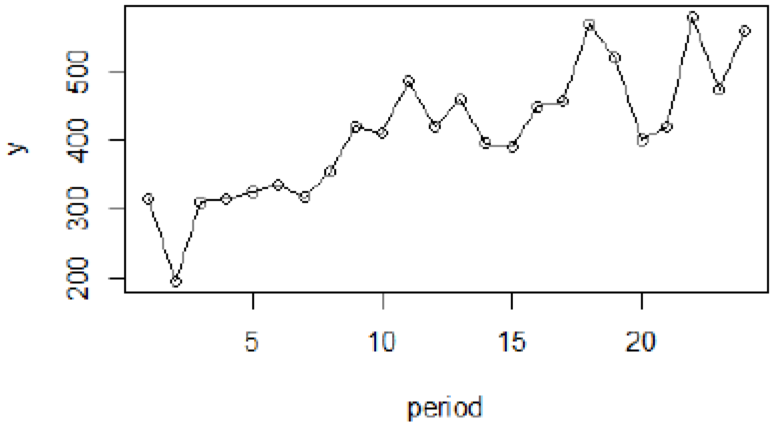
Time Series homework4

104304033 統計四 劉書宏

Chapter 4

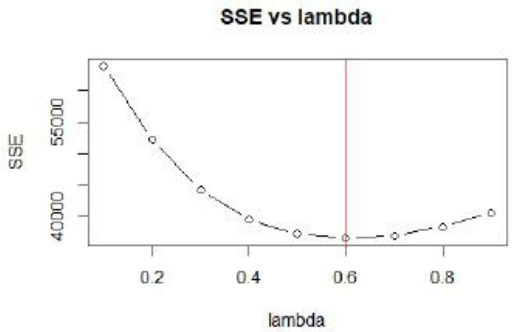
8.

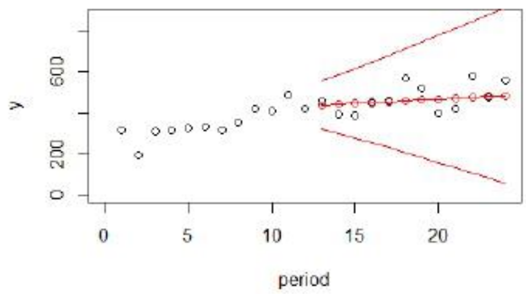
(a)



We can see that there is a going-up trend.

(b)





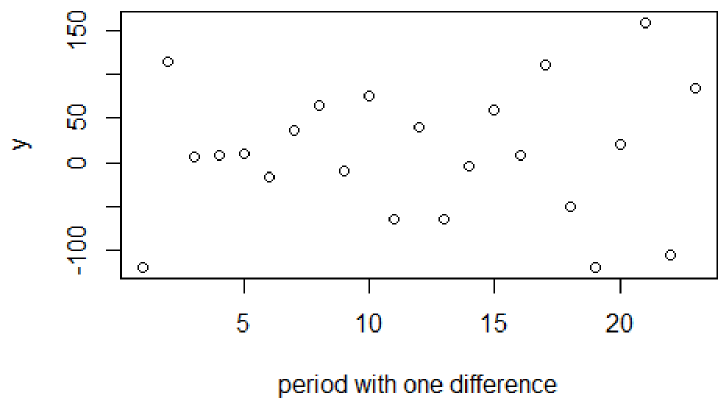
I calculated the best lambda, and using double exponential smoothing.

