DATA624 Homework 1

Bin Lin

2018-3-25

Exercise 2.8 1. For each of the following series (from the fma package), make a graph of the data. If transforming seems appropriate, do so and describe the effect.

a. Monthly total of people on unemployed benefits in Australia (January 1956-July 1992).

b. Monthly total of accidental deaths in the United States (January 1973-December 1978).

c. Quarterly production of bricks (in millions of units) at Portland, Australia (March 1956-September 1994).

Hints: data(package="fma") will give a list of the available data. To plot a transformed data set, use plot(BoxCox(x,0.5)) where x is the name of the data set and 0.5 is the Box-Cox parameter.

Approaches: After downloading the datasets, I printed out the Time Plot, Seasonal Plot, and Seasonal Subseries Plot before I make a decision that if data transforming is appropriate. The Time Plot is the graph that plots the observations against the time of observations, with consecutive observations joined by straight lines. Seasonal Plot is very similar to Time Plot, except its x-axis is corresponding to seasons. The Seasonal Subseries Plots a particular type of Seasonal Plots where data for each season are collected together in separate mini time plots.

a. Interpretations: The Time Plot a significant increase in terms of people who collect unemployment benefits in Australia, in the year of 1975, 1982-1983, and after 1990. In the meantime, there is significant decline between 1982 to 1990. The changes does not seem to be periodical or seasonal. The evidences are showned in Seasonal Plot and Seasonal Subseries Plot. The average number of people collecting unemployment benefit which is indicated by the hotizonal line in Seasonal Subseries Plot shows throughout the year the observations decreases slightly with a slight increase in the month of december. Since the changes of the number of observations are huge. The differences tends to be exponential, I think it is appropriate to adopt Box-Cox transformations. The shape of the graph does not changed much, however, the scale of the y-interval falls within 0 to 300 right now.

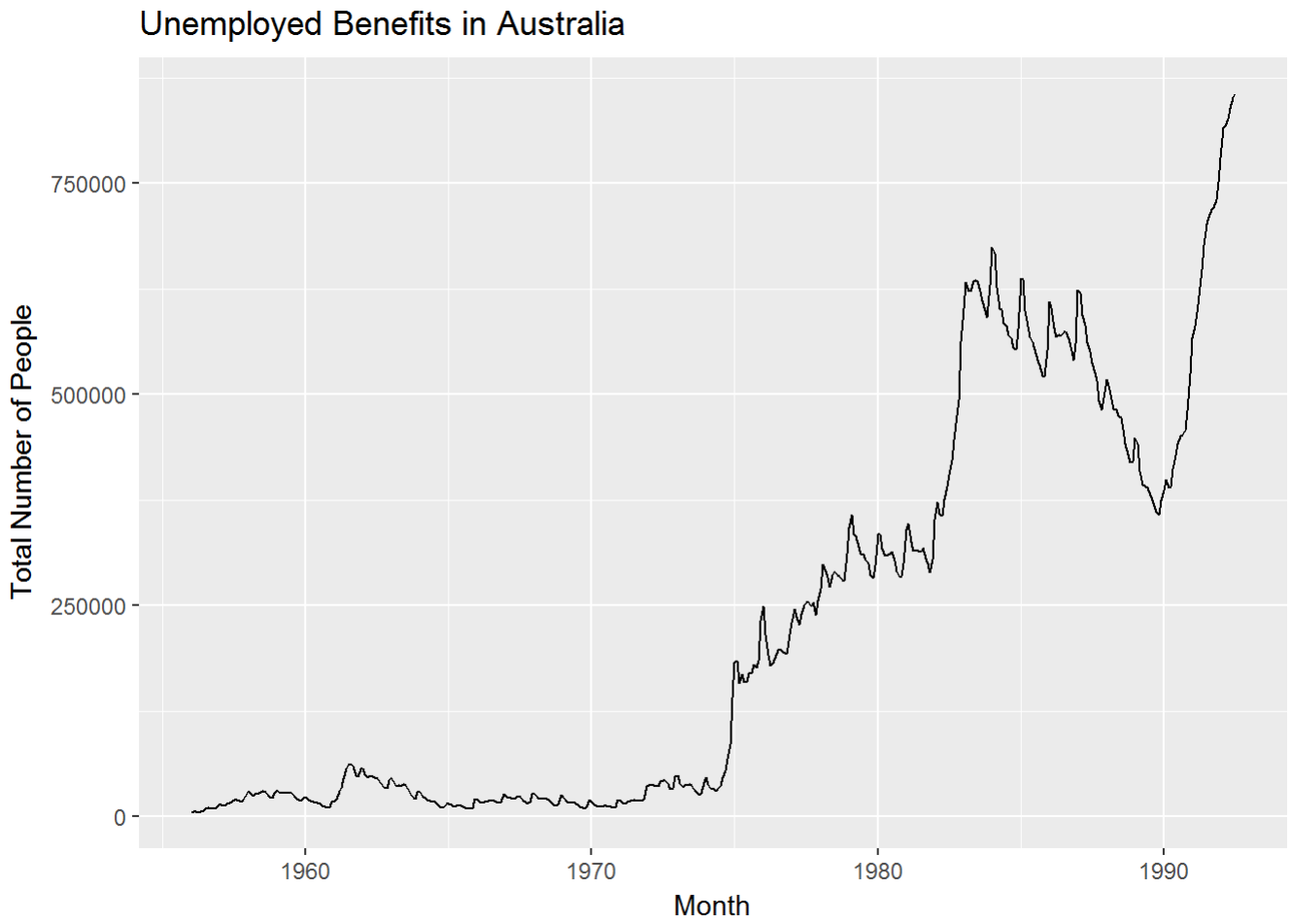
```
#install.packages("fma")
#install.packages("xts")
#install.packages("ggplot2")
#install.packages("forecast")
library("forecast")
library("xts")
```

```
## Loading required package: zoo
```

