time_series_analysis

Setup

Define working directory where the data is stored.

```
setwd("C:/Users/marce/OneDrive/Education/ETH/Autumn 2021/Applied Financial Analytics for Strategic Deci
```

Load neccessary R packages

```
# install.packages("forecast")
# install.packages("lubridate")
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.5

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

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.5

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Load the time series from disk

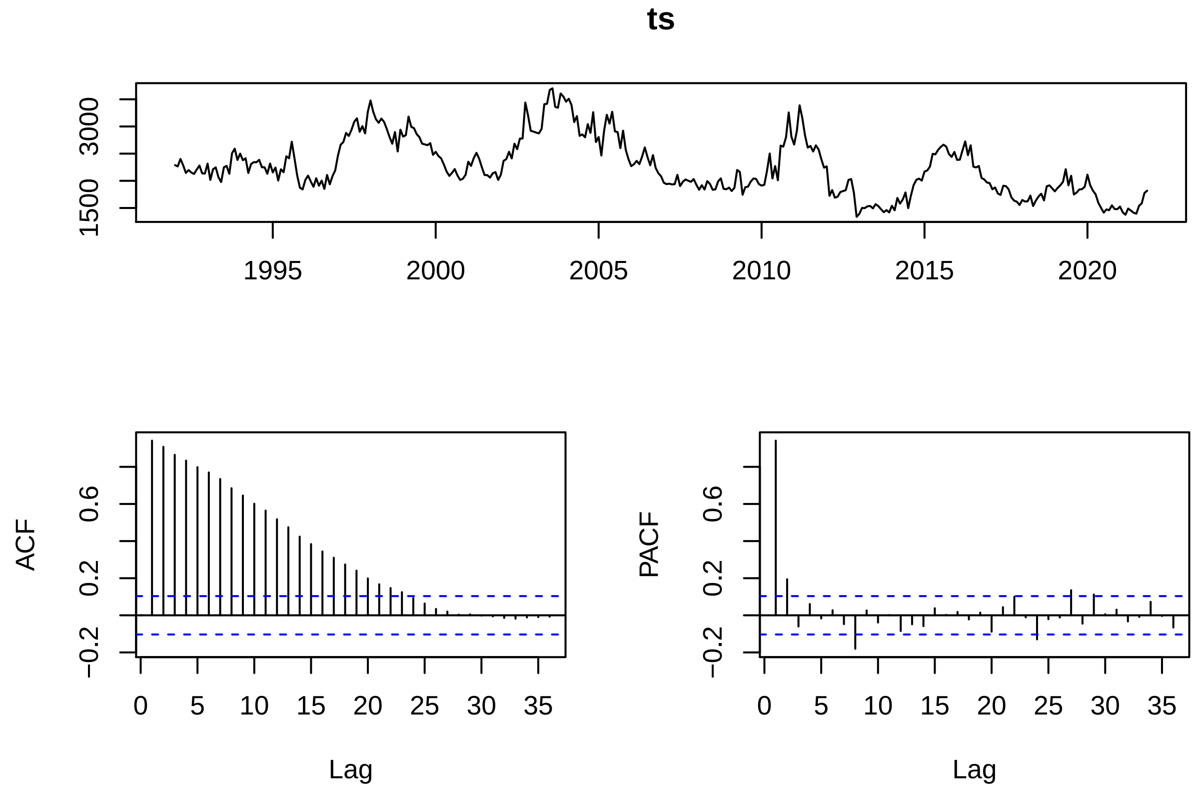
```
# load cocoa price with units US-Dollar/Ton
dat <- read.csv("cocoa_price_monthly.csv",header=TRUE, sep = ";")
# define second date column which contains last date of each month
dat[,"Last_Day_Month"] = ceiling_date(as.Date(dat[,1], format = "%d.%m.%Y"))
```

```r
# load exchange rate Us-Dollar/CHF to convert the cocoa price to CHF/ton unit
dollar_frank_rate <- read.csv("usd-chf.csv",header=TRUE, sep = ";", row.names=NULL)
row.names(dollar_frank_rate) <- format(as.Date(dollar_frank_rate[,"INDEX"]), "%d.%m.%Y")
# extract the exchange rates for each day where we observe a cocoa price
dollar_frank_rate_matched <- dollar_frank_rate[dat[,1],]
# correct mistakes because of mismatch between dates
dollar_frank_rate_matched["NA","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="2021-06-01","US
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.05.2021"
dollar_frank_rate_matched["NA.1","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="2011-12-30",
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.12.2011"
dollar_frank_rate_matched["NA.2","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="2010-06-01",
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.05.2010"
dollar_frank_rate_matched["NA.3","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="2004-06-01",
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.05.2004"
dollar_frank_rate_matched["NA.4","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="1999-06-01",
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.05.1999"
dollar_frank_rate_matched["NA.5","USD.CHF"] = dollar_frank_rate[dollar_frank_rate$INDEX =="1993-06-01",
rownames(dollar_frank_rate_matched)[rownames(dollar_frank_rate_matched)=="NA"] = "31.05.1993"

# multiply the cocoa price with the exchange rate to yield the CHF/Ton Cocoa Price
dat[,2] = dat[,2]*dollar_frank_rate_matched[,"USD.CHF"]


# convert time series data to time series object
ts <- ts(dat[,2], start=1992, frequency=12)
# apply first difference transformation to the cocoa price data
ts_diff <- diff(ts)
```

Time Series Analysis over full time horizon (1992-2021)

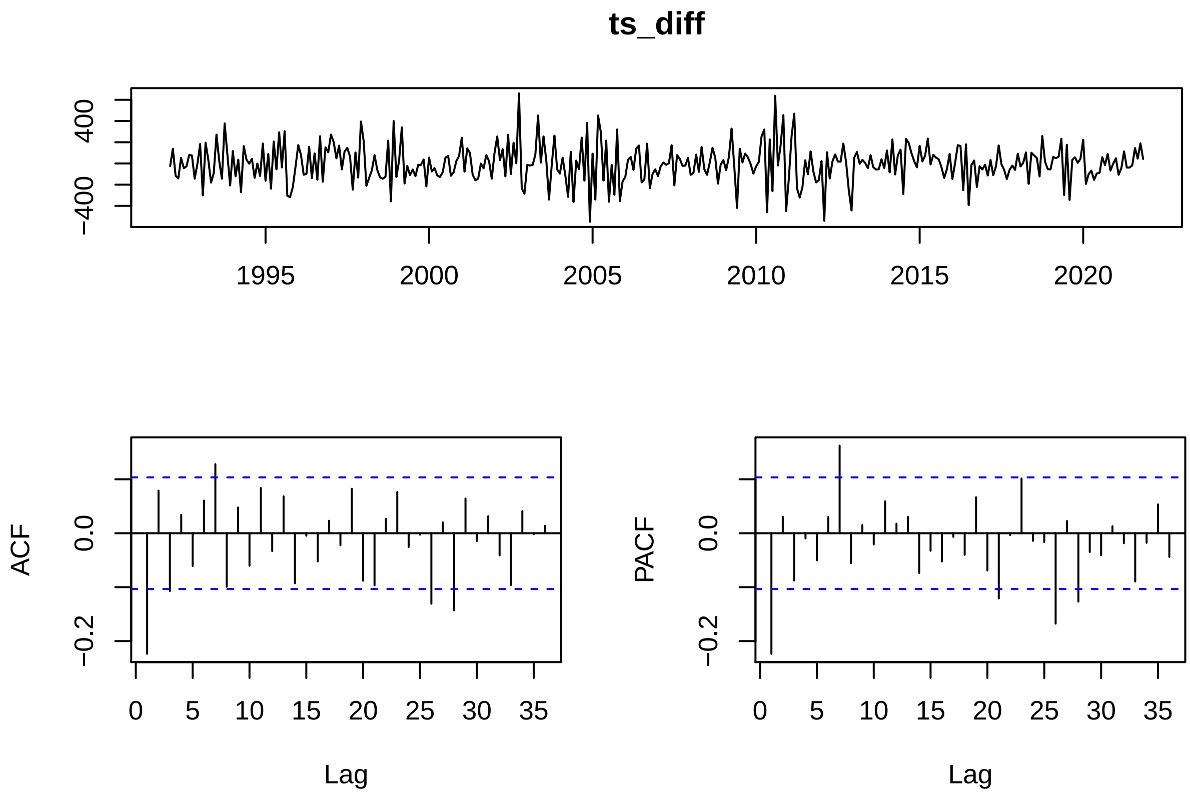
Plot time series and ACF and PACF function to investigate stationarity assumption.

