HW4

Question 7.1

I currently work at an automation company that manufactures packaging machinery. Our customers that use these machines will normally track day to day or shift to shift production metrics. If we have data collected for day1-shiftA, day1-shiftB, day2-shiftA, day2-shiftB, etc. and production metric for each point (18,000 products produced), we would have a time series model.

We can then look at this model to see the baseline for typical expected production throughput, any trends that can be investigated (are operators getting better over time as they get more familiar with machines? Is machine performance getting worse as a certain part wears out?). We can also look for a seasonal component which could indicate whether a particular shift (A or B) performs better than the other over the span of months.

I would expect alpha to be closer to 0. There would be a lot of randomness in the system as you can have machine downtime, stoppages, or rejects for a wide variety of reasons. Some days you can happen to have more issues so you would rather trust your baseline.

Question 7.2

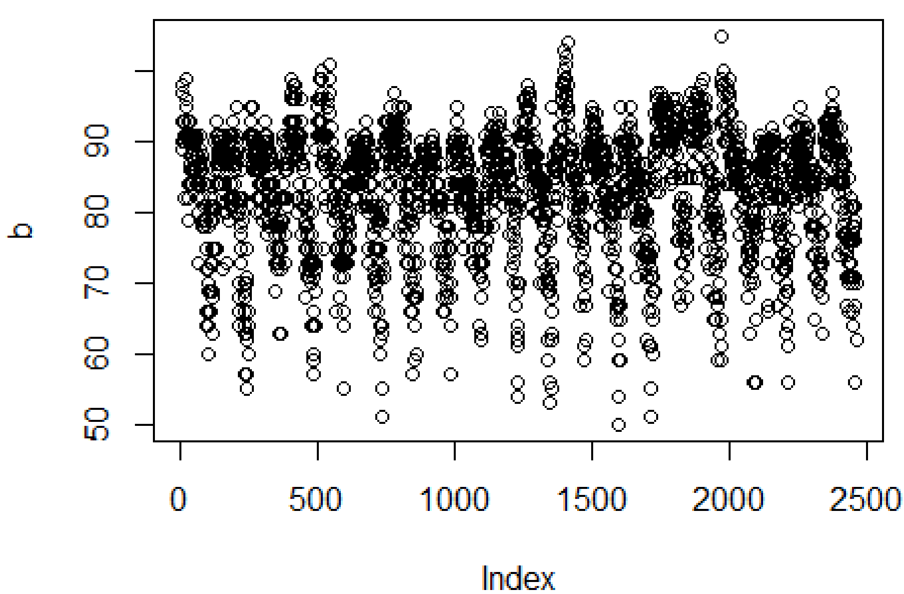
Given the data we are dealing with, we would expect to have a seasonal component, but don't know about trend. We run HoltWinters using triple exponential smoothing and double. When running triple exponential smoothing, trend graph is a straight line and optimal beta value is 0 indicating there is no trend.

Given this, we can export the seasonal component of the data from the double exponential smoothing model (alpha and gamma). When performing CUSUM on this we would notice a trend for when change is detected throughout the years if summer is starting later.

However, CUSUM on the seasonal component **does not show any noticeable trend** with the data we have. It goes down after the first couple years, then stays consistent and has minor variations afterwards.

