Assignment1_Seshadri

Sri Seshadri

4/7/2018

Section 2.8, Question 2.1 page 59 - Effect of transformation on time series data.

a) Monthly total of peole on unemployment benefits in Australia (Jan 1956 - July 1992)

Figure 1 shows the time series of number of people in unemplyment benefits in Australia by month. The time series is affected by

* Population growth over time
* External factors like:
    * state of the economy
    * the benefit provided by the government.

It would be useful to normalize the data by population, to get the percent unemployed of the total population. Then if need be a transformation on the normalized data can be made.

```
data(package = 'fma',dole)
plot(dole, main = "Total people on unemployment benefits in Australia", ylab = "Number of people")
```

b) Monthly total accidental deaths in the United States

The top plot in figure 2 shows the total accidental deaths by month in the US. There is seasonality in the data, where the total accidents peaking at July. The variation in the seasonality may be mitigated by normalizing the totals by dividing by the number of days in the month.

The middle plot in figure 2 shows the normalized total by days in the months; i.e. Monthly average accidental deaths per day. We see that there is some smoothing of the raw data. While there is not much variation in seasonality, there is interest in further making the size of the seasonal variation equal across seasons.

The bottom chart of figure 2 shows a box-cox transformation of the average accidental deaths. Its seen that there isn't much affect as expected.

```
data("usdeaths")
par(mfrow = c(3,1))
plot(usdeaths, type = "b",main = " Monthly accidental deaths", ylab = "")
grid(nx = NULL, ny = 72/12, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
plot(usdeaths/monthdays(usdeaths), type = "b",main = "Avg accidental deaths per day", ylab ="")
grid(nx = NULL, ny = 72/12, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
usdeaths.lambda <- BoxCox.lambda(usdeaths/monthdays(usdeaths))
plot(BoxCox(usdeaths/monthdays(usdeaths),usdeaths.lambda), type = "b", main = "Avg accidental deaths pe
grid(nx = NULL, ny = 72/12, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
```