```r
tsdisplay(ts, points = FALSE)
```



non-stationary

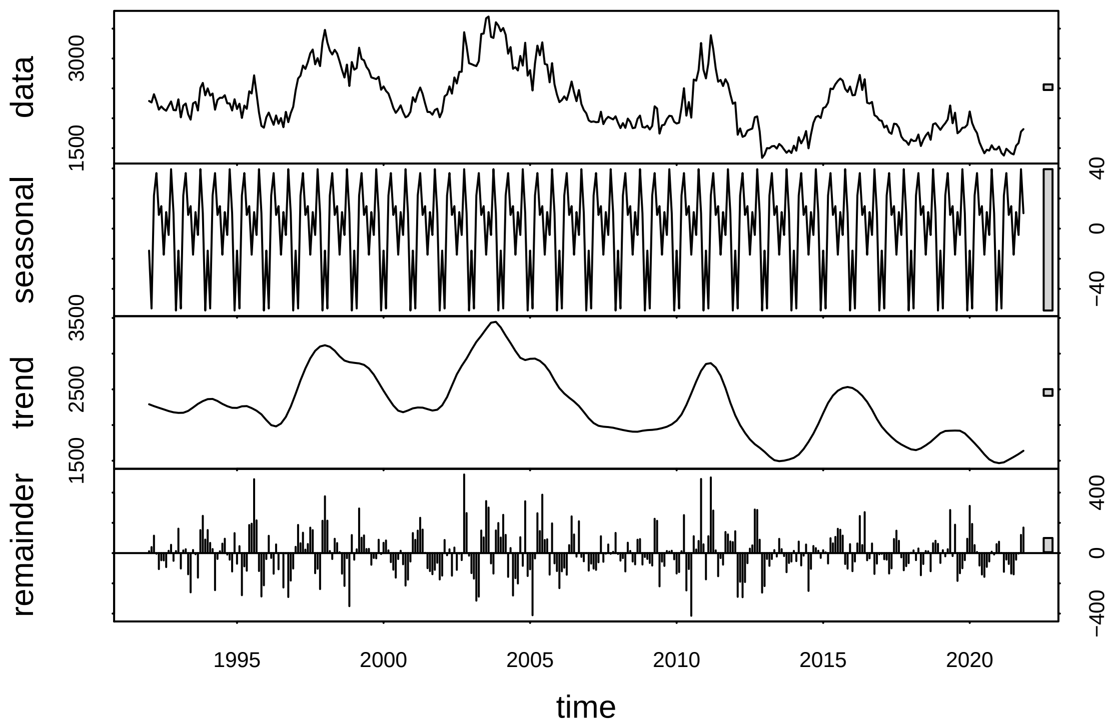
```
tsdisplay(ts_diff, points = FALSE)
```



–> better but still not fully stationary.

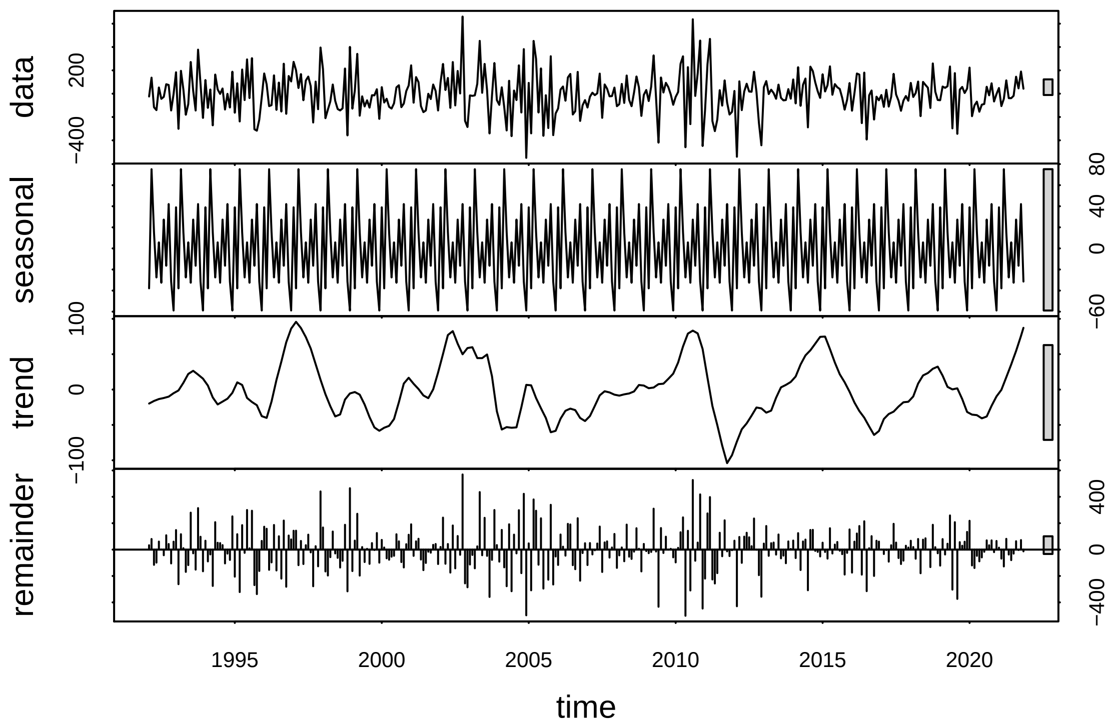
Plot automatic decomposition of time series into seasonal, trend and remainder component