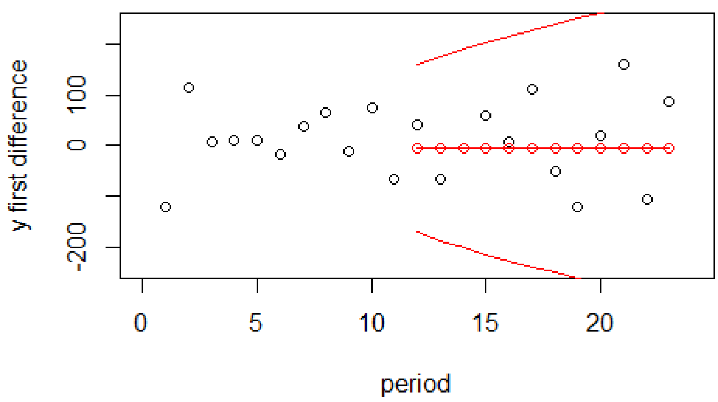
(c)

```
 [1]  -20.683340   48.325133   57.333607    1.342080   -2.649447 -110.640974  -56.632501   67.375973   51.384446
[10] -104.607081    4.401392  -76.590135
```

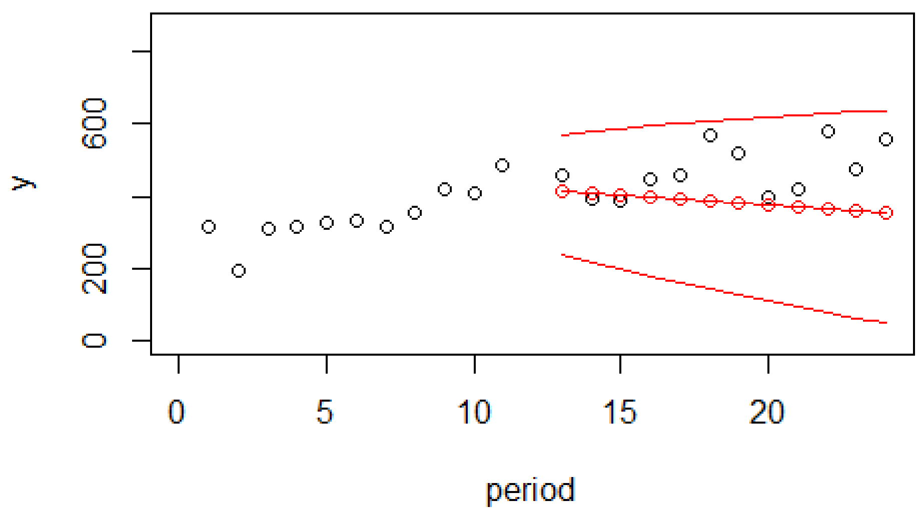
Here is the forcasing errors , and the errors change a lot from -110.640974 to 67.375973, so I think that the forecasting is a little bad.

9.





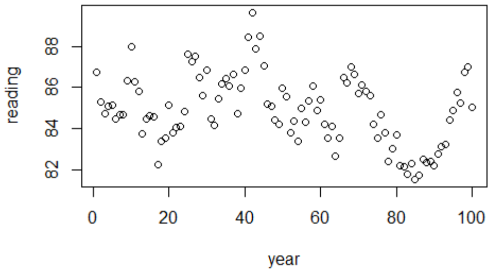
Taking the first difference , we still can’t see a trend at the plot, and forecasting are quite bad.

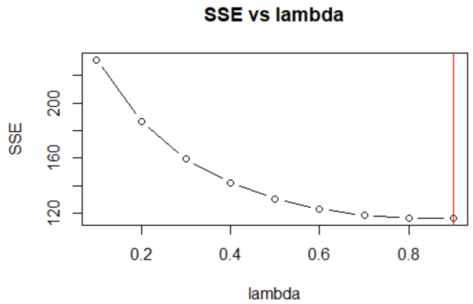


By using simple exponential smoothing , I think that the forecasting by using simple exponential smoothing are better than the forecasting bytaking the first difference.

15.