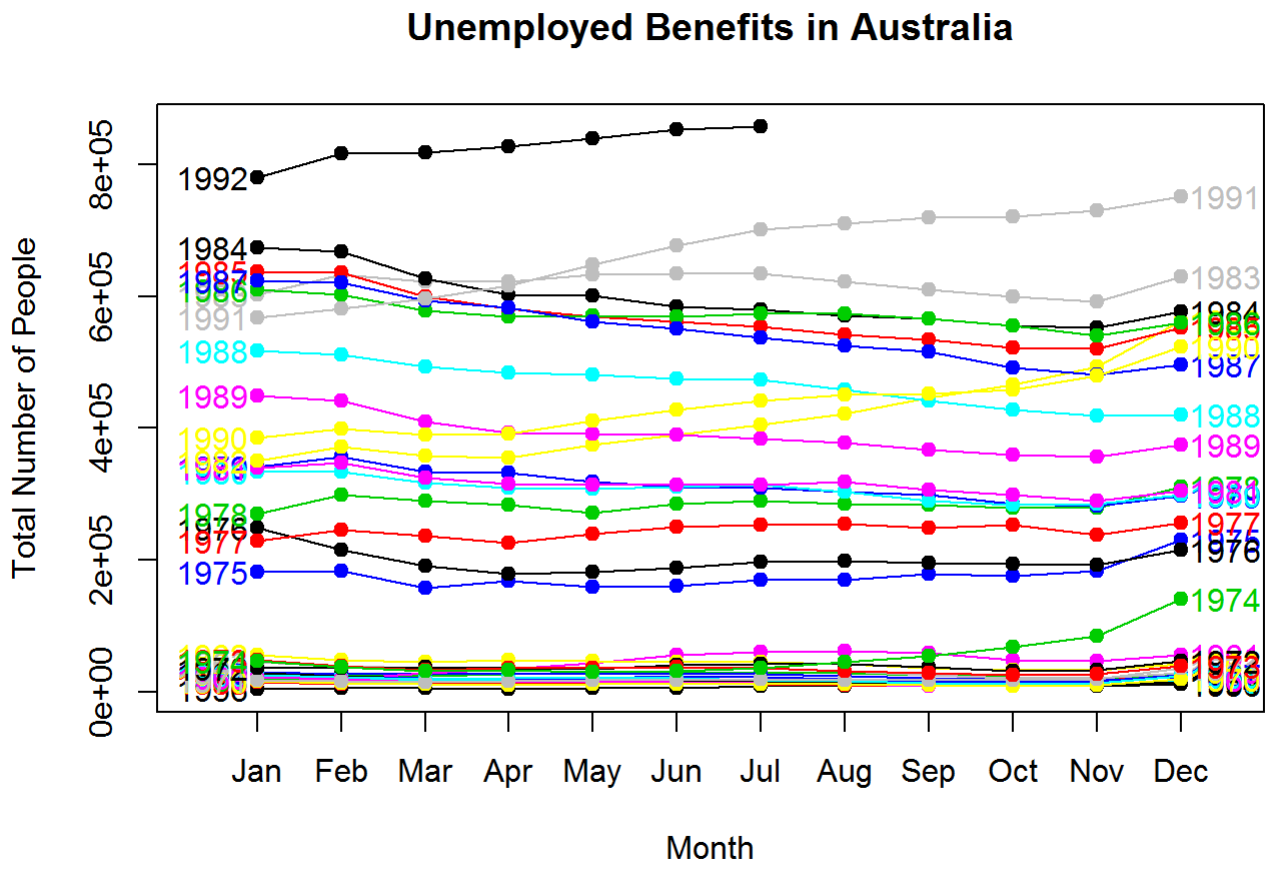
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

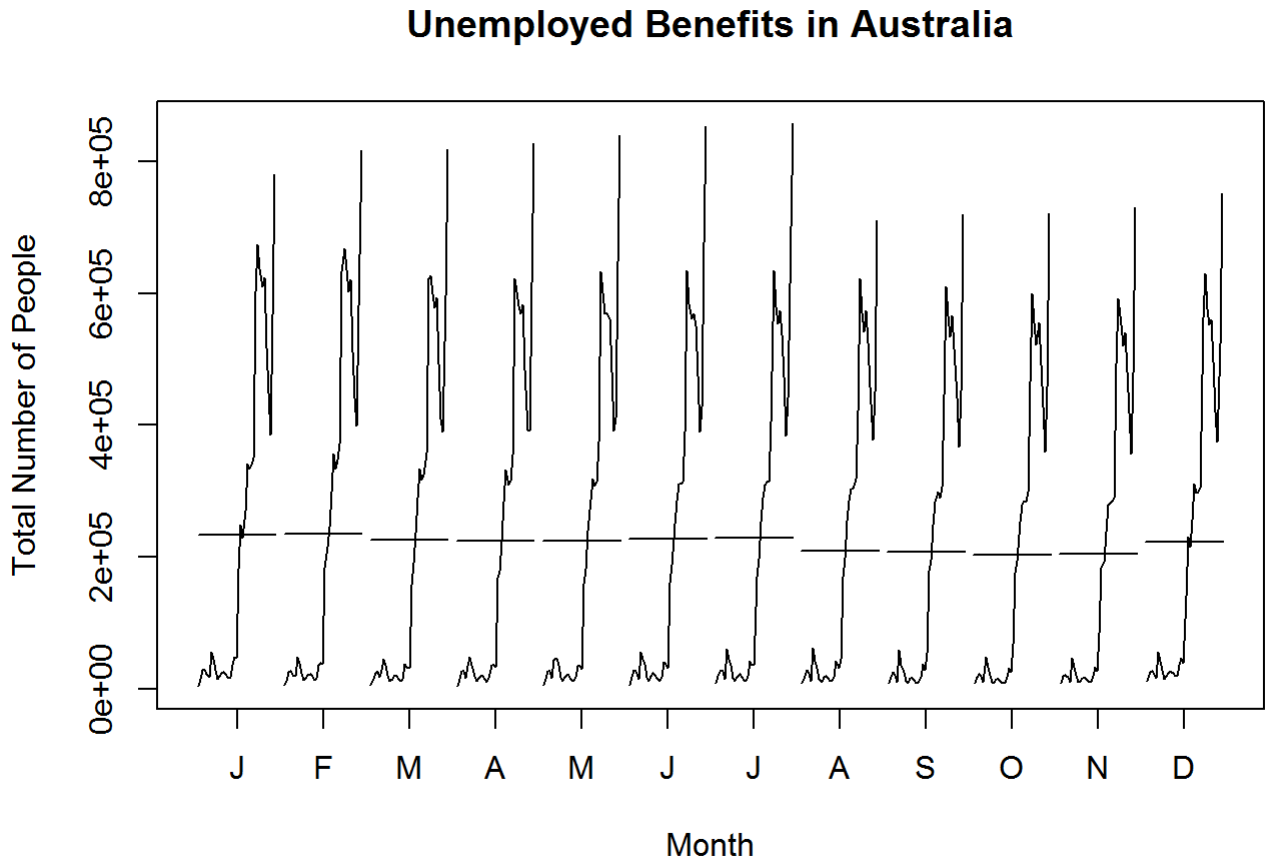
```
library("fma")
library("ggplot2")
#data(package="fma")
#data("dole")
#head(dole)

autoplot(dole, main="Unemployed Benefits in Australia", ylab="Total Number of People", xlab="Mon
th")
```