```
H.stl <- stl(ts, s.window="periodic")
plot(H.stl)
```



–> seasonal component seems to dominate + a slight trend componenet

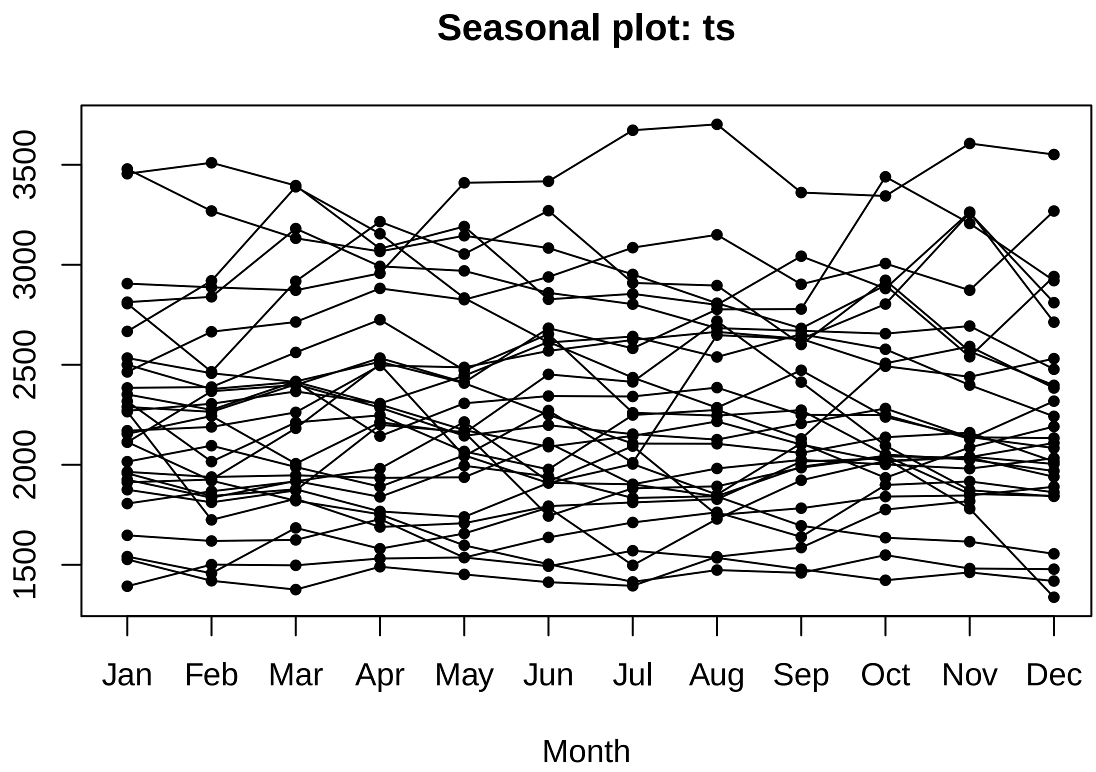
```
H.stl_diff <- stl(ts_diff, s.window="periodic")
plot(H.stl_diff)
```



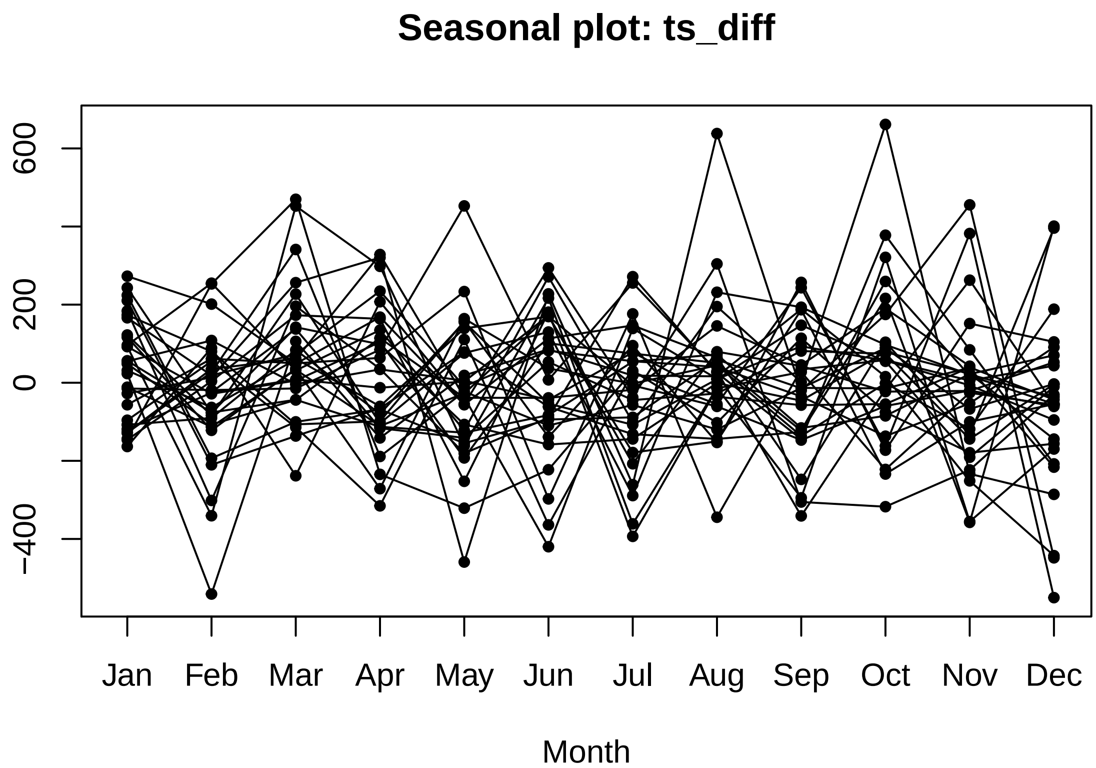
–> seasonal component seems to dominate + a slight trend componenet

Inspect seasonality component in further detail