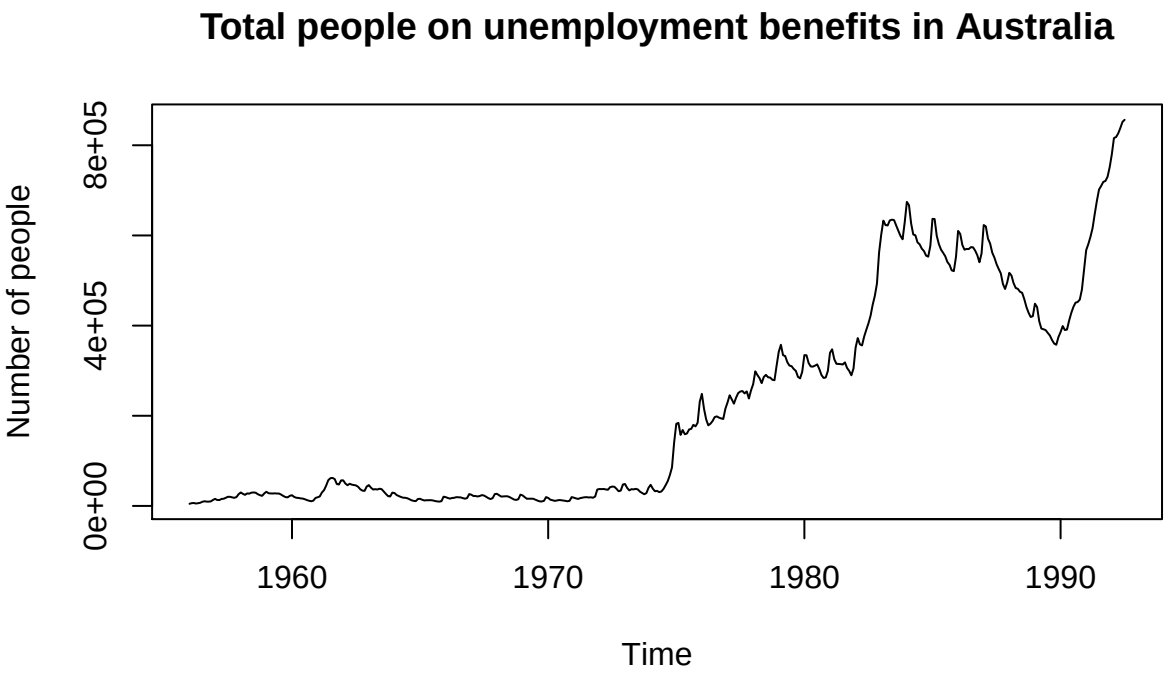


Figure 1: Monthly number of people on unemployment benefits in Australia

c) Quarterly production of bricks (in millions) at Portland, Australia

The top chart in Figure 3 shows the time series plot of quarterly brick production at Portland Australia. The time series exhibits an increasing trend and seasonality. The variation increases with time / levels. Box-Cox transformation is appropriate for this case. The ideal lambda for the data was 0.255. The bottom chart of figure 3 shows the transformed data. The variation is better across time, of course the huge downward spikes is not fully mitigated.

```
data("bricksq")
par(mfrow = c(2,1))
plot(bricksq,main = "Quarterly production of bricks (in millions) at Portland, Australia",cex.main = 1,y
grid(ny = NULL, nx = 156/4, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
bricksq.lambda <- BoxCox.lambda(bricksq)
transformedBricksq <- BoxCox(bricksq,bricksq.lambda)
plot(transformedBricksq, main = "Transformed Quarterly production of bricks at Portland, Australia",cex
grid(ny = NULL, nx = 156/4, col = "lightgray", lty = "dotted",
     lwd = par("lwd"), equilogs = TRUE)
```

Question 2.2 Page 60

Time series modeling of Dow Jones index

In this section the Dow jones industrial average is modeled as a time series. The drift method is used to fit the time series and forecasted for the next 10 periods. Figure 4 shows the time series plot with forecast from drift,mean and naive mean methods. Table 1 shows the forecasts from drift method and the blue line in fig 4 plots the forecasts. The drift method forecast is nothing but the extension of the line that joins the first and the last point. The slope of the line is the difference between the last and the first point divided by the