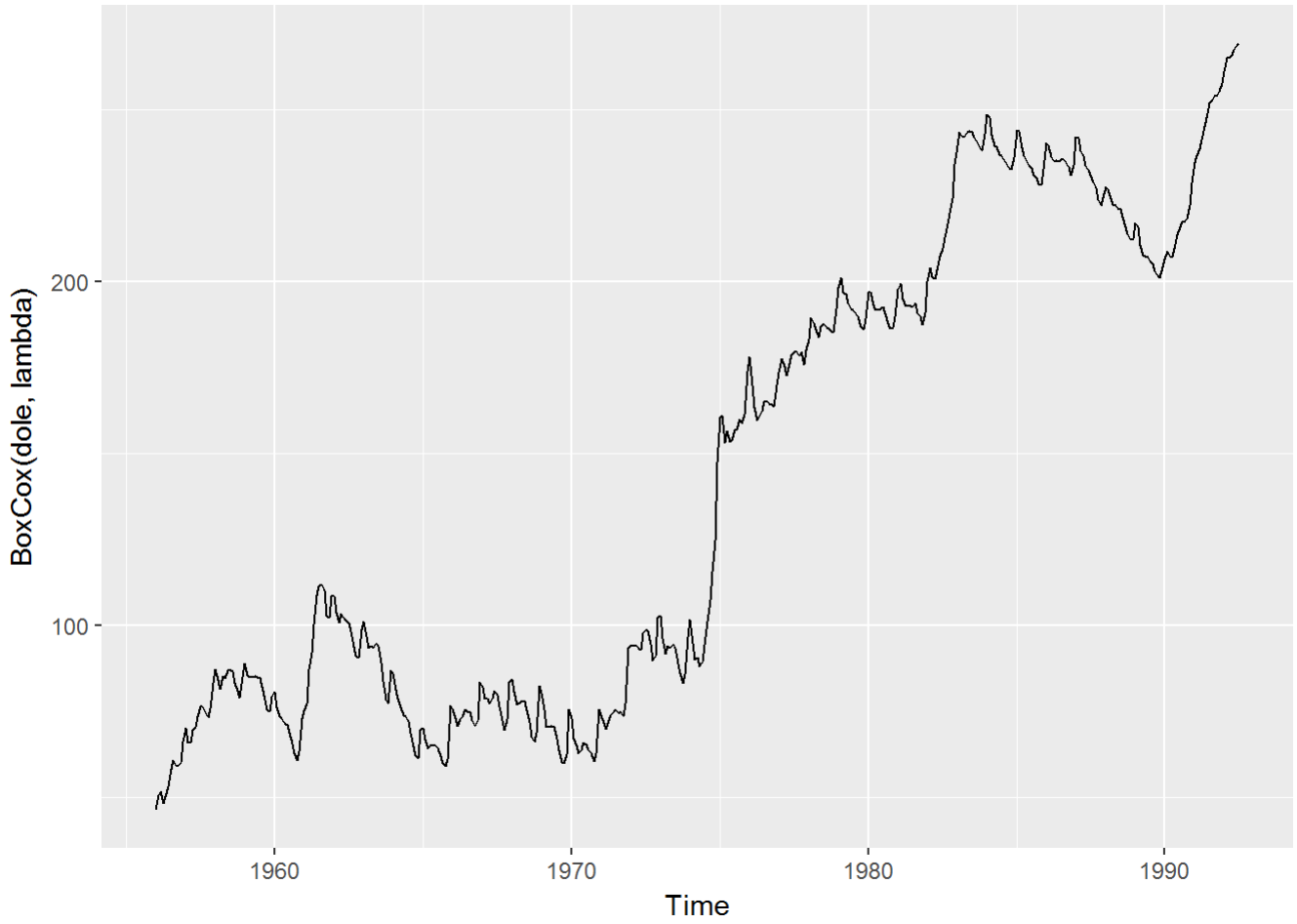

```
seasonplot(dole, main="Unemployed Benefits in Australia", ylab="Total Number of People", xlab="M
onth",year.labels=TRUE, year.labels.left=TRUE, col=1:20, pch=19)
```




```
monthplot(dole, main="Unemployed Benefits in Australia", ylab="Total Number of People", xlab="Mo
nth")
```