```
seasonplot(ts, pch=20)
```

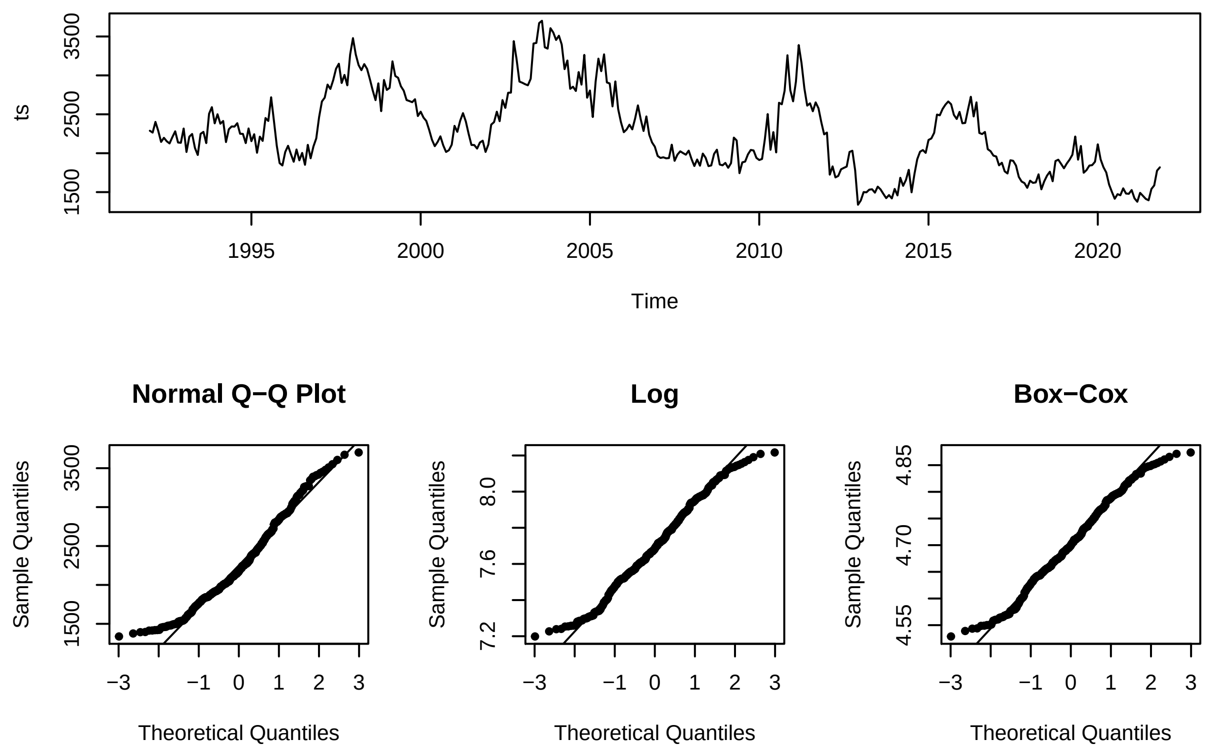



```
seasonplot(ts_diff, pch = 20)
```



Try out some transformations on the data too see if it leads to less violations of the stationarity assumption

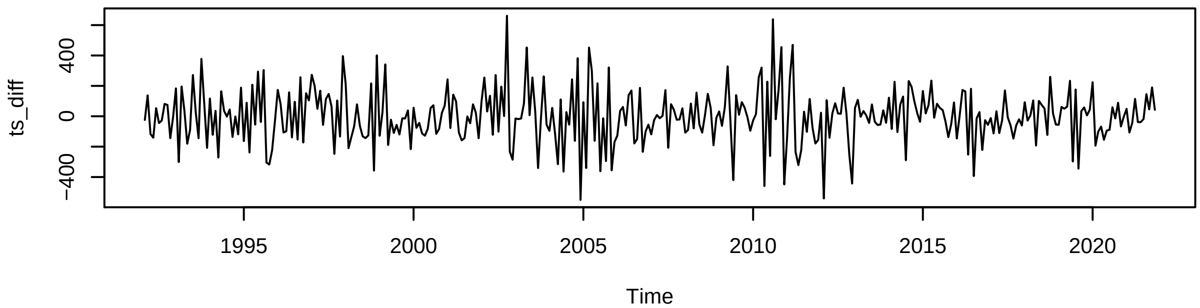
```
layout(matrix(c(1, 1, 1, 2, 3, 4), 2, 3, byrow = TRUE))
plot(ts)
qqnorm(ts, pch=20); qqline(ts)
qqnorm(log(ts), pch=20, main="Log"); qqline(log(ts))
tdf <- BoxCox(ts, lambda="auto")
qqnorm(tdf, pch=20, main="Box-Cox"); qqline(tdf)
```

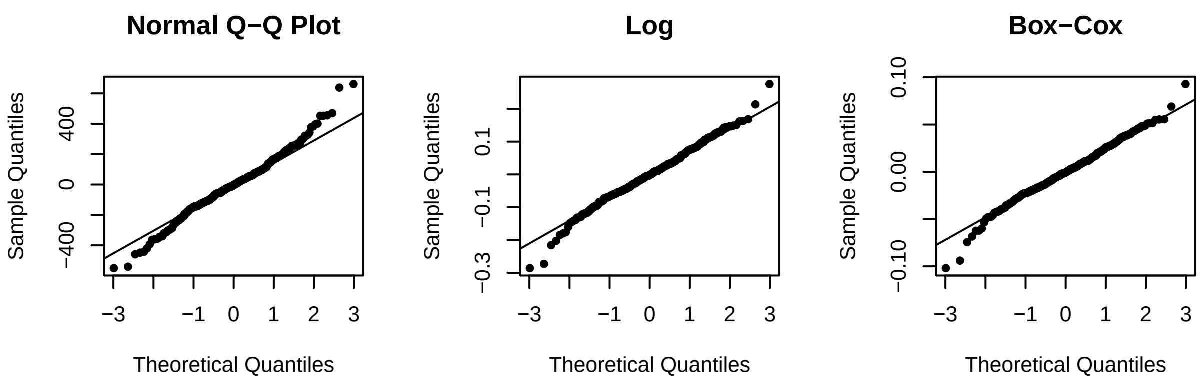


–> none of the transformations really helps

Try out some transformations on the differenced time series data too see if it leads to less violations of the stationarity assumption.

```
layout(matrix(c(1, 1, 1, 2, 3, 4), 2, 3, byrow = TRUE))
plot(ts_diff)
qqnorm(ts_diff, pch=20); qqline(ts_diff)
qqnorm(diff(log(ts)), pch=20, main="Log"); qqline(diff(log(ts)))
tdf <- diff(BoxCox(ts, lambda="auto"))
qqnorm(tdf, pch=20, main="Box-Cox"); qqline(tdf)
```






```r
# reset plot layout
par(mfrow=c(1,1))
```

–> log and Box-Cox transformation would lead to more plausible assumption of normally distributed time series data

Time Series Analysis over reduced time horizon (2013-2021)

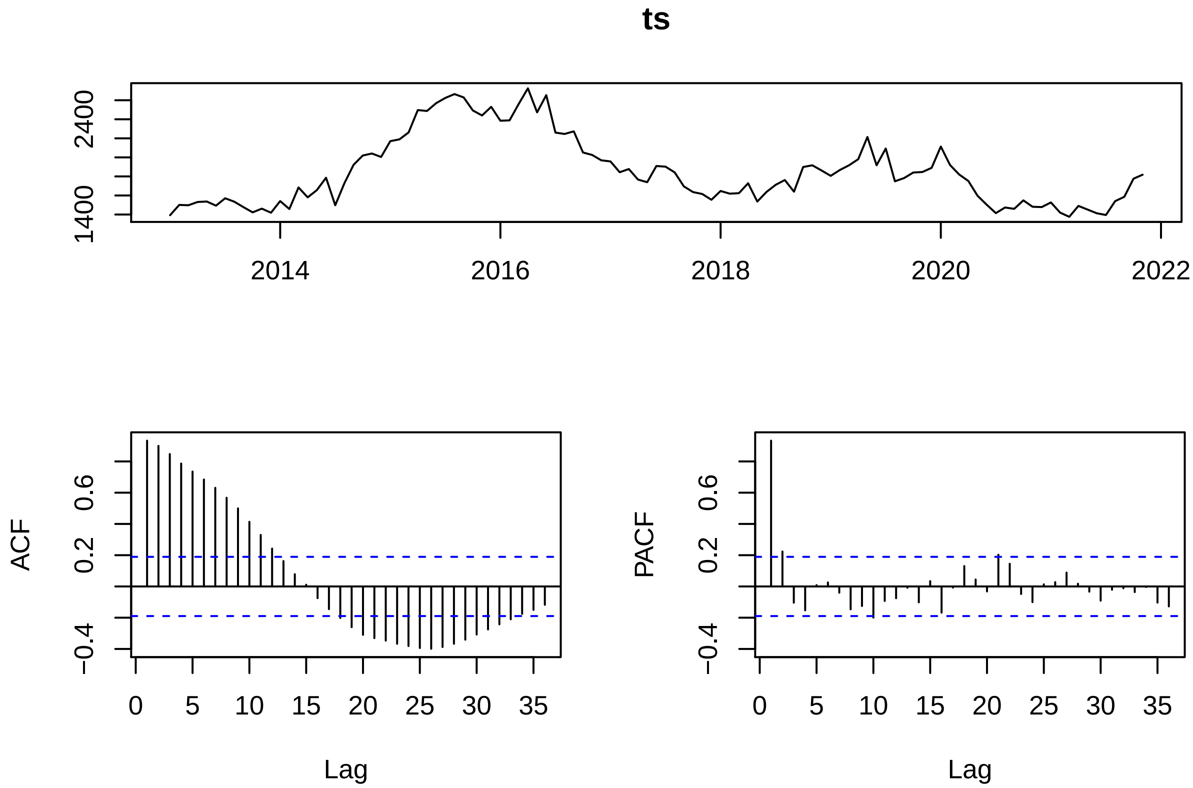
Check the stationarity of the time period that was identified during the search for the optimal time period and corresponding optimal model.

Adjust the time period to investigate