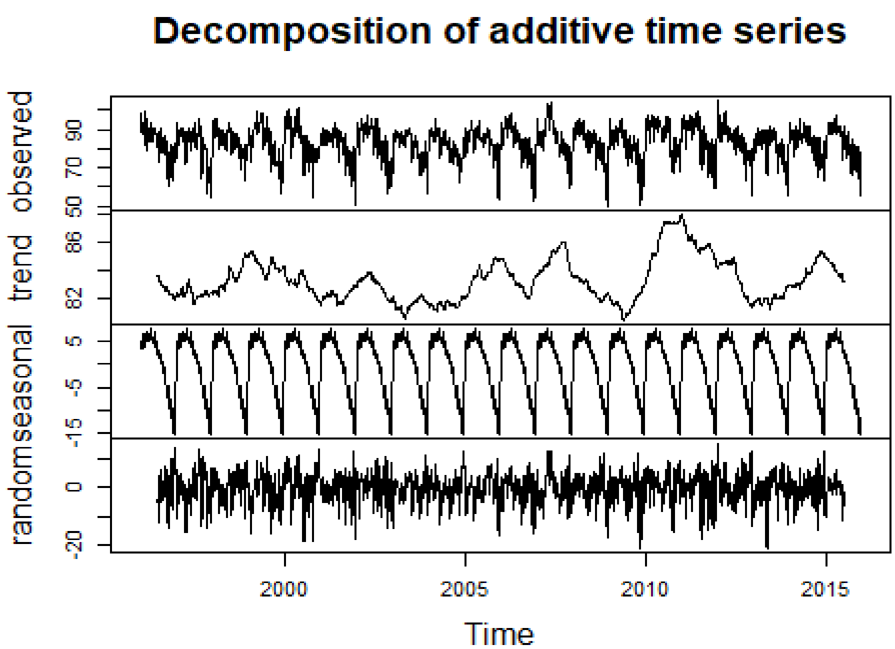
27-Sep	0.115421	0.114851	0.101021	0.084205	0.075728	0.065006	0.077628	0.090617	0.067336	0.069028	0.076193	0.069767	0.067041	0.068809	0.100333	0.088144	0.067597	0.077411	0.071306
28-Sep	0.115421	0.114851	0.101021	0.084205	0.075728	0.065006	0.077628	0.090617	0.067336	0.069028	0.076193	0.069767	0.067041	0.068809	0.100333	0.088144	0.071742	0.077411	0.076111
29-Sep	0.12023	0.114851	0.101021	0.091228	0.083008	0.086362	0.084741	0.099872	0.083157	0.071031	0.090019	0.088666	0.071733	0.091979	0.100333	0.088144	0.076849	0.079374	0.087654
30-Sep	0.217086	0.169485	0.157364	0.157463	0.137158	0.127691	0.105387	0.099872	0.084523	0.08651	0.090019	0.088666	0.080651	0.091979	0.100333	0.098104	0.125749	0.115088	0.087654
01-Oct	0.217086	0.202422	0.170171	0.162509	0.149879	0.134373	0.118994	0.110611	0.099173	0.094473	0.090019	0.088666	0.105273	0.096714	0.100333	0.129998	0.127269	0.115088	0.09548
02-Oct	0.217086	0.202422	0.170171	0.162509	0.149879	0.134373	0.123523	0.131134	0.122838	0.11172	0.1049	0.095634	0.105273	0.104204	0.103284	0.129998	0.127269	0.117234	0.11223
03-Oct	0.217086	0.202422	0.170171	0.162509	0.149879	0.134373	0.123523	0.131134	0.122838	0.113911	0.10928	0.117868	0.105273	0.104204	0.129046	0.129998	0.127269	0.117234	0.126981
04-Oct	0.217086	0.202422	0.170171	0.162509	0.149879	0.134373	0.123523	0.131134	0.122838	0.113911	0.10928	0.117868	0.105273	0.104204	0.129046	0.129998	0.127269	0.117234	0.130774
05-Oct	0.217086	0.202422	0.170171	0.162509	0.149879	0.134373	0.123523	0.131134	0.122838	0.113911	0.10928	0.117868	0.105273	0.133544	0.129046	0.129998	0.127269	0.117234	0.130774
06-Oct	0.217086	0.202422	0.20716	0.197554	0.203684	0.209413	0.192393	0.181364	0.177708	0.185972	0.190298	0.171201	0.15822	0.13647	0.129046	0.129998	0.127269	0.117234	0.130774
07-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.226097	0.215855	0.21097	0.217143	0.20392	0.192965	0.161486	0.135265	0.138513	0.171079	0.182501	0.153495
08-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.226097	0.225694	0.219464	0.220748	0.216162	0.23174	0.211928	0.191647	0.184345	0.201581	0.205043	0.182208
09-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.211928	0.195969	0.206867	0.201581	0.205043	0.182208
10-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.212439	0.223602	0.208773	0.205043	0.192112
11-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.212439	0.223602	0.208773	0.206515	0.203021
12-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.203021
13-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.203021
14-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.219006
15-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126
16-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126
17-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126
18-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126
19-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126
20-Oct	0.263205	0.247761	0.223937	0.212209	0.228858	0.230823	0.232788	0.231977	0.235545	0.237834	0.220748	0.223944	0.23174	0.220424	0.220607	0.223602	0.208773	0.206515	0.223126

```
data <- read.table('temps.txt', header=TRUE, row.names=1)

#unrwaps df into one long vector
b <- unlist(data)

plot(b)
```