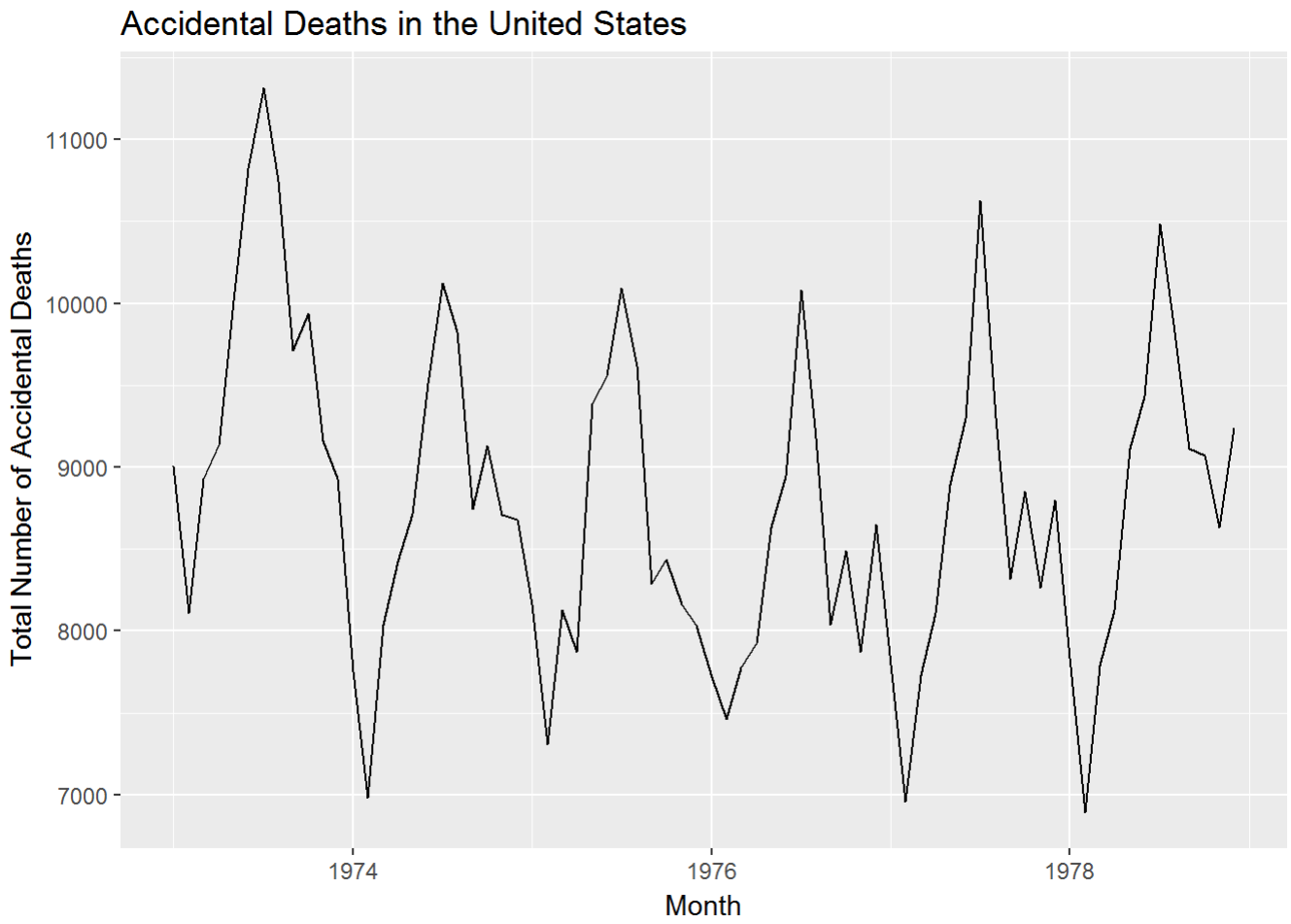



```
lambda <- BoxCox.lambda(dole)
autoplot(BoxCox(dole, lambda))
```