```r
ts = window(ts, 2013)
ts_diff = window(ts_diff, 2013)
```

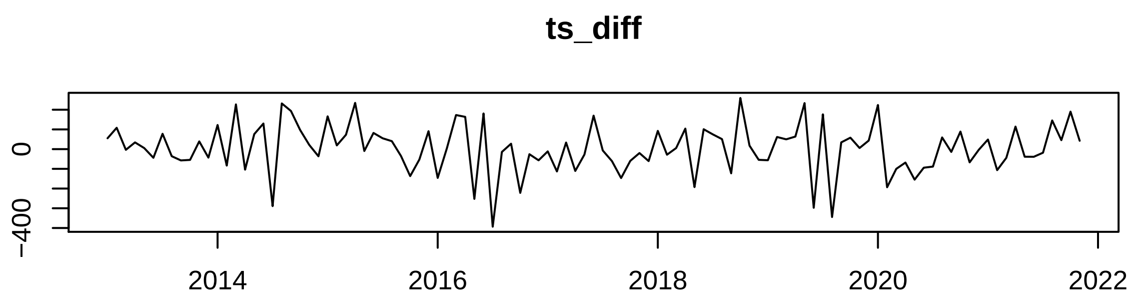
Plot time series and ACF and PACF function to investigate stationarity assumption.

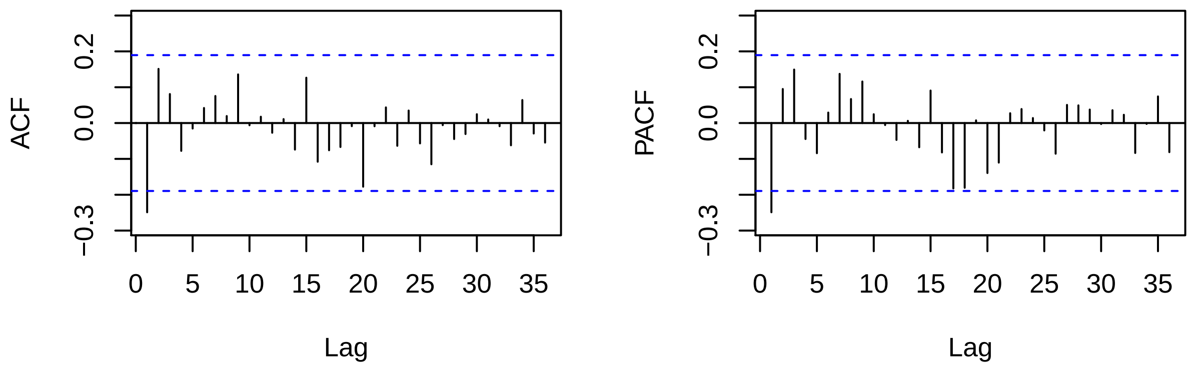
```r
tsdisplay(ts, points = FALSE)
```



–> non-stationary

```
tsdisplay(ts_diff, points = FALSE)
```

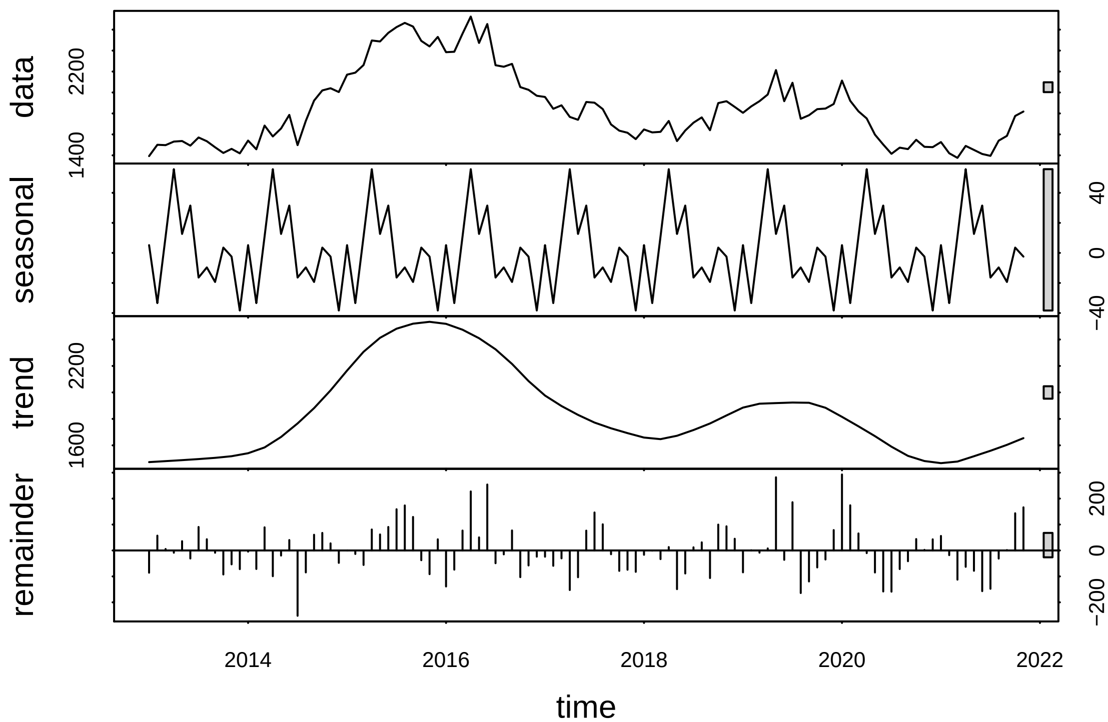




–> could be argued to be stationary considering the lag structure for a ARMA process

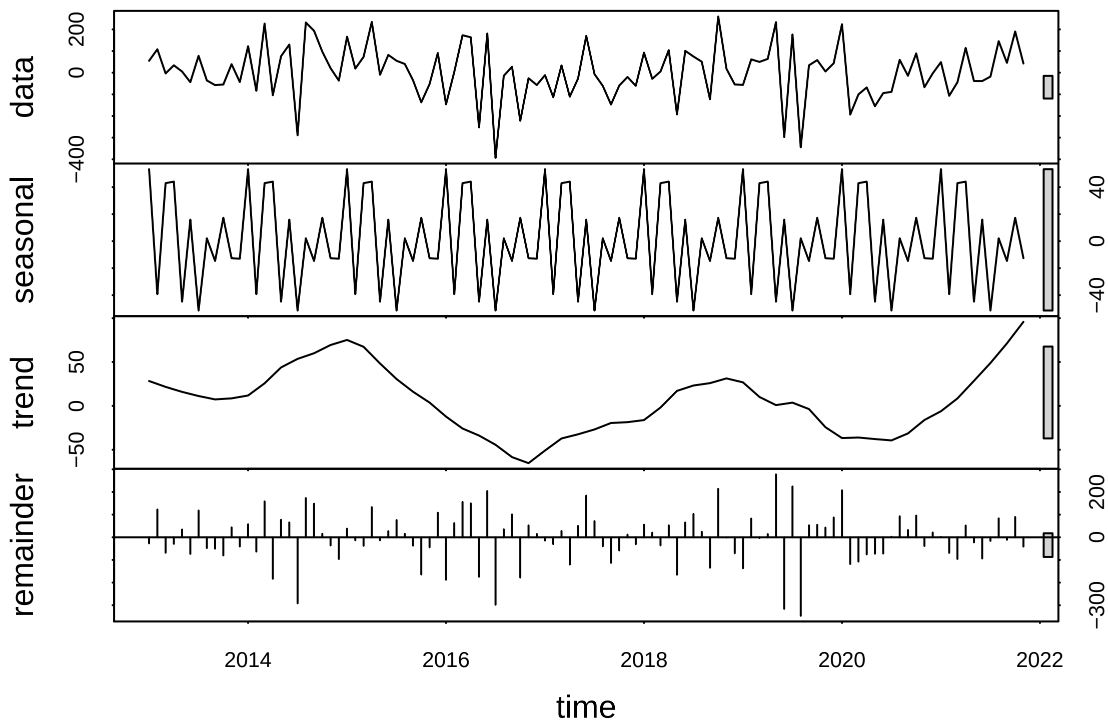
Plot automatic decomposition of time series into seasonal, trend and remainder component

