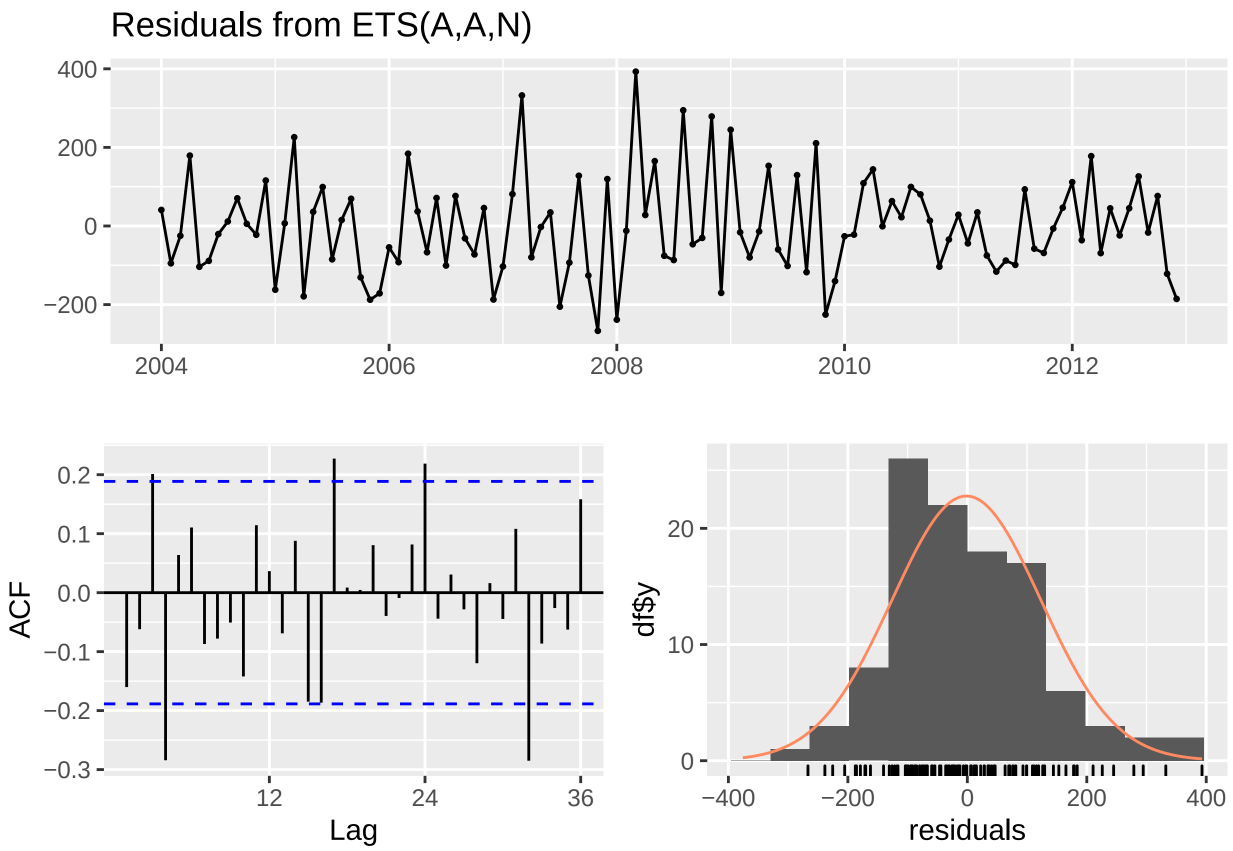
```
## ETS(A,A,N)
##
## Call:
##  ets(y = tsdata.C)
##
##   Smoothing parameters:
##     alpha = 1e-04
##     beta  = 1e-04
##
##   Initial states:
##     l = 864.1147
##     b = 4.9319
##
##   sigma:  126.5776
##
##      AIC     AICc      BIC
## 1557.219 1557.807 1570.630
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set -1.496107 124.2115 97.80701 -1.45376 8.917129 0.6744342 -0.1601161
```

```
summary(fit_ets2)
```

```
## ETS(M,A,N)
##
## Call:
##  ets(y = tsdata.G)
##
##   Smoothing parameters:
```

```
##     alpha = 0.0243
##     beta  = 0.0243
##
##   Initial states:
##     l = 171.9191
##     b = 3.3851
##
##   sigma:  0.1587
##
##      AIC     AICc      BIC
## 1247.095 1247.684 1260.506
##