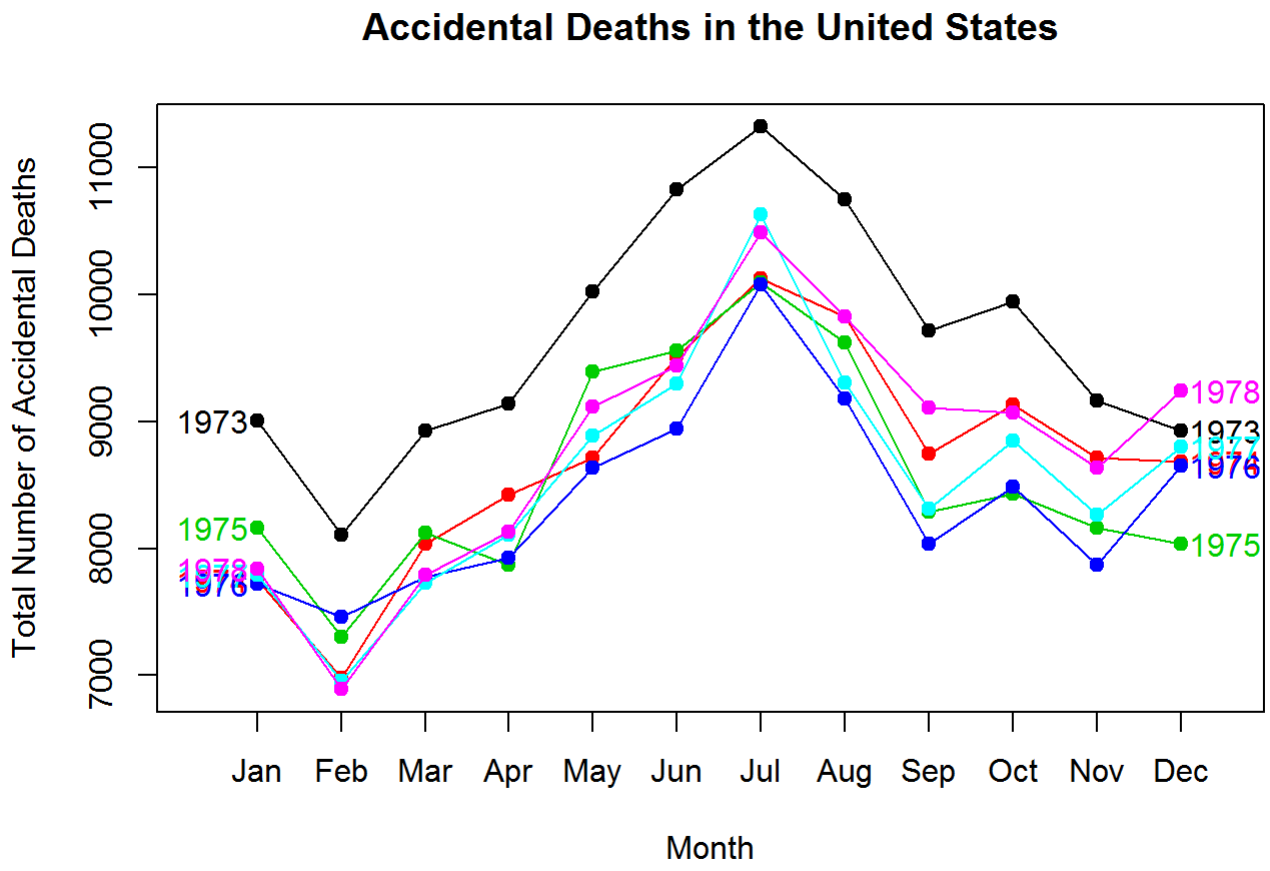


b. Interpretation: The Time Plot shows the observations oscillate over the years. There are peaks and troughs periodically. The Seasonal Plot shows for each year, the peak usually happen during the summer, especially the month of July, we will see the highest number of accidental death during that year. From the Seasonal Subseries Plot, we are able to tell that between 1973 and 1978, the accidetanl deaths goes down first then goes up again.

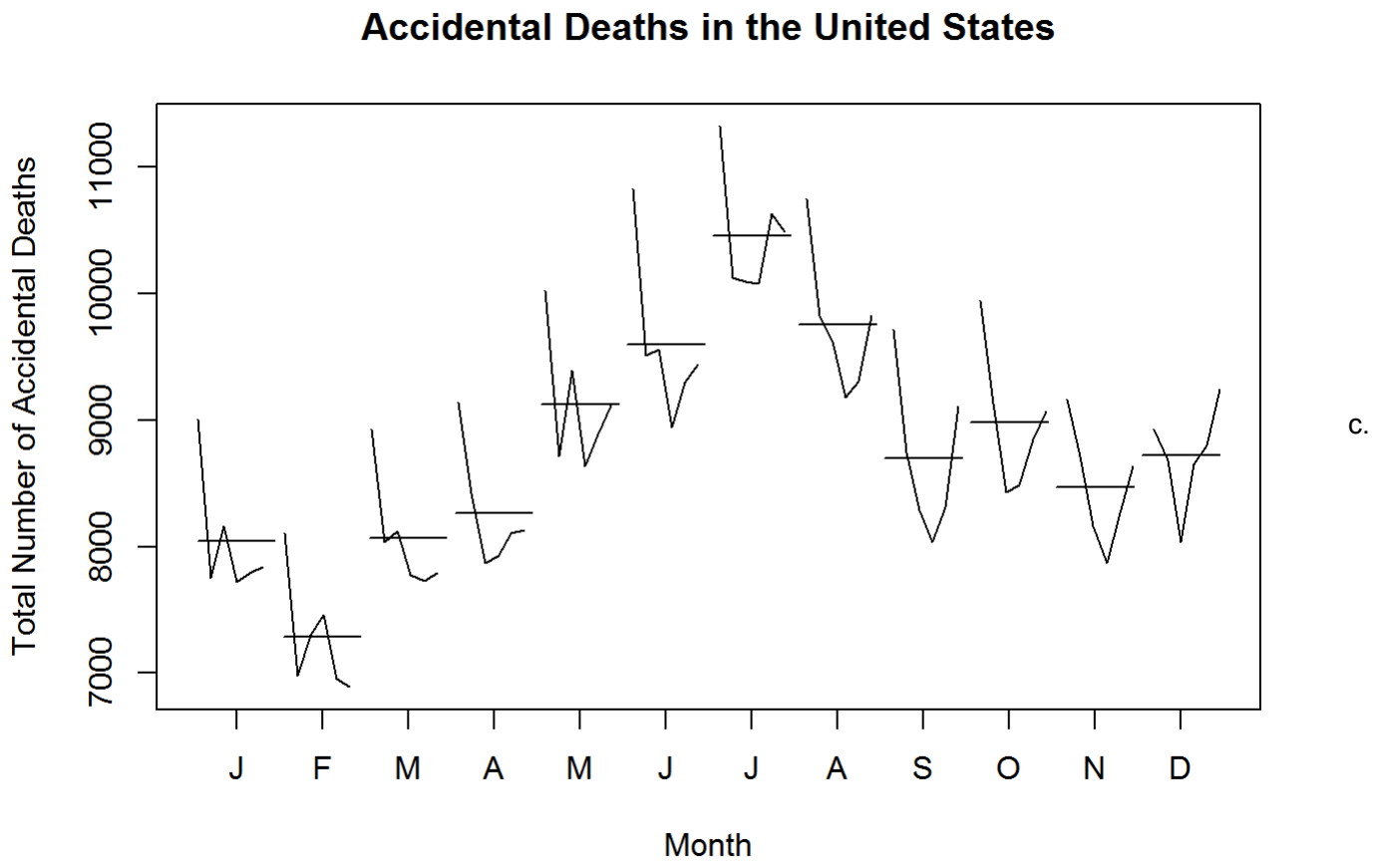
```
autoplot(usdeaths, main="Accidental Deaths in the United States", ylab="Total Number of Accident
al Deaths", xlab="Month")
```




```
seasonplot(usdeaths, main="Accidental Deaths in the United States", ylab="Total Number of Accide
ntal Deaths", xlab="Month",year.labels=TRUE, year.labels.left=TRUE, col=1:20, pch=19)
```