```
H.stl <- stl(ts, s.window="periodic")
plot(H.stl)
```



–> seasonal component seems to dominate + a slight trend component

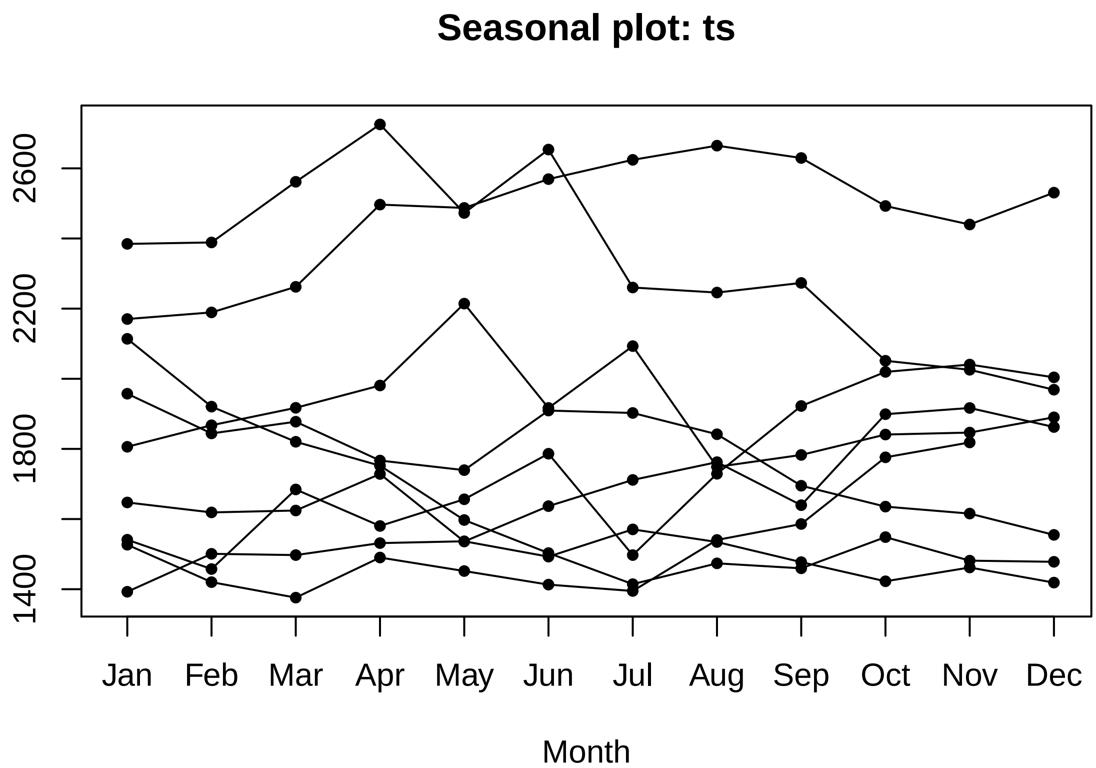
```
H.stl_diff <- stl(ts_diff, s.window="periodic")
plot(H.stl_diff)
```



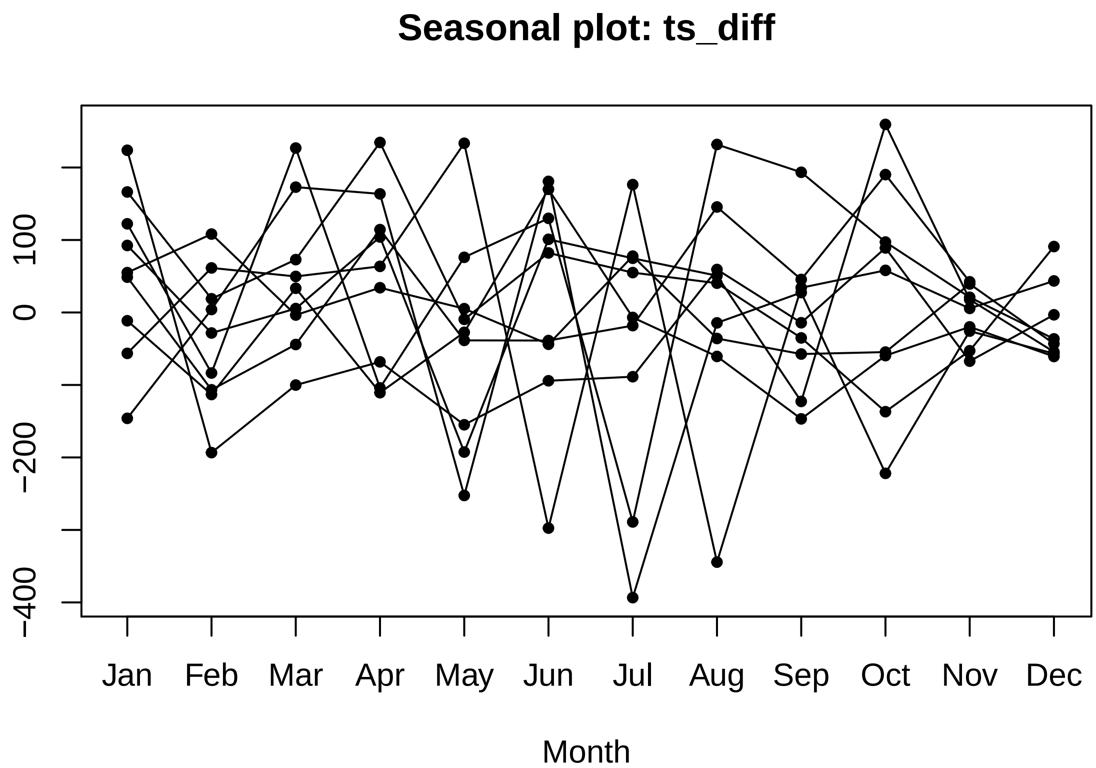
–> –> seasonal component + a trend component

Inspect seasonality component in further detail

```
seasonplot(ts, pch=20)
```

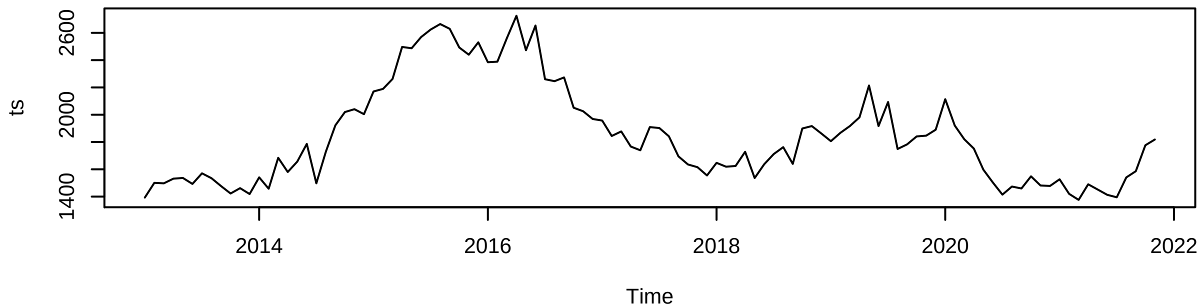


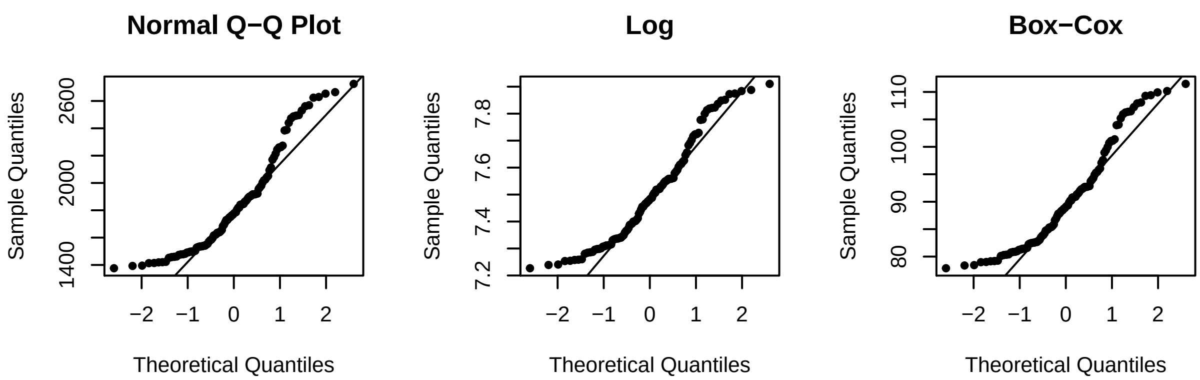