(a)



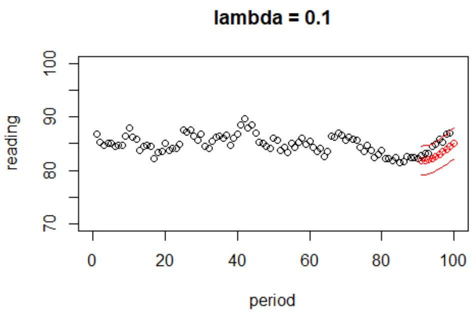


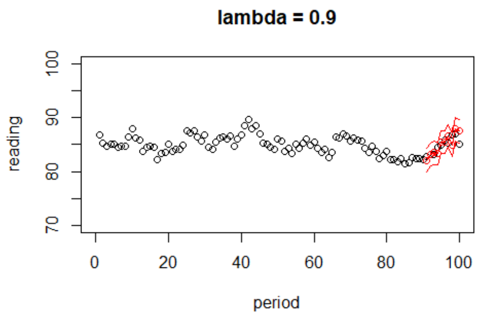
We can take the optimal value of lambda = 0.9

```
> print(measacc(data15[,2],0.1))
       SSE       MAPE        MAD        MSD
231.578730   1.511824   1.283660   2.315787
>   #lambda 0.9 error
> print(measacc(data15[,2],0.9))
        SSE        MAPE         MAD         MSD
116.0241921   1.0043571   0.8550947   1.1602419
```

We can see that using lambda = 0.9 is better than using lambda = 0.1

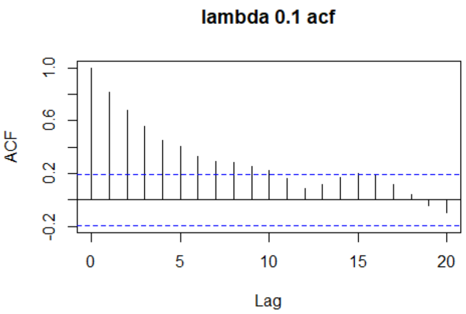
(b)

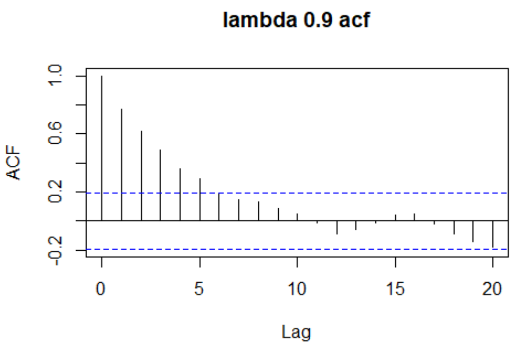




The forecasting of lambda = 0.9 is better.

(c)

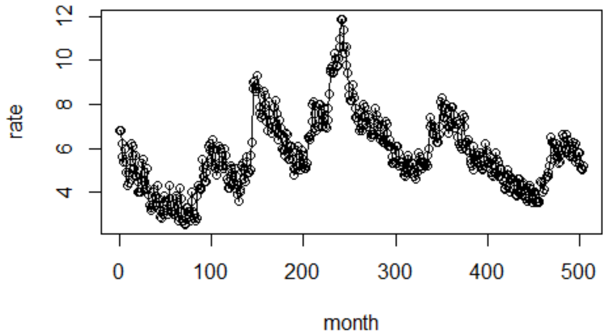


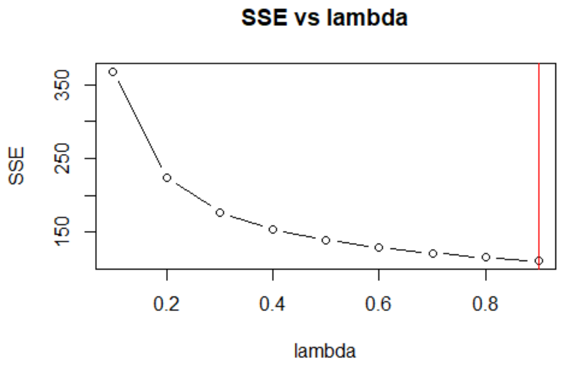


We can see that lambda = 0.9 ‘s acf is better that lambda = 0.1

24.