## Training set error measures:
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.1196048 30.09862 25.09183 -1.759791 13.26588 0.7012564
##                     ACF1
## Training set 0.09277948
```

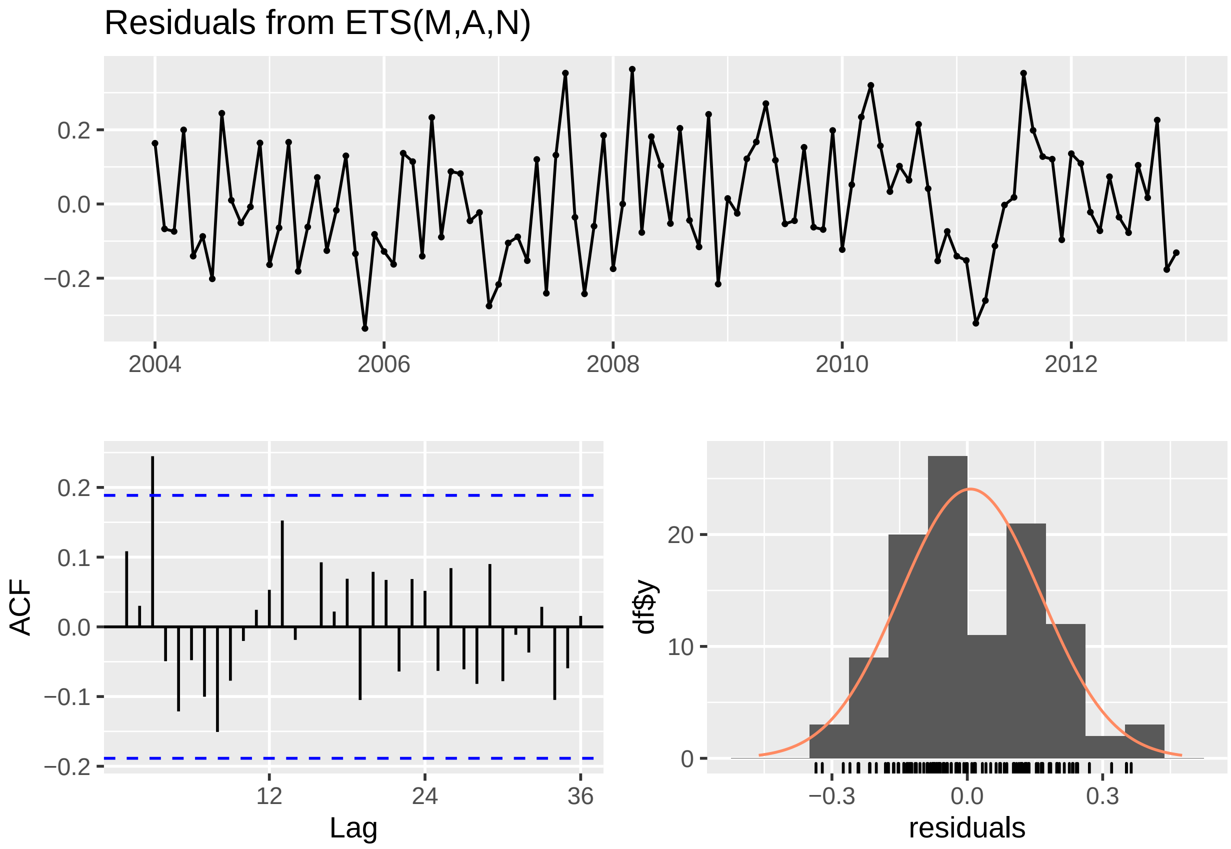
```
checkresiduals(fit_ets)
```



```
##
##  Ljung-Box test
```

```
##
## data:  Residuals from ETS(A,A,N)
## Q* = 43.382, df = 18, p-value = 0.0007065
##
## Model df: 4.   Total lags used: 22
```

```
checkresiduals(fit_ets2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,A,N)
## Q* = 23.747, df = 18, p-value = 0.1635
##
## Model df: 4.   Total lags used: 22
```

Chlamydia model seems to show some right skewness (bottom right plot). We can observe that the residuals may still be correlated (bottom left plot).