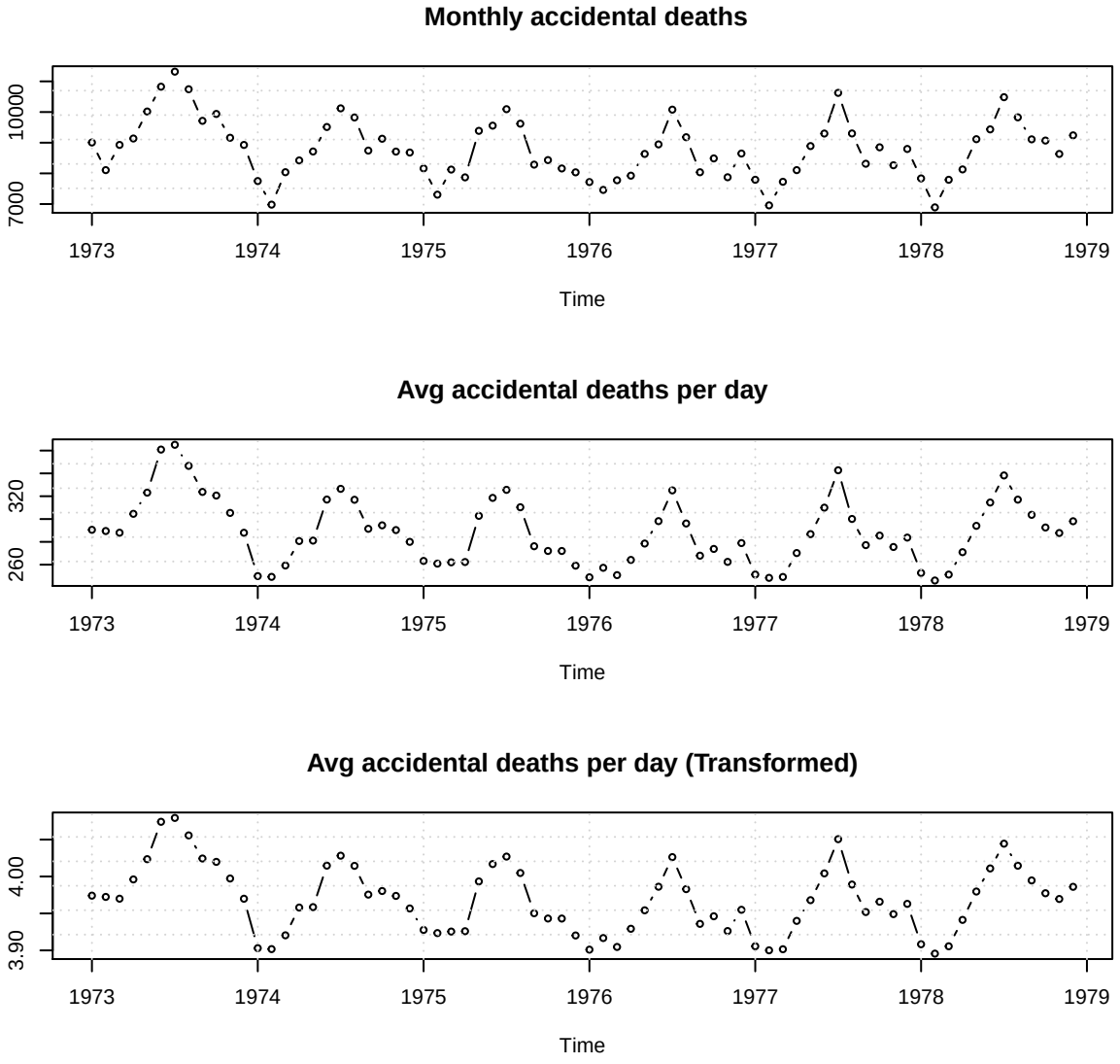


Figure 2: Top: Total accidental deaths by month in the US, Middle: Monthly Average Accidental deaths per day, Bottom: Transformed monthly average accidental deaths per day