(a)





We take the best lambda = 0.9, and consider lambda = 0.2

```
> print(measacc(data24[,2],0.2))
        SSE        MAPE         MAD         MSD 
224.2225047   9.2513706   0.5187099   0.4448859 
>   #lambda 0.9 error
> print(measacc(data24[,2],0.9))
        SSE        MAPE         MAD         MSD 
110.6944769   6.4248868   0.3556993   0.2196319 
```

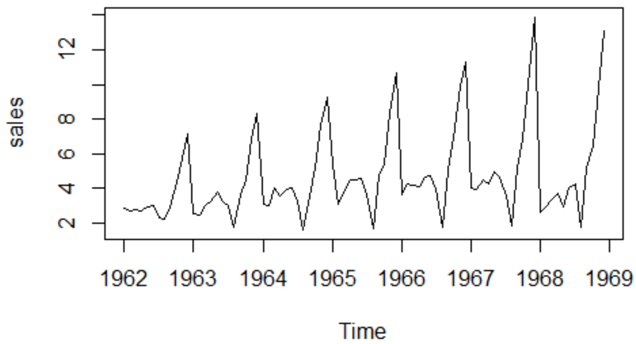
We can see that the errors about lambda = 0.9 are smaller than lambda = 0.2

(b)

I think that using the optimal value can get the better forecasts.

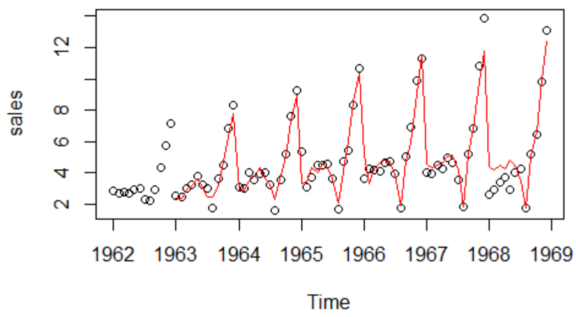
29.

(a)



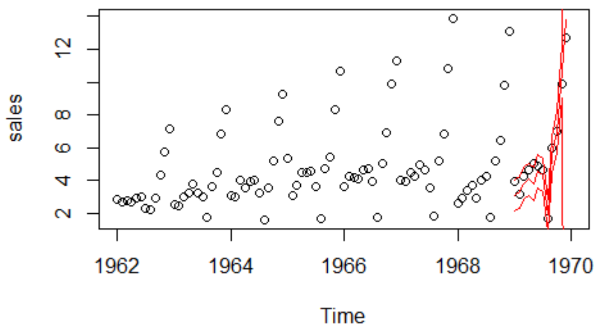
We can see a strong trend about seasonal effects, maybe the sales change by the season changing.

(b)



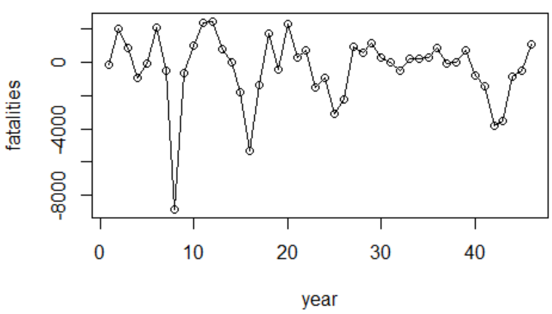
Using Winter's multiplicative method , the plot is similar to (a)

(c)



53.

(a)