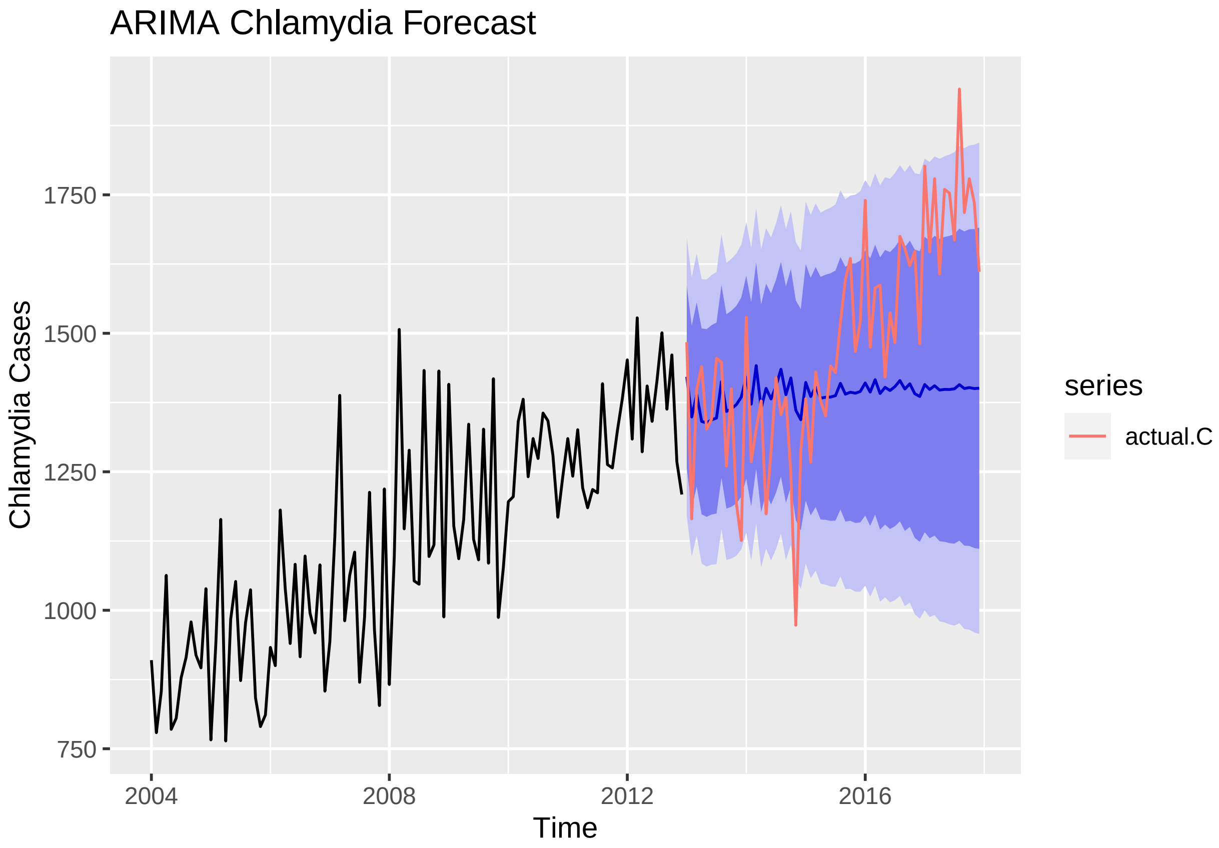
Gonorrhea model seems to be performing well. We can observe that the residuals are uncorrelated (bottom left plot) and do not exhibit any obvious seasonality (the top plot). It is difficult to tell whether the residuals are normally distributed with zero mean since it seems to divide in the center (bottom right plot).