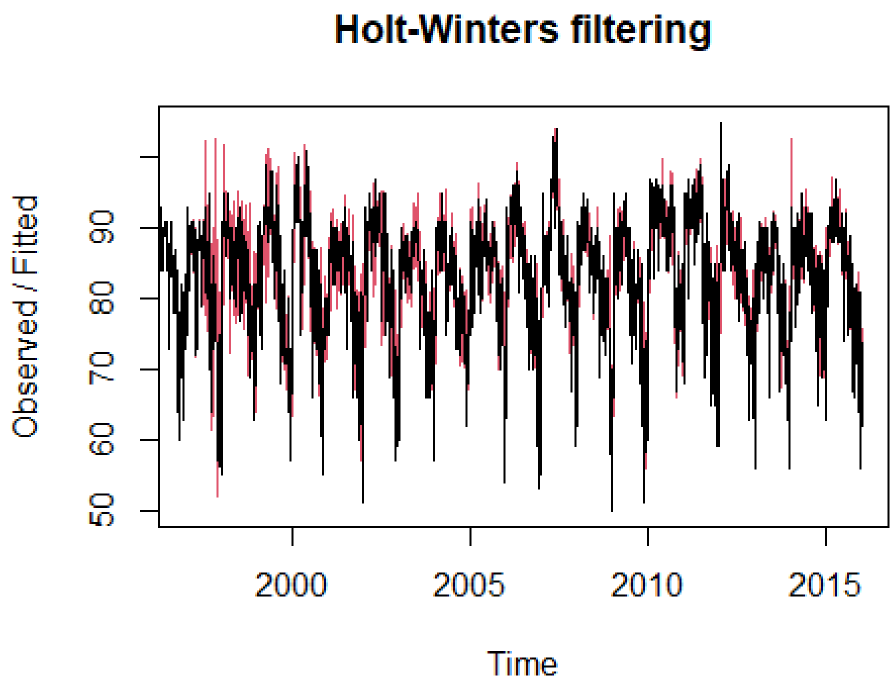


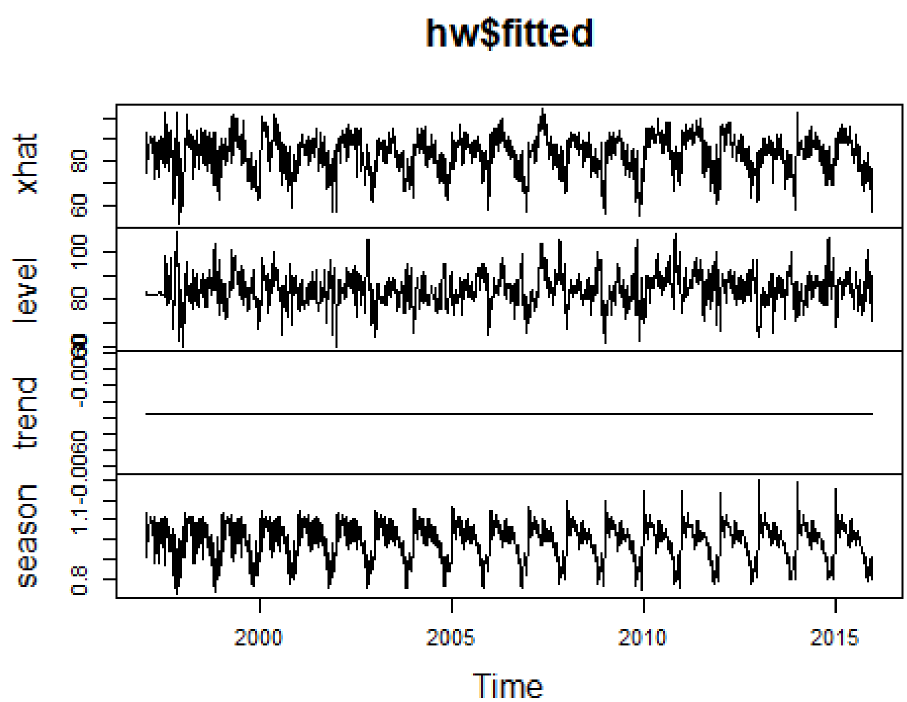

```
#convert to time series
#repeat after frequency 123 (123 days before year changes)
data_ts <- ts(b,start=1996, frequency=123)

#decompose uses moving average to assess trend & overall seasonality by avera
```

```
ging it over all periods
#Good for exploratory data analysis to see trends/seasonalities
#it tells you there is seasonality but does not tell you how seasonality is i
solated to a particular period
plot(decompose(data_ts))
```