I can't see anything about this plot

R code

```
#4.8
data8 =
matrix(c(1:24,315,195,310,316,325,335,318,355,420,410,485,420,460,395,390,450,458,570,520,400,420,5
80,475,560),24,2)
data8 = as.data.frame(data8)
colnames(data8) = c("period","y")
  #plot
plot(data8[,1],data8[,2],xlab = "period",ylab = "y")
lines(data8[,2])
  #smooth function
firstsmooth = function(y,lambda,start = y[1]){
  ytilde = y
  ytilde[1] = lambda*y[1] + (1-lambda)*start
  for(i in 2:length(y)){
    ytilde[i] = lambda*y[i]+(1-lambda)*ytilde[i-1]
  }
  ytilde
}
  #error function
measacc = function(y,lambda){
  out = firstsmooth(y,lambda)
  T = length(y)
  pred = c(y[1],out[1:(T-1)])
  prederr = y-pred
  SSE = sum(prederr^2)
  MAPE = 100*sum(abs(prederr/y))/T
  MAD = sum(abs(prederr))/T
  MSD = sum(prederr^2)/T
  ret1 = c(SSE,MAPE,MAD,MSD)
  names(ret1) = c("SSE","MAPE","MAD","MSD")
  return(ret1)
}
  #best lambda
lambda.vec = seq(0.1,0.9,0.1)
ssecal = function(sc){measacc(data8[1:12,2],sc)[1]}
ssevec = sapply(lambda.vec, ssecal)
opt.lambda = lambda.vec[ssevec==min(ssevec)]
plot(lambda.vec,ssevec,type="b",main = "SSE vs lambda",xlab = "lambda",ylab = "SSE")
abline(v=opt.lambda,col = "red")
mtext(text = paste("SSE min = ",round(min(ssevec),2),"\n lambda = ",opt.lambda))
  #second order forecast 12 step ahead
```

```
blambda = 0.6
smooth1 = firstsmooth(data8[1:12,2],blambda)
smooth2 = firstsmooth(smooth1,blambda)
hat = 2*smooth1- smooth2
tau = 1:12
t = length(smooth1)
yforecast = (2+tau*(blambda/(1- blambda)))*smooth1[t]-(1+tau*(blambda/(1- blambda)))*smooth2[t]
ctau =sqrt(1+(blambda/((2- blambda)^3))*(10-14*blambda+5*(blambda^2)+2*tau*blambda*(4-
3*blambda)+2*(tau^2)*(blambda^2)))
alpha = 0.05
est = sqrt(var(data8[2:12,2]-hat[1:11]))
cl = qnorm(1-alpha/2)*(ctau/ctau[1])*est
plot(data8[1:12,2],type = "p",xlim = c(0,24),ylim = c(0,1.5*max(data8[,2])),xlab = "period",ylab = "y")
points(13:24,data8[13:24,2])

lines(13:24,yforecast,col = "red")
points(13:24,yforecast,col = "red")
lines(13:24,yforecast+cl,col = "red")
lines(13:24,yforecast-cl,col = "red")

thiserror = rep(0,12)
thiserror = yforecast[1:12] - data8[13:24,2]
print(thiserror)


#4.9
data9 =
matrix(c(1:24,315,195,310,316,325,335,318,355,420,410,485,420,460,395,390,450,458,570,520,400,420,5
80,475,560),24,2)
diff9 = diff(data9)
diff9[,1] = 1:23
diff9 = as.data.frame(diff9)
colnames(diff9) = c("period","time")
   #plot
plot(diff9[,1],diff9[,2],xlab = "period with one difference",ylab = "y")
   #best lambda
lambda.vec = seq(0.1,0.9,0.1)
ssecal = function(sc){measacc(diff9[1:11,2],sc)[1]}
ssevec = sapply(lambda.vec, ssecal)
opt.lambda = lambda.vec[ssevec==min(ssevec)]
plot(lambda.vec,ssevec,type="b",main = "SSE vs lambda",xlab = "lambda",ylab = "SSE")
abline(v=opt.lambda,col = "red")
```

```
mtext(text = paste("SSE min = ",round(min(ssevec),2),"\n lambda = ",opt.lambda))
    #simple exponential
library(forecast)
simpfore = ses(diff9[1:11,2],h = 12,initial = "simple")