```
frct <- forecast(fit_ets, h = 60)
#autoplot(frct, main = "ETS Forecast Chlamydia")  # include = 60 shows last 5 years of data
#summary(frct) #prints point forecast
frct2 <- forecast(fit_ets2, h = 60)
#autoplot(frct2, main = "ETS Forecast Gonorrhea")
```

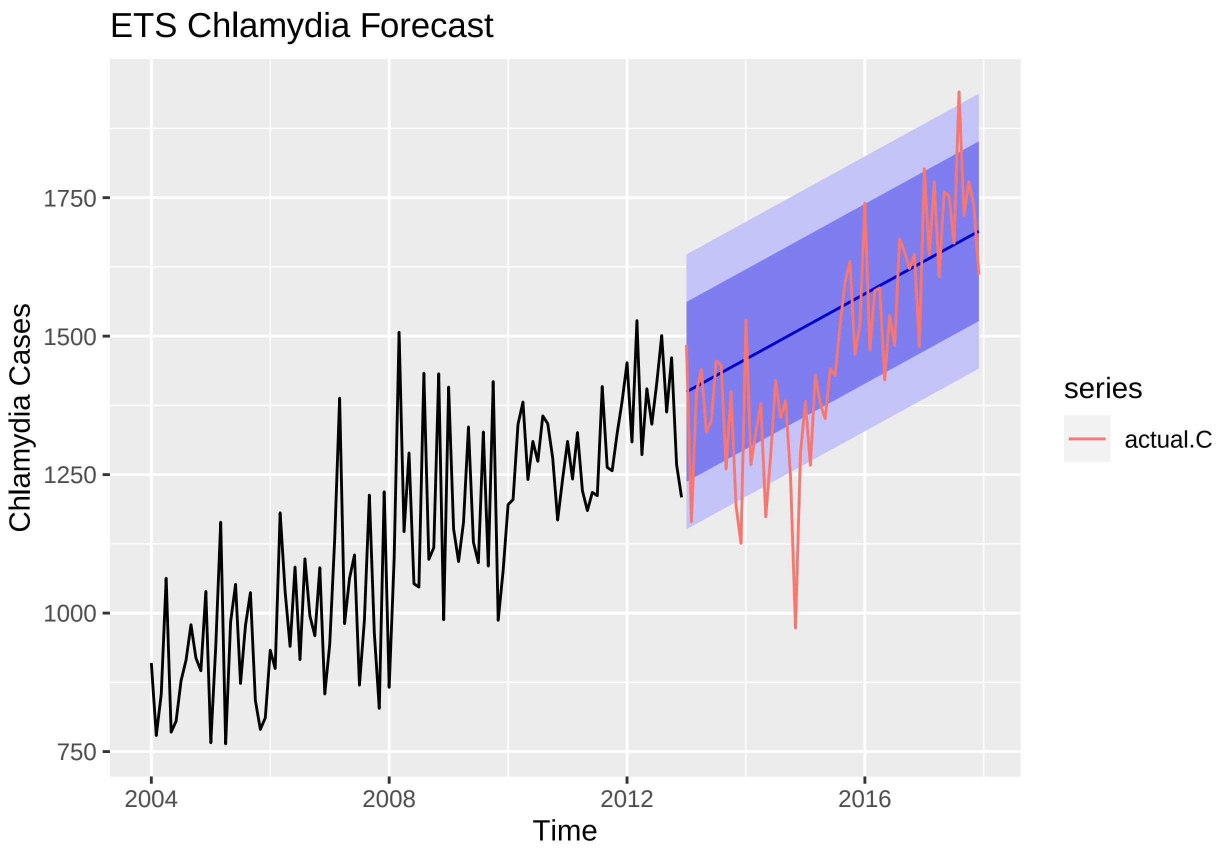
Obtained the forecast for ETS

## 5 Model Comparison

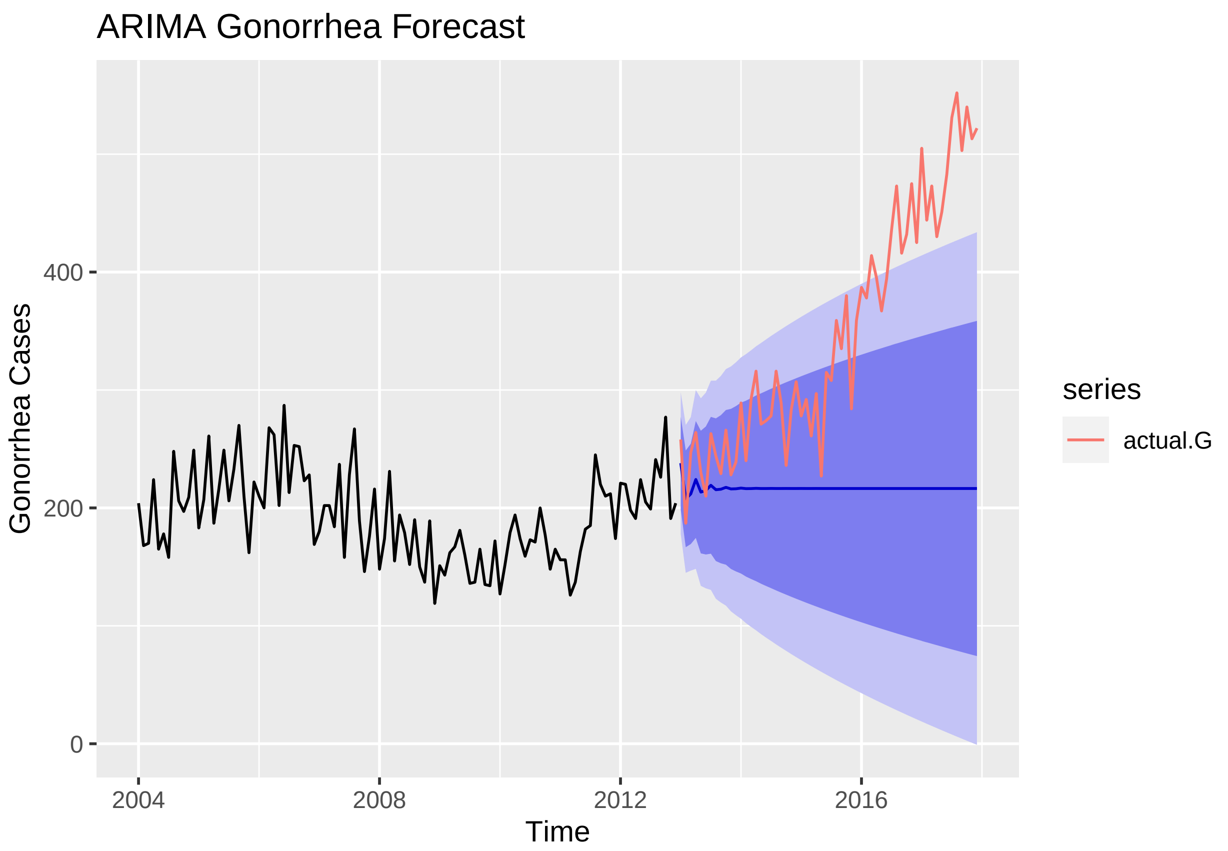
```
# Current data 2018 and 2019 data only
actual.C = ts(SD.df.test$Chlamydia, frequency = 12, start = (2013)) # Chlamydia
actual.G = ts(SD.df.test$Gonorrhea, frequency = 12, start = (2013)) # Gonorrhea


# Chlamydia Models
par(mfrow=c(1,2))
autoplot(C.forecast, main = "ARIMA Chlamydia Forecast", ylab = "Chlamydia Cases") + autolayer(a
```



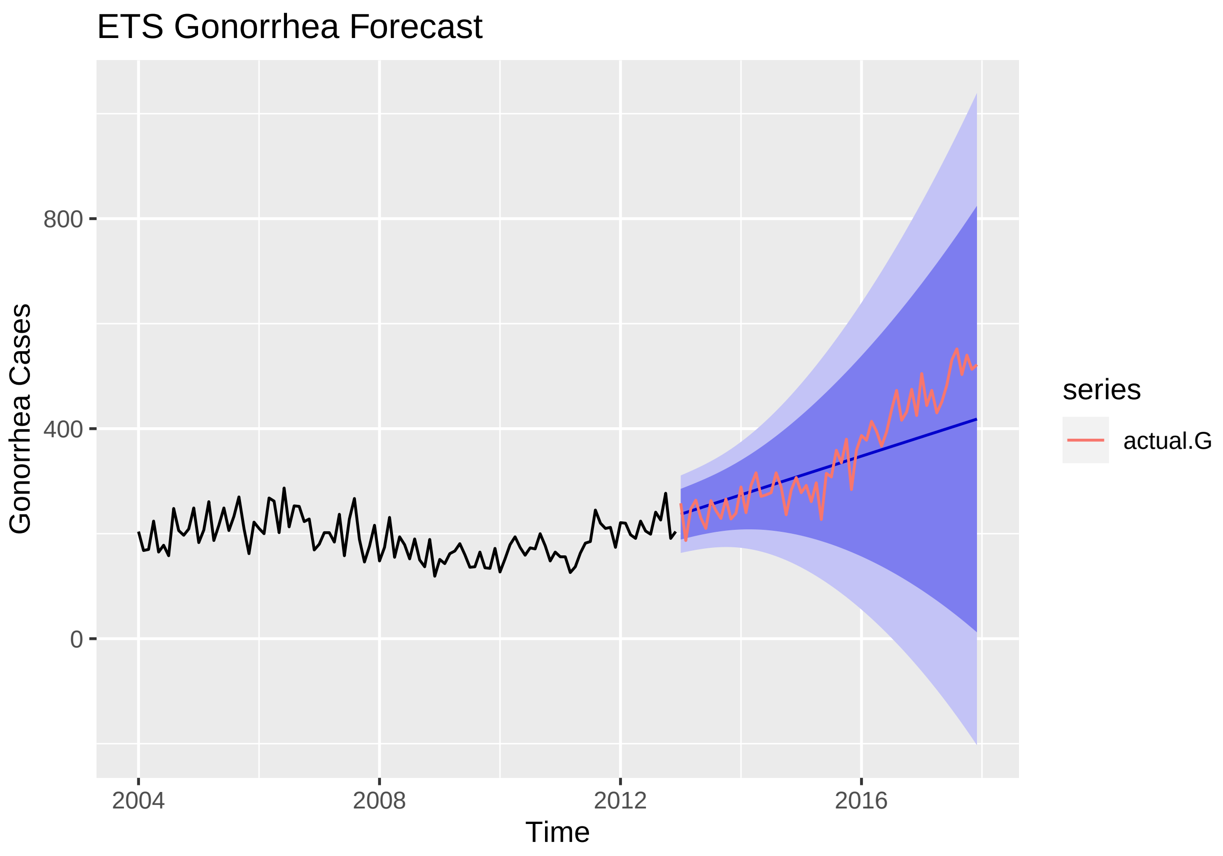
```
autoplot(frct, main = "ETS Chlamydia Forecast",  ylab = "Chlamydia Cases") +
    autolayer(actual.C)
```



```
# ETS
par(mfrow=c(1,2))
autoplot(G.forecast, main = "ARIMA Gonorrhea Forecast",  ylab = "Gonorrhea Cases") + autolayer
```



```r
autoplot(frct2, main = "ETS Gonorrhea Forecast",  ylab = "Gonorrhea Cases") +
    autolayer(actual.G)
```



```
    # include = 60 shows last 5 years of data