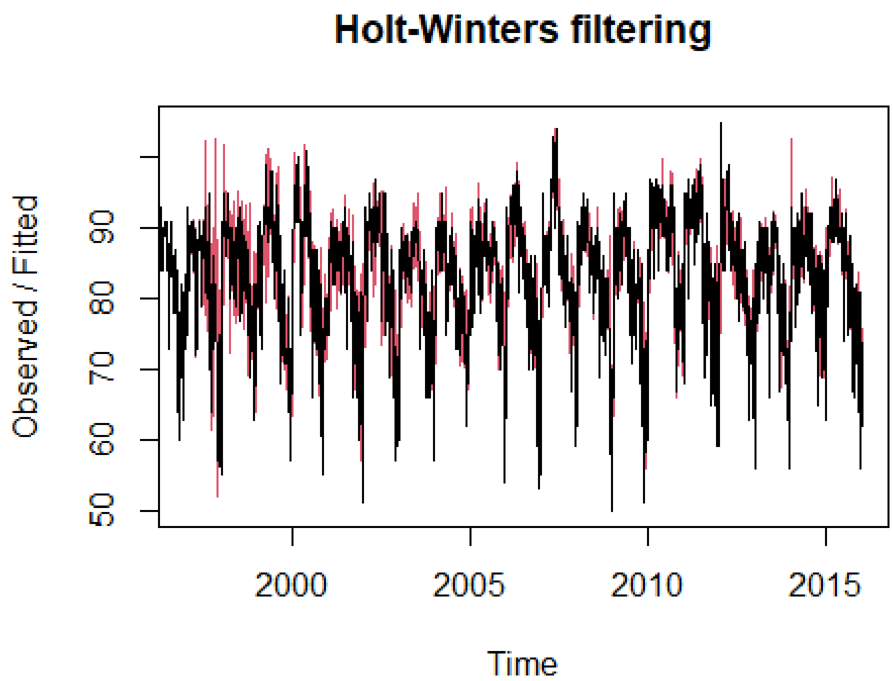


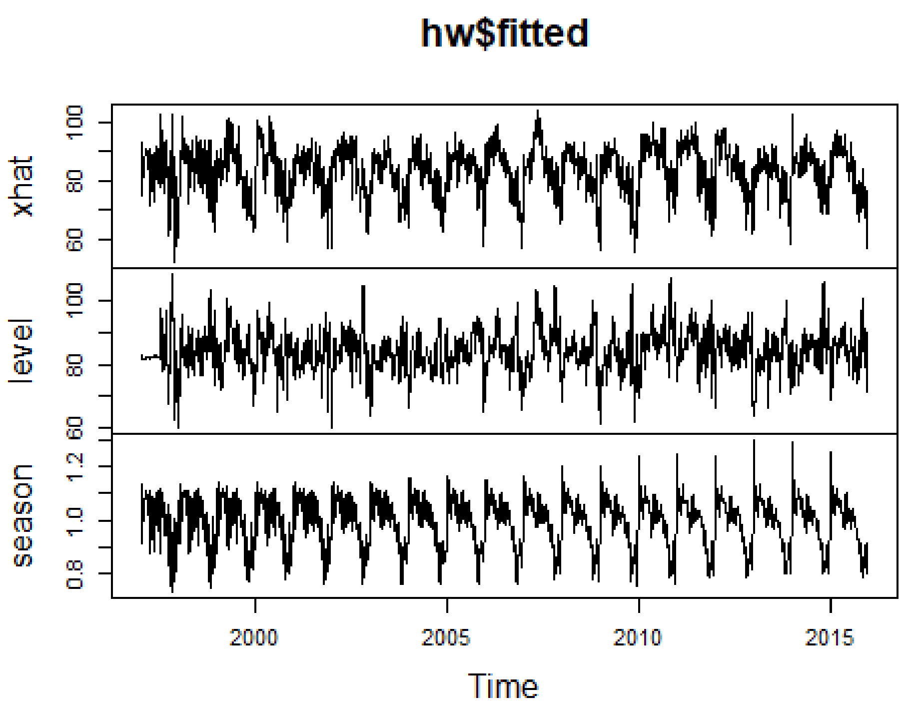

```
holtwinter <- function(a,b,g,seasonal){

  hw <- HoltWinters(data_ts,alpha=a,beta=b,gamma=g,seasonal=seasonal)
  print(hw)
  plot(hw)

  plot(hw$fitted)

  #print squared error
  cat("error: ",hw$SSE)

  return(hw$fitted)
}

hw_abg <- holtwinter(NULL,NULL,NULL,"multiplicative")
```