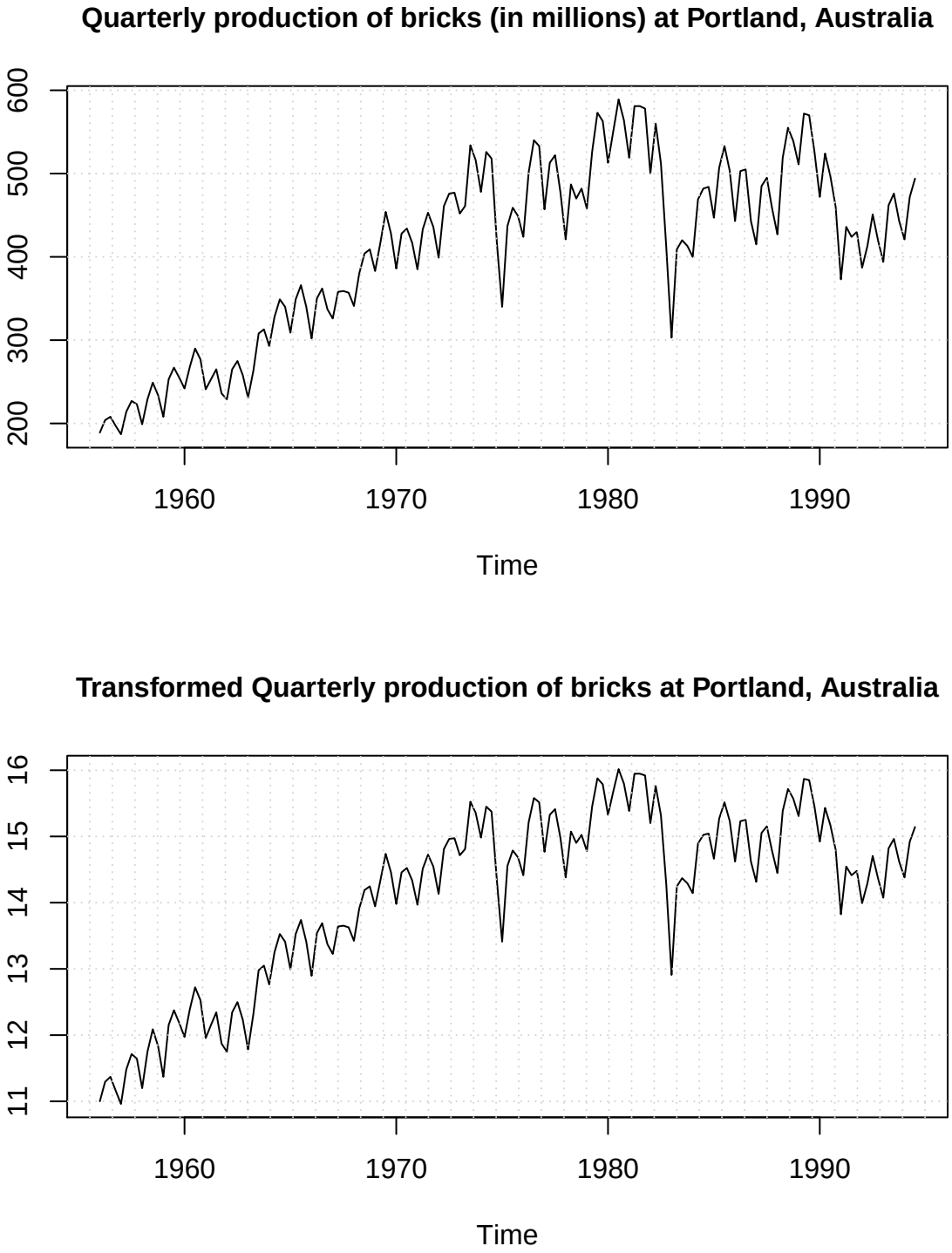


Figure 3: Top: Quarterly production of bricks (in millions) at Portland, Australia, Bottom : BoxCox transformed Quarterly production of bricks