```
seasonplot(ts_diff, pch = 20)
```



Try out some transformations on the data too see if it leads to less violations of the stationarity assumption

```
layout(matrix(c(1, 1, 1, 2, 3, 4), 2, 3, byrow = TRUE))
plot(ts)
qqnorm(ts, pch=20); qqline(ts)
qqnorm(log(ts), pch=20, main="Log"); qqline(log(ts))
tdf <- BoxCox(ts, lambda="auto")
qqnorm(tdf, pch=20, main="Box-Cox"); qqline(tdf)
```

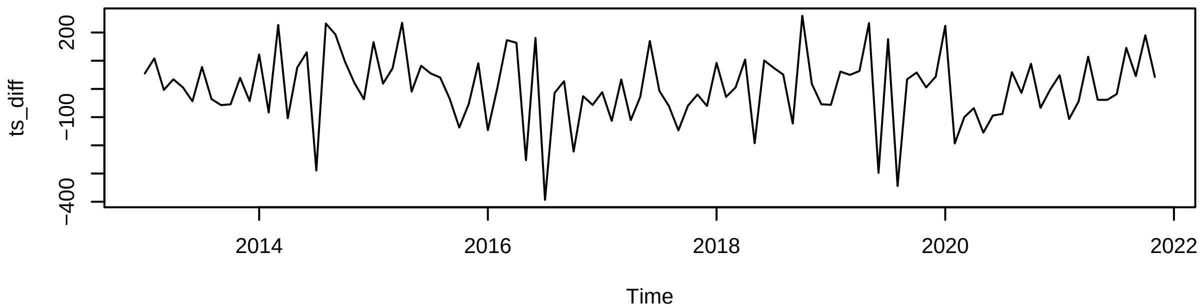


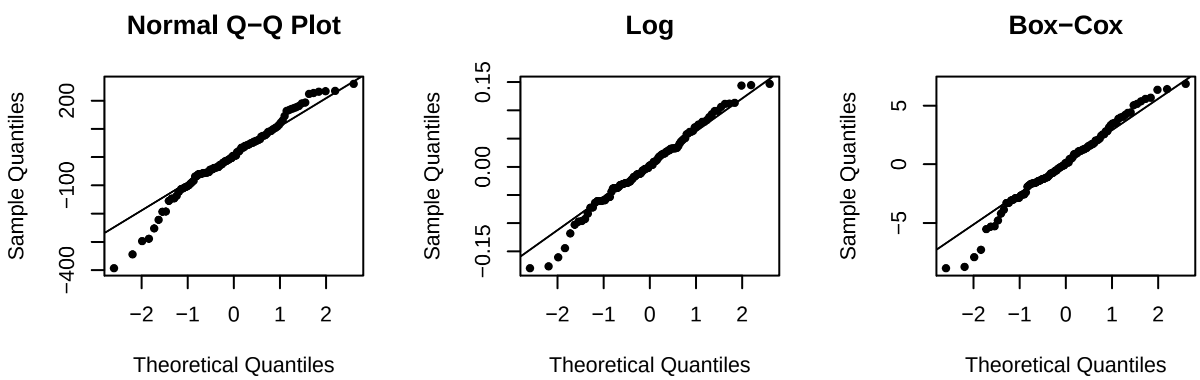


–> none of the transformations really helps

Try out some transformations on the differenced time series data too see if it leads to less violations of the stationarity assumption.

```
layout(matrix(c(1, 1, 1, 2, 3, 4), 2, 3, byrow = TRUE))
plot(ts_diff)
qqnorm(ts_diff, pch=20); qqline(ts_diff)
qqnorm(diff(log(ts)), pch=20, main="Log"); qqline(diff(log(ts)))
tdf <- diff(BoxCox(ts, lambda="auto"))
qqnorm(tdf, pch=20, main="Box-Cox"); qqline(tdf)
```






```
# reset plot layout
par(mfrow=c(1,1))
```

–> all of the three QQ-plots seem to provide no obvious evidence against a normal distribution assumption

Automatically define Model to predict the future cocoa price and evaluate its performance

Divide data into training and testing window to evaluate performance

```
ts_train = window(ts, 2013, c(2020,10))
ts_test = window(ts, c(2020,11))
```

Fit ARMA model

In the preceding analysis we saw that first-order differencing helped to remove non-stationary patterns in the time series / a possible linear trend in the time series. Thus, we either apply first-order differencing before modelling or we allow for a non-zero mean when estimating the model (both equivalent). Criteria for modelling the time series as an AR-Process: Time series needs to be stationary, shows an ACF with approximately exponentially decaying envelope and a PACF with a recognizable cut-off at some lag p smaller than about 5-10. Criteria for modelling the time series as an AR-Process: Time series needs to be stationary, shows an PACF with approximately exponentially decaying envelope and a ACF with a recognizable cut-off at some lag q smaller than about 5-10. Criteria for modelling the time series as an ARMA-Process: neither the ACF nor the PACF clearly show a cut-off strictly at a certain lag. Instead, they both show some infinite behavior, i.e. an exponential decay in the magnitude of the coefficients.However, superimposed on that is a sudden drop-off in both ACF and PACF. We analyze the ACF and the PACF plot for the differenced time series for the time period 2013-most recent date and observe the following: Both, the ACF and the PACF plot shows some infinite behavior. However, one could state that the first and the second autocorrelation coefficient and the first partial autocorrelation coefficient is higher than the other, which could indicate a superimposed cut-off. This would lead us to try out an ARMA-model with non-zero mean and a lag of 2 for the MA-component and a lag 1 for the AR-component.

```
fit_arma <- arima(ts_train, order = c(1,0,2), include.mean = TRUE)
fit_arma
```