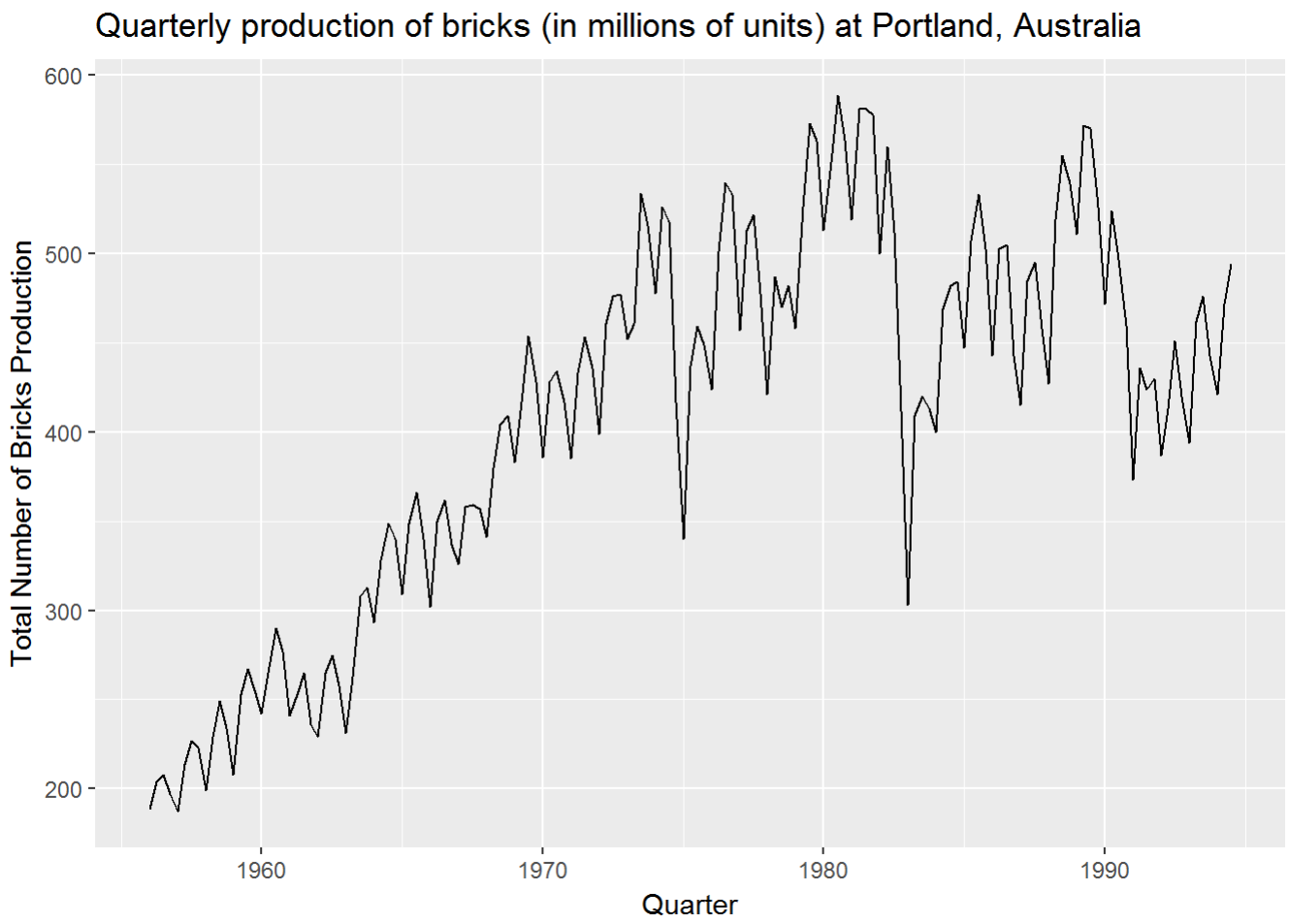

```
monthplot(usdeaths, main="Accidental Deaths in the United States", ylab="Total Number of Acciden
tal Deaths", xlab="Month")
```



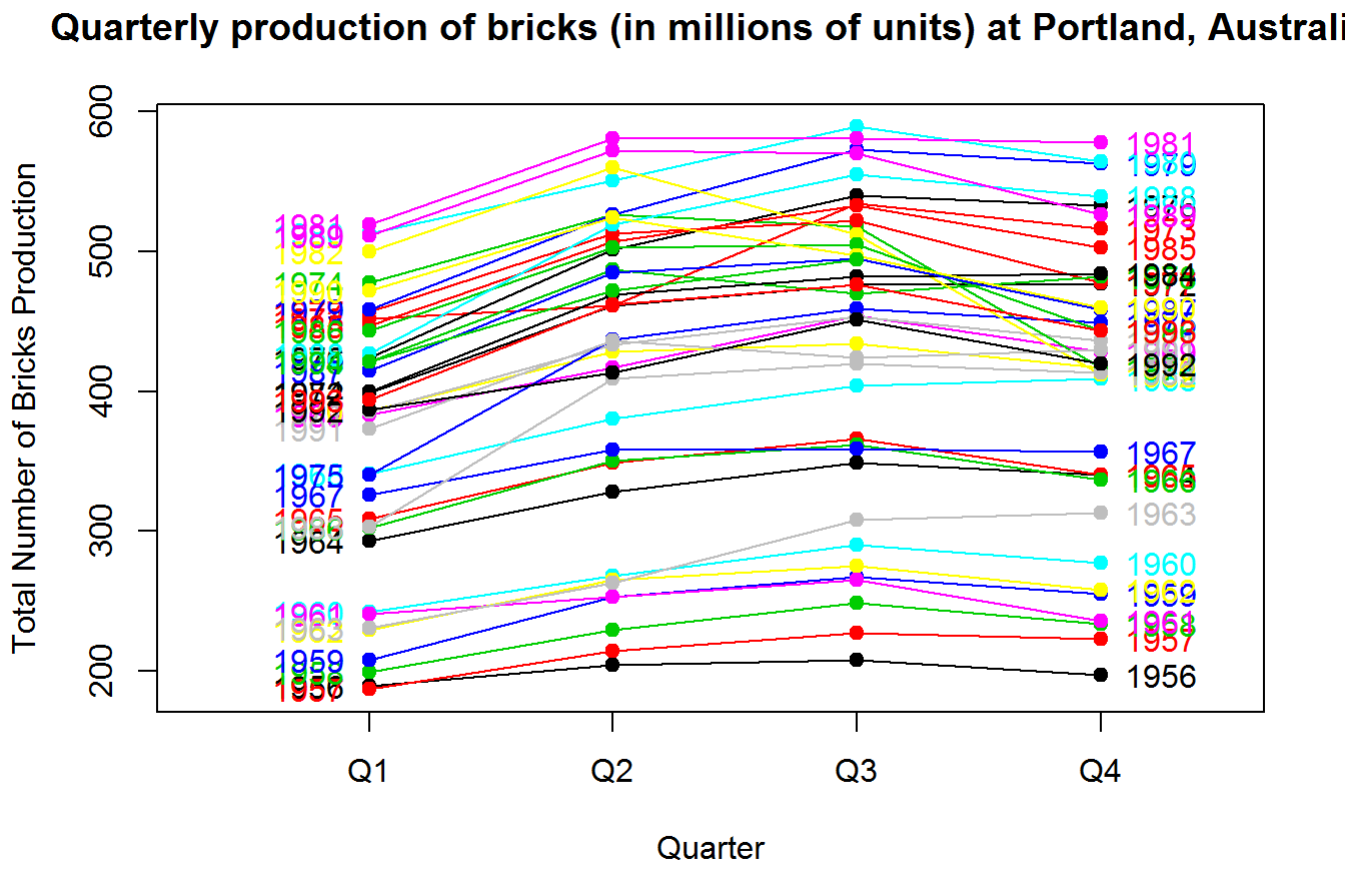
c.