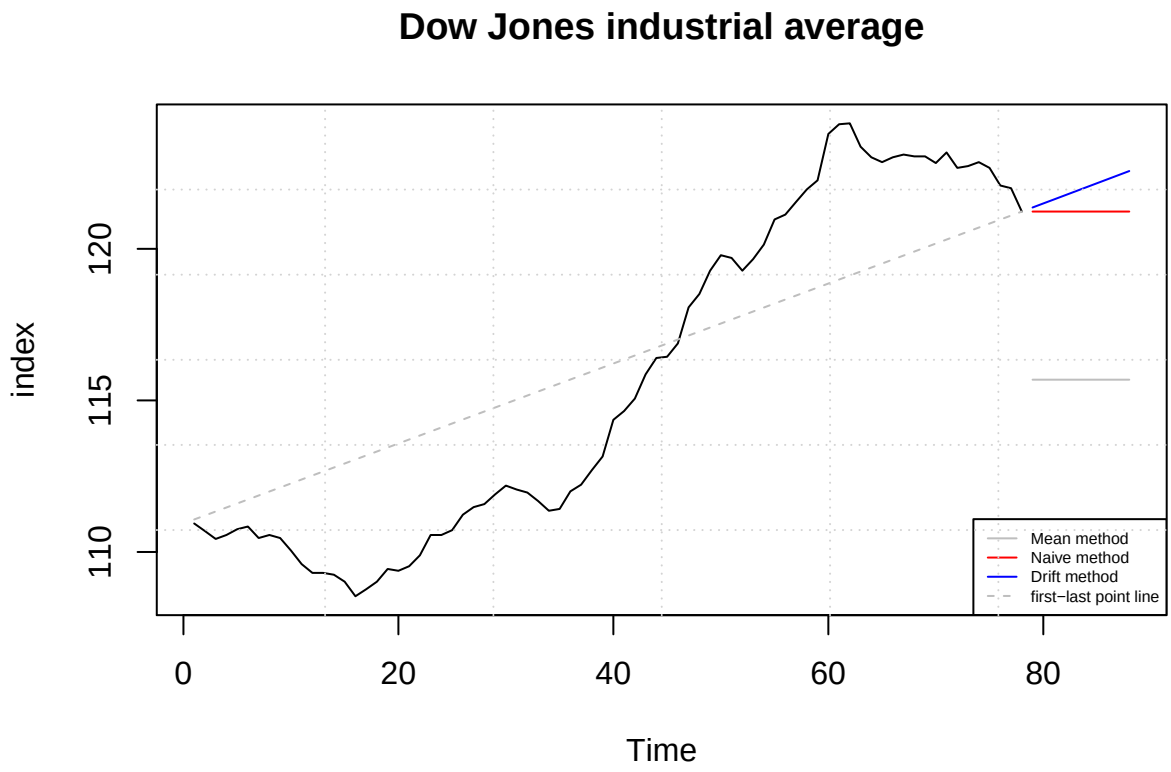


Figure 4: Time series modeling of Dow Jones industrial average

number of data points. The grey dashed line in fig 4 shows the fitted drift for the Dow Jones data.

```
data("dowjones")
slope <- (dowjones[78] - dowjones[1])/length(dowjones - 1)
y <- dowjones[1] + seq(1:78)*slope
driftmeth <- forecast::rwf(dowjones, h = 10, drift = T)
meanmethod <- forecast::meanf(dowjones,h=10)
naivemethod <- forecast::naive(dowjones,h =10)
seasonnaivemethod <- forecast::snaive(dowjones,h = 10)
plot(dowjones,xlim = c(1,88), ylab = "index", main = "Dow Jones industrial average")
lines(driftmeth$mean, col = "blue")
lines(y,col = "grey",lty =2)
lines(naivemethod$mean, col = "red")
lines(meanmethod$mean, col = "grey")
#lines(seasonnaivemethod$mean, col = "green")
legend("bottomright", legend = c("Mean method", "Naive method", "Drift method", "first-last point line")
grid(nx = 6)
```

```
knitr::kable(data.frame(Time = 79:88, index = driftmeth$mean), caption = "Drift method forecast for 10 p
```

Table 1: Drift method forecast for 10 periods

Time	index
79	121.3636
80	121.4973
81	121.6309
82	121.7645

Time	index
83	121.8982
84	122.0318
85	122.1655
86	122.2991
87	122.4327
88	122.5664

Comparison between models

The drift method is compared with other models like the Mean of the time series, the naive method and the seasonal naive method. The accuracy methods are compared in table 2. Since the time series is set up as a daily closing index. The seasonal forecast for the next point is identical as the previous point (same season as the last point). Hence the Naive and Seasonal Naive forecasts are the same for this scenario.

The Drift method is the best performing model amongt the competing naive models, based on MAE / MAPE. This is not surprising as we see an increasing trend in the Dow Jones index. The other methods use the mean or the last value of the time series as the forecast. The naive models do not account for the increasing trend. The Drift method does.