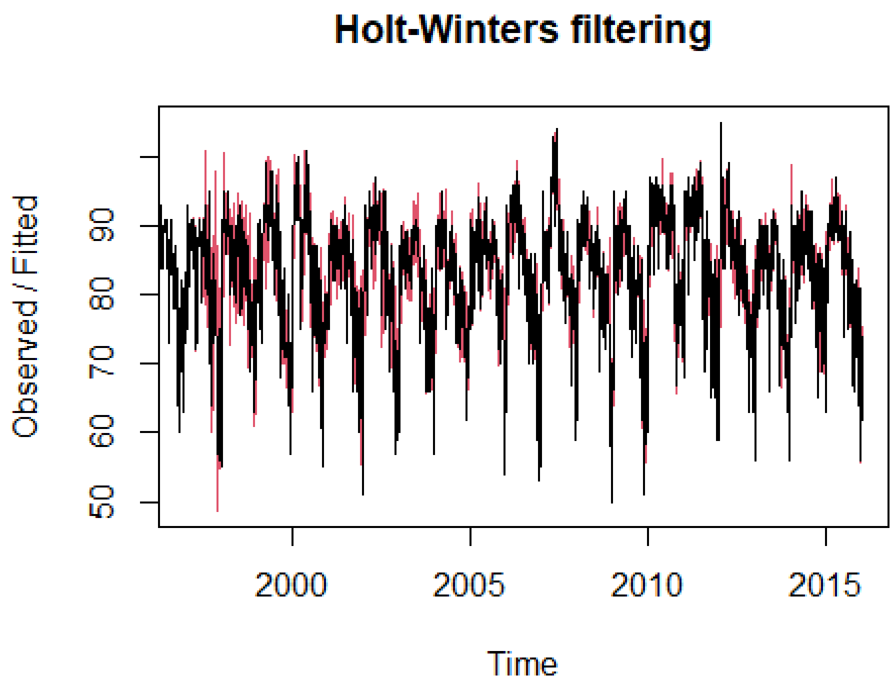

```
## error:  68904.57

hw_ag <- holtwinter(NULL,FALSE,NULL,"multiplicative")
```

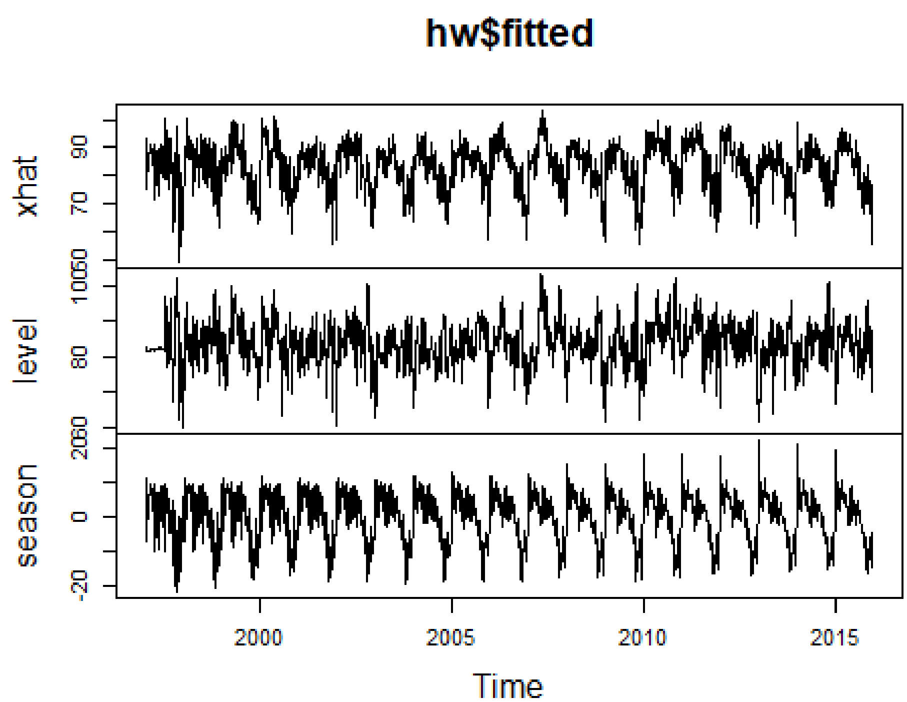





```
## error:  68905.32
```

```
hw_ag_add <- holtwinter(NULL,FALSE,NULL,"additive")
```






```
## error:  66244.54

#we pick hw_ag as there is no trend component in the temps data, but there is
a seasonal component
```

```r
#Extract xhat (the predicted value at time t) & seasonal data
#convert data back to rows and columns using our chosen Holtwinter model
xhat <- matrix(hw_ag[,1],nrow=123)
season <- matrix(hw_ag[,3],nrow=123)

#return xhat - predicted value at time t
write.csv(xhat,file="temp_xhat.csv")
write.csv(season,file="temp_season.csv")
```