Interpretation: The Time Plot shows steady increase of quarterly production of bricks before the year of 1975, then Portland experienced sudden decrease and increase of production few times after 1975. The Seaon Plot shows that the first quarter tends to have weaker production, while Seaonal Subseries Plot also demonstrate that.

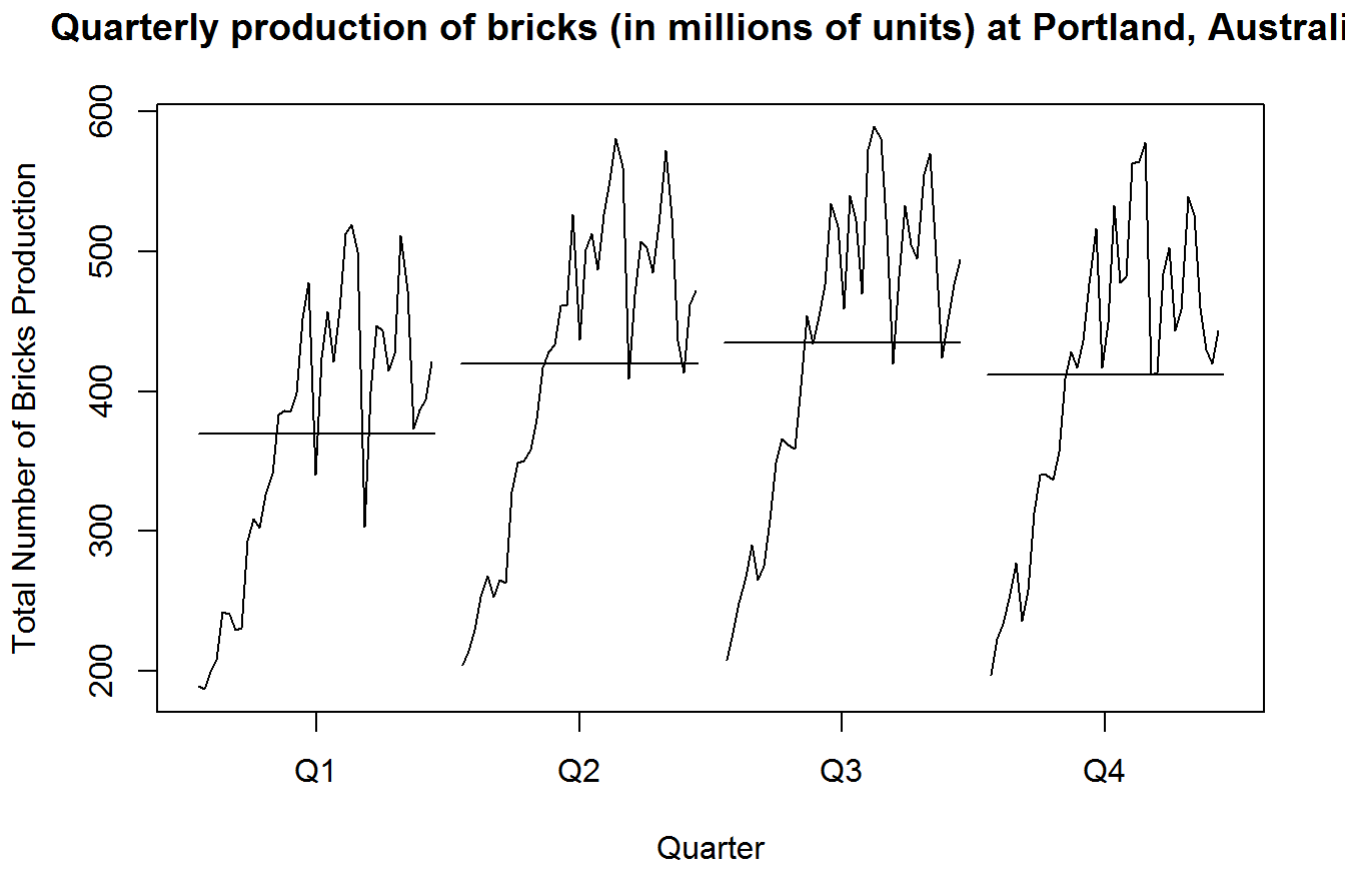
```
autoplot(bricksq, main="Quarterly production of bricks (in millions of units) at Portland, Austr
alia", ylab="Total Number of Bricks Production", xlab="Quarter")
```




```
seasonplot(bricksq, main="Quarterly production of bricks (in millions of units) at Portland, Aus
tralia", ylab="Total Number of Bricks Production", xlab="Quarter",year.labels=TRUE, year.labels.
left=TRUE, col=1:20, pch=19)
```




```
monthplot(bricksq, main="Quarterly production of bricks (in millions of units) at Portland, Aust
ralia", ylab="Total Number of Bricks Production", xlab="Quarter")
```



3. Consider the daily closing IBM stock prices (data set ibmclose).

a. Produce some plots of the data in order to become familiar with it.