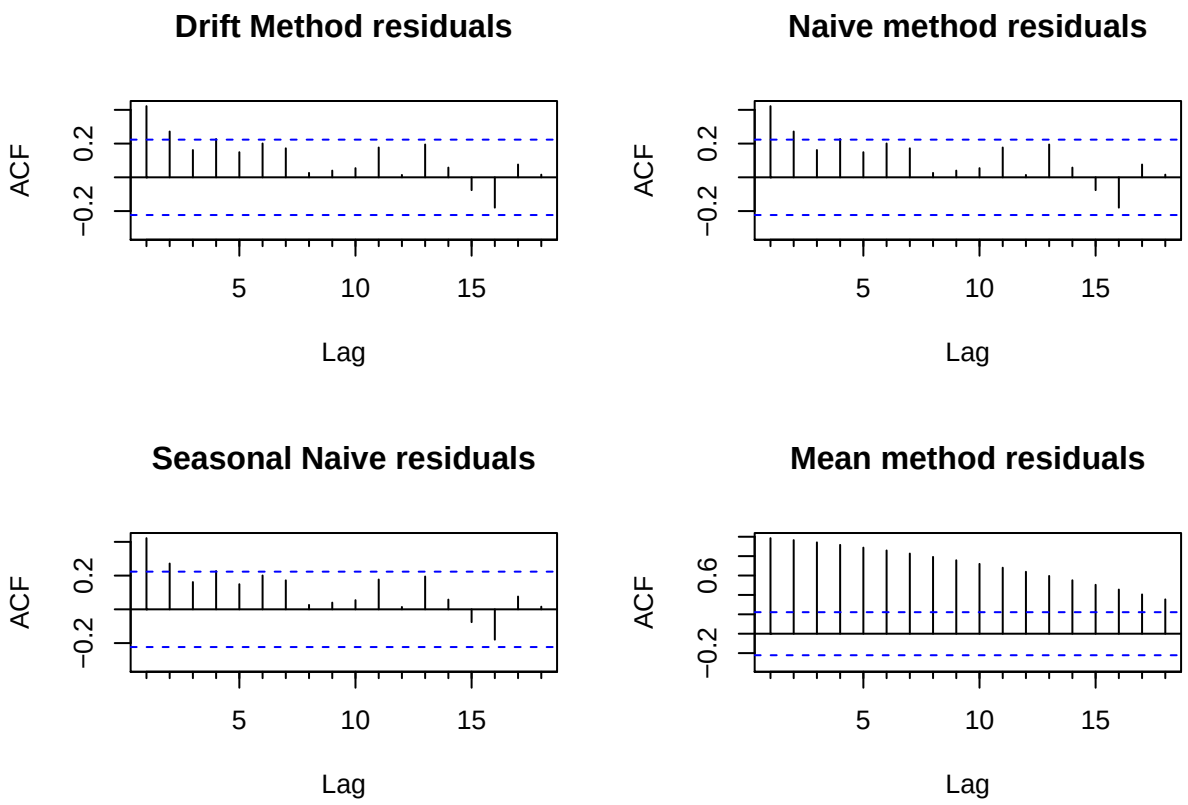


Figure 5: Autocorrelation plot of residuals

```
Metrics <- rbind.data.frame(Drift = accuracy(driftmeth), Mean = accuracy(meanmethod),Naive = accuracy(na
knitr::kable(round(Metrics,3), caption = "Accuracy metrics of benchmark models")
```

Table 2: Accuracy metrics of benchmark models

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Drift	0.000	0.424	0.325	-0.001	0.280	0.952	0.422
Mean	0.000	5.471	5.104	-0.222	4.400	14.938	0.985
Naive	0.134	0.445	0.342	0.114	0.294	1.000	0.422
SeasonalNaive	0.134	0.445	0.342	0.114	0.294	1.000	0.422