```

Here we plot the forecast data with actual data and compare results. We want to see whether ARIMA or ETS models can forecast STD cases better. ETS models seem to be performing better than ARIMA models since the actual data seems to fall more inbetween the predicted intervals.

```
#Chlamydia models comparison
print("ARIMA Chlamydia")
```

```
## [1] "ARIMA Chlamydia"
```

```
accuracy(C.forecast, actual.C)
```

```
##                    ME     RMSE       MAE       MPE      MAPE      MASE
## Training set 21.81811 125.0719  97.27085 0.8719199  8.611593 0.6707371
## Test set     88.04906 203.6084 160.13662 4.3674940 10.529038 1.1042318
##                     ACF1 Theil's U
## Training set -0.03905842        NA
## Test set      0.66513572  1.131453
```

```
print("ETS Chlamydia")
```

```
## [1] "ETS Chlamydia"
```

```
accuracy(frct, actual.C)
```

```
##                       ME     RMSE       MAE       MPE     MAPE      MASE
## Training set  -1.496107 124.2115  97.80701 -1.453760 8.917129 0.6744342
## Test set     -64.625310 152.9252 117.84374 -5.655187 8.736956 0.8125987
##                    ACF1 Theil's U
## Training set -0.1601161        NA
## Test set      0.3438948 0.9826528
```

```
#Gonorrhea models comparison
print("ARIMA Gonorrhea")
```

```
## [1] "ARIMA Gonorrhea"
```

```
accuracy(G.forecast, actual.G)
```

```
##                        ME      RMSE       MAE       MPE     MAPE      MASE
## Training set   0.4305478  30.20989  24.13111 -1.674696 12.80990 0.6744065
## Test set     131.4833377 165.04725 132.33172 32.586123 33.03047 3.6983538
##                    ACF1 Theil's U
## Training set -0.0375223        NA
## Test set      0.8789409  2.781427
```

```
print("ETS Gonorrhea")
```

```
## [1] "ETS Gonorrhea"
```

```
accuracy(frct2, actual.G)
```

```
##                       ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.1196048 30.09862 25.09183 -1.759791 13.26588 0.7012564
## Test set     20.4534504 57.54668 44.91692  2.240308 12.26379 1.2553201
##                    ACF1 Theil's U
## Training set 0.09277948        NA
## Test set     0.67358090  1.007027
```

We observe smaller root mean square error (RMSE) and smaller mean absolute error (MAE) for ETS models. The ETS models performed better than ARIMA models for forecasting San Diego STD cases.