```
##
## Call:
## arima(x = ts_train, order = c(1, 0, 2), include.mean = TRUE)
##
## Coefficients:
##          ar1      ma1     ma2  intercept
##       0.9425  -0.2527  0.2836   1781.273
## s.e.  0.0354   0.1013  0.1159    198.209
##
## sigma^2 estimated as 14532:  log likelihood = -585.05,  aic = 1180.09
```

We now check if the automatic arima modelling procedure would yield a similar model:

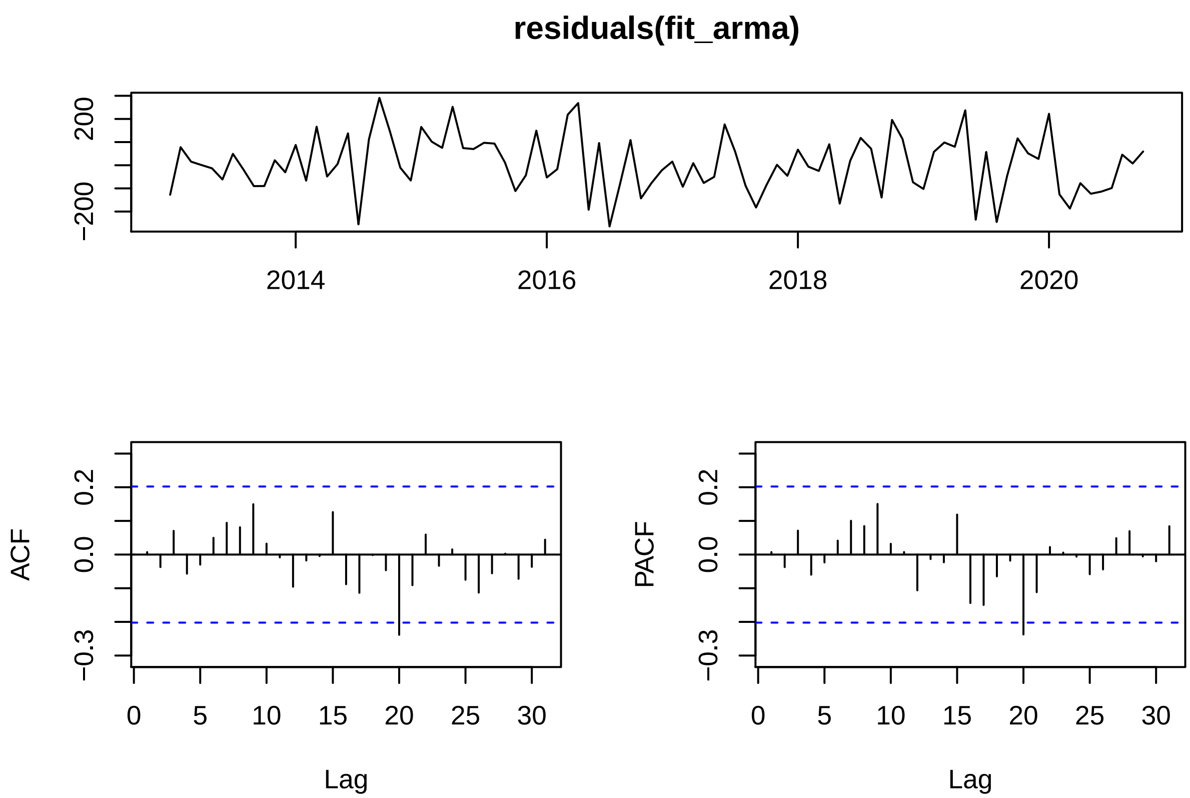
```
fit_auto_arma <- auto.arima(ts_train, max.p=5, max.q=5,
                  stationary=TRUE, allowmean=TRUE,
                  stepwise=FALSE, ic="aic")
fit_auto_arma
```

```
## Series: ts_train
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ma1     ma2      mean
##       0.9425  -0.2527  0.2836  1781.273
## s.e.  0.0354   0.1013  0.1159   198.209
##
## sigma^2 estimated as 15178:  log likelihood=-585.05
## AIC=1180.09   AICc=1180.78   BIC=1192.81
```

Indeed, the same model specification would result from the automatic model specification. We continue by analyzing the prediction trajectory and the residuals.

Analysis of the resiudals

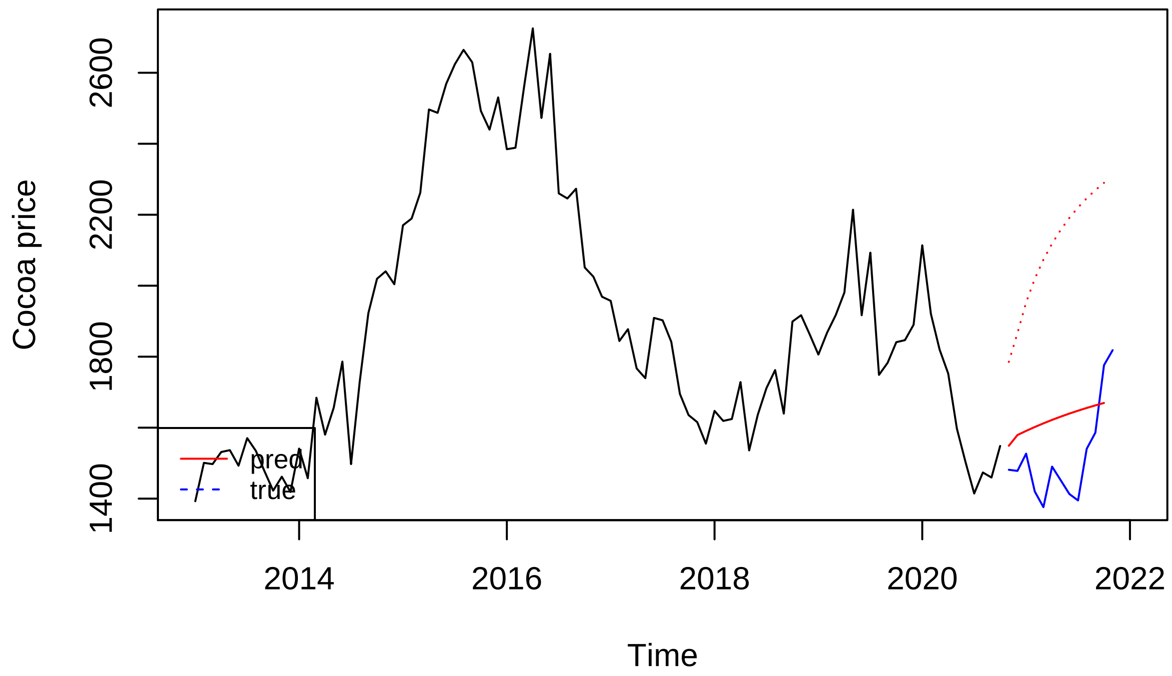
```
tsdisplay(residuals(fit_arma), points=FALSE)
```



Residual Analysis gives no clear reason to suspect that the model assumptions are violated or that the model poorly fits. The high ACF / PACF at lag 20 can be neglected.

Plot predicted cocoa prices against true prices

```
plot(ts_train, xlim=c(2013,2022), ylab="Cocoa price")#, xlim=c(0,250), ylim=c(-10,15), main="..."
lines(ts_test, col="blue")
pred <- predict(fit_arma, n.ahead=12)
lines(pred$pred, col="red")
lines(pred$pred + 1.96*pred$se, col="red", lty=3)
lines(pred$pred - 1.96*pred$se, col="red", lty=3)
legend("bottomleft", legend=c("pred", "true"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



Evaluate precision of the model - 12 Months

```
mape = mean(100*abs((ts_test-pred$pred)/ts_test));mape
```

```
## [1] 9.937414
```

```
mae <- mean(abs(ts_test-pred$pred)); mae
```

```
## [1] 144.9917
```

```
rmse <- sqrt(mean((ts_test-pred$pred)^2)); rmse
```

```
## [1] 158.9276
```

Evaluate precision of the model - 6 Months

```
mape = mean(100*abs((ts_test[1:6]-pred$pred[1:6])/ts_test[1:6]));mape
```

```
## [1] 9.066958
```

```r
mae <- mean(abs(ts_test[1:6]-pred$pred[1:6])); mae
```

```
## [1] 130.405
```

```r
rmse <- sqrt(mean((ts_test[1:6]-pred$pred[1:6])^2)); rmse
```

```
## [1] 144.3676
```

Evaluate precision of the model - 3 Months

```r
mape = mean(100*abs((ts_test[1:3]-pred$pred[1:3])/ts_test[1:3]));mape
```

```
## [1] 5.202375
```

```r
mae <- mean(abs(ts_test[1:3]-pred$pred[1:3])); mae
```

```
## [1] 77.62777
```

```r
rmse <- sqrt(mean((ts_test[1:3]-pred$pred[1:3])^2)); rmse
```

```
## [1] 79.40045
```