The Time Plot shows that in the first 100 days, the stock price increase steadily. The following around 150 days, it has been flat. The latter 150 days, the stock price plummeted, and the daily price change has been dramatic.

```
autoplot(ibmclose, main="Daily Closing IBM Stock Prices", ylab="Stock Price", xlab="Day")
```

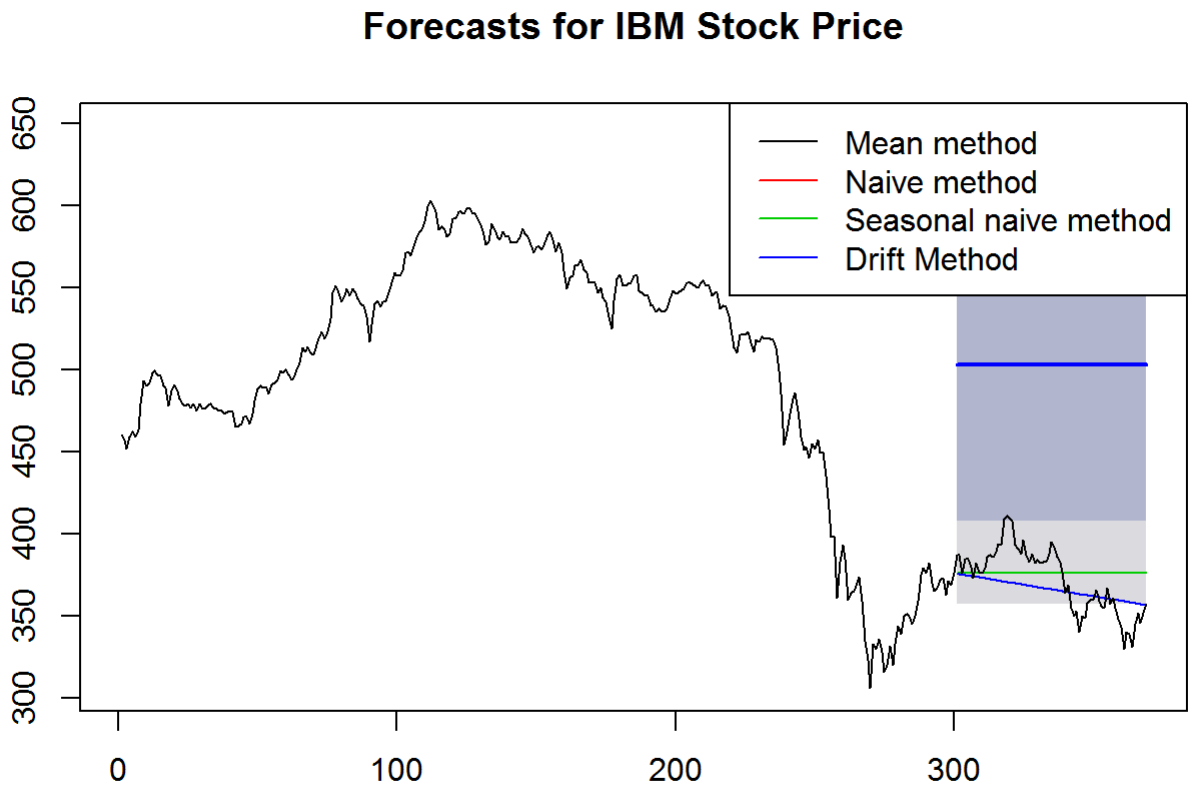
b. Split the data into a training set of 300 observations and a test set of 69 observations.

```
#training <- ibmclose[1: 300]
#test <- ibmclose[301:369]

training <- window(ibmclose, start = 1, end = 300)
test <- window(ibmclose, start = 301, end = 369)
```

c. Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

Approaches: There are several firecasting methods. Average method forecasts of all future values are equal to the mean of the historical data. Naive method forecasts all future value are simply the value of the last observation. Seasonal Naive Method forecasts all future values are be equal to the last observed value from the same season of the year. Drift Method forecasts the future values will increase or decrease over time, where the amount of change over time (called the drift) is set to be the average change seen in the historical data.

```
ibmfit1 <- meanf(training, h=69)
ibmfit2 <- naive(training, h=69)
ibmfit3 <- snaive(training, h=69)
ibmfit4 <- rwf(training, h = 69, drift=TRUE)

plot(ibmfit1, main="Forecasts for IBM Stock Price")
lines(ibmfit2$mean, col=2)
lines(ibmfit3$mean, col=3)
lines(ibmfit4$mean, col=4)
legend("topright", lty = 1,col = 1:4, legend=c("Mean method","Naive method","Seasonal naive meth
od", "Drift Method"))
lines(ibmclose)
```



Interpretation: From the following table, we are able to tell that out of all the methods, Drift Method is the best method in this case to predict the IBM stock price, because IBM has the lowest Mean Absolute Error(MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error(MAPE), Mean Absolute Scaled Error(MASE). The mean method is the worst, whiel both Naive Method and Seasonal Naive Method produce the same result.

```
ibm_accuracy <- data.frame(rbind(accuracy(ibmfit1, test)[2, c(2, 3, 5, 6)],
                         accuracy(ibmfit2, test)[2, c(2, 3, 5, 6)],
                         accuracy(ibmfit3, test)[2, c(2, 3, 5, 6)],
                         accuracy(ibmfit4, test)[2, c(2, 3, 5, 6)]
                         ))

row.names(ibm_accuracy) <- c("Mean method","Naive method","Seasonal naive method", "Drift Metho
d")

ibm_accuracy
```

```
##                             RMSE       MAE      MAPE      MASE
## Mean method            132.12557 130.61797 35.478819 25.626492
## Naive method            20.24810  17.02899  4.668186  3.340989
## Seasonal naive method   20.24810  17.02899  4.668186  3.340989
## Drift Method            17.06696  13.97475  3.707888  2.741765
```