plot(diff9[1:11,2],type = "p",xlim = c(0,24),ylim = c((-1.5)*max(diff9[,2]),1.5*max(diff9[,2])),xlab =
"period",ylab = "y first difference")
points(12:23,diff9[12:23,2])
lines(12:23,simpfore$mean,col = "red")
points(12:23,simpfore$mean,col = "red")
lines(12:23,simpfore$upper[,2],col = "red")
lines(12:23,simpfore$lower[,2],col = "red")


data9$fore=0
data9[1:12,3] = data9[1:12,2]
for(i in 1:12){
    data9[12+i,3] = data9[11+i,3] + simpfore$mean[i]
}

plot(data9[1:11,2],type = "p",xlim = c(0,24),ylim = c(0,1.5*max(data9[,2])),xlab = "period",ylab = "y")
points(13:24,data9[13:24,2])
lines(13:24,data9[13:24,3],col = "red")
points(13:24,data9[13:24,3],col = "red")
lines(13:24,data9[13:24,3]+simpfore$upper[,2]+simpfore$mean,col = "red")
lines(13:24,data9[13:24,3]+simpfore$lower[,2]+simpfore$mean,col = "red")

thiserror = rep(0,12)
thiserror = data9[13:24,3] - data9[13:24,2]
print(thiserror)


#4.15
data15 = read.csv("b4.csv")
data15[,2] = as.character(data15[,2])
data15[,2] = gsub(",","",data15[,2])
data15[,2] = as.numeric(data15[,2])
    #plot
plot(data15[,1],data15[,2],xlab = "year",ylab = "production")
    #get best lambda
lambda.vec = seq(0.1,0.9,0.1)
```

```
ssecal = function(sc){measacc(data15[1:38,2],sc)[1]}
ssevec = sapply(lambda.vec, ssecal)
opt.lambda = lambda.vec[ssevec==min(ssevec)]
plot(lambda.vec,ssevec,type="b",main = "SSE vs lambda",xlab = "lambda",ylab = "SSE")
abline(v=opt.lambda,col = "red")
mtext(text = paste("SSE min = ",round(min(ssevec),2),"\n lambda = ",opt.lambda))
  #lambda 0.1 error
print(measacc(data15[,2],0.1))
  #lambda 0.9 error
print(measacc(data15[,2],0.9))

  #one step ahead lambda = 0.1
blambda = 0.1
t = 39
tau = 10
alpha = 0.05
yforecast = rep(0,tau)
cl = rep(0,tau)
smooth1 = rep(0,t+tau)
smooth2 = rep(0,t+tau)
for(i in 1:tau){
  smooth1[1:(t+i-1)] = firstsmooth(data15[1:(t+i-1),2],blambda)
  smooth2[1:(t+i-1)] = firstsmooth(smooth1[1:(t+i-1)],blambda)
  yforecast[i] = (2+(blambda/(1- blambda)))*smooth1[t+i-1]-(1+(blambda/(1- blambda)))*smooth2[t+i-1]
  hat = 2*smooth1[1:(t+i-1)]-smooth2[1:(t+i-1)]
  est = sqrt(var(data15[2:(t+i-1),2]-hat[1:(t+i-2)]))
  cl[i] = qnorm(1-alpha/2)*est
}
plot(data15[1:t,2],type = "p",xlim = c(0,t+tau),ylim = c(0,1.5*max(data15[,2])),xlab = "period",ylab =
"production",main = "lambda = 0.1")
points((t+1):(t+tau),data15[(t+1):(t+tau),2])
points((t+1):(t+tau),yforecast,col = "red")
lines((t+1):(t+tau),yforecast,col = "red")
lines((t+1):(t+tau),yforecast+cl,col = "red")
lines((t+1):(t+tau),yforecast-cl,col = "red")

thiserror = rep(0,10)
thiserror = yforecast[1:10] - data15[39:48,2]
print(thiserror)

autocorr = acf(c(data15[1:38,2],yforecast[1:10]),plot = FALSE)
autocorr
```

```
autocorrplot = acf(c(data15[1:38,2],yforecast[1:10]),main = "lambda 0.1 acf")

  #lambda = 0.9
blambda = 0.9
t = 39
tau = 10
alpha = 0.05
yforecast = rep(0,tau)
cl = rep(0,tau)
smooth1 = rep(0,t+tau)
smooth2 = rep(0,t+tau)
for(i in 1:tau){
  smooth1[1:(t+i-1)] = firstsmooth(data15[1:(t+i-1),2],blambda)
  smooth2[1:(t+i-1)] = firstsmooth(smooth1[1:(t+i-1)],blambda)
  yforecast[i] = (2+(blambda/(1- blambda)))*smooth1[t+i-1]-(1+(blambda/(1- blambda)))*smooth2[t+i-1]
  hat = 2*smooth1[1:(t+i-1)]-smooth2[1:(t+i-1)]
  est = sqrt(var(data15[2:(t+i-1),2]-hat[1:(t+i-2)]))
  cl[i] = qnorm(1-alpha/2)*est
}
plot(data15[1:t,2],type = "p",xlim = c(0,t+tau),ylim = c(0,1.5*max(data15[,2])),xlab = "period",ylab =
"production",main = "lambda = 0.9")
points((t+1):(t+tau),data15[(t+1):(t+tau),2])
points((t+1):(t+tau),yforecast,col = "red")
lines((t+1):(t+tau),yforecast,col = "red")
lines((t+1):(t+tau),yforecast+cl,col = "red")
lines((t+1):(t+tau),yforecast-cl,col = "red")

thiserror = rep(0,10)
thiserror = yforecast[1:10] - data15[39:48,2]
print(thiserror)

autocorr = acf(c(data15[1:38,2],yforecast[1:10]),plot = FALSE)
autocorr
autocorrplot = acf(c(data15[1:38,2],yforecast[1:10]),main = "lambda 0.9 acf")

#4.24
data24 = read.csv("b8.csv")
data24 = data24[,-c(3:12)]
data24[,1] = 1:504
diff24 = matrix(0,503,2)
diff24[,1] = 1:503
diff24[,2] = diff(data24[,2])
```

```
diff24 = as.data.frame(diff24)
colnames(diff24) = c("period","rate")
data24 = as.data.frame(data24)
colnames(data24) = c("period","rate")
  #plot
plot(data24[,1],data24[,2],xlab = "month",ylab = "rate")
lines(data24[,2])
  #get best lambda
lambda.vec = seq(0.1,0.9,0.1)
ssecal = function(sc){measacc(data24[,2],sc)[1]}
ssevec = sapply(lambda.vec, ssecal)
opt.lambda = lambda.vec[ssevec==min(ssevec)]
plot(lambda.vec,ssevec,type="b",main = "SSE vs lambda",xlab = "lambda",ylab = "SSE")
abline(v=opt.lambda,col = "red")
mtext(text = paste("SSE min = ",round(min(ssevec),2),"\n lambda = ",opt.lambda))
  #lambda 0.2 error
print(measacc(data24[,2],0.2))
  #lambda 0.9 error
print(measacc(data24[,2],0.9))

#4.29
data29 = read.csv("b11.csv")
  #convert to time series data
ts29 = ts(data29[1:84,2],start = c(1962,1),frequency = 12)
  #plot
plot(ts29, ylab = "sales")
  #holt winters model
hw = HoltWinters(ts29,seasonal = "multiplicative")
plot(ts29,type = "p",ylab = "sales")
lines(hw$fitted[,1],col = "red")
  #forecast
ts29.2 = ts(data29[85:96,2],start = c(1969,1),frequency = 12)
yforecast = predict(hw,n.ahead = 12,prediction.interval = TRUE)
plot(ts29,type = "p",ylab = "sales",xlim = c(1962,1970))
points(ts29.2)
lines(yforecast[,1],col = "red")
lines(yforecast[,2],col = "red")
lines(yforecast[,3],col = "red")

thiserror = rep(0,12)
thiserror = yforecast[1:12] - data29[85:96,2]
print(thiserror)
```

```
#4.53
data53 = read.csv("b25.csv")
for(i in 2:7){
   data53[,i] = as.character(data53[,i])
   data53[,i] = gsub(",","",data53[,i])
   data53[,i] = as.numeric(data53[,i])
}

diff53 = matrix(0,46,7)
diff53[,1] = 1:46
for(i in 2:7){
   diff53[,i] = diff(data53[,i])
}
   #plot
plot(diff53[,1],diff53[,2],xlab = "year",ylab = "fatalities")
lines(diff53[,2])
```