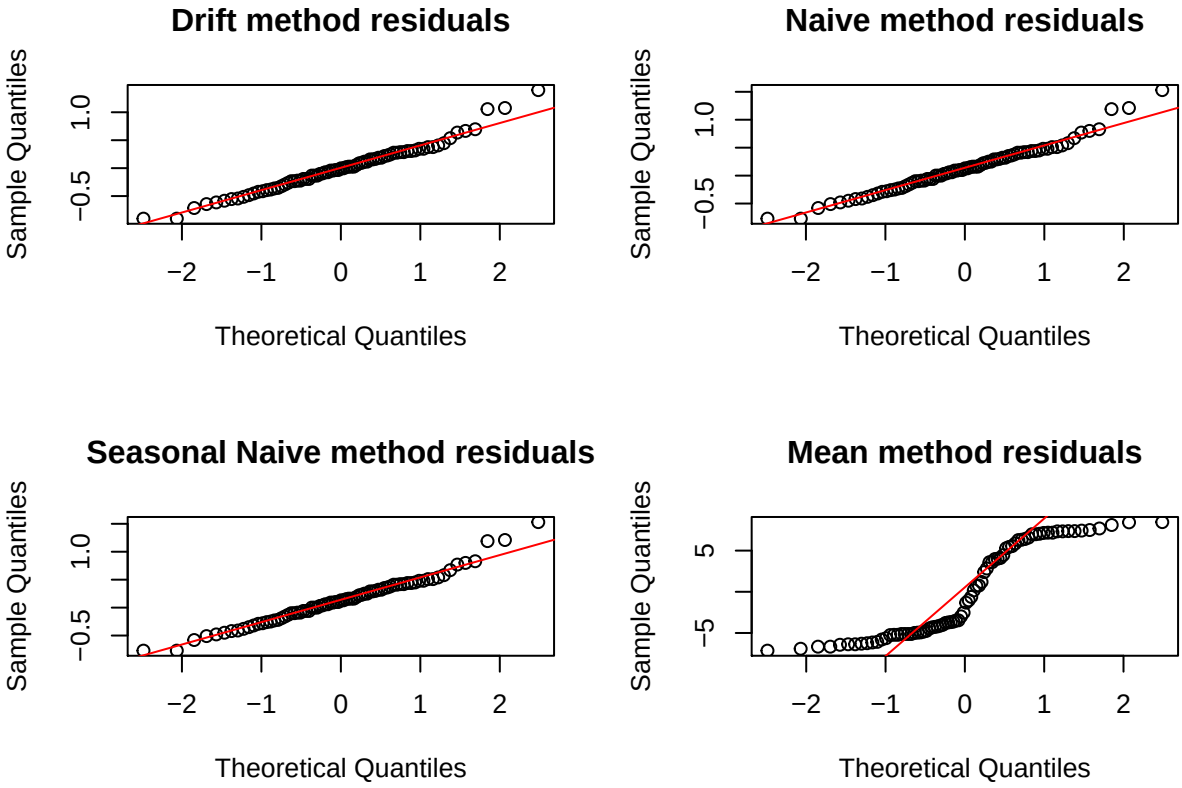
Residual Analysis

Figure 5 and 6 shows the residual analysis of the time series model fits. The mean model's residuals show that the means are significantly different from the other models and show it's poor performance.

```
par(mfrow = c(2,2))
Acf(driftmeth$residuals, main = "Drift Method residuals",cex.main = 0.6)
Acf(naivemethod$residuals, main = "Naive method residuals", cex.main = 0.5)
Acf(seasonnaivemethod$residuals, main = "Seasonal Naive residuals",cex.main = 0.5)
Acf(meanmethod$residuals, main = "Mean method residuals", cex.main = 0.5)
```

```
par(mfrow = c(2,2))
stats::qqnorm(as.numeric(driftmeth$residuals), type = "p",main = "Drift method residuals")
```

```
qqline(as.numeric(driftmeth$residuals), col = "red")
stats::qqnorm(as.numeric(naivemethod$residuals), type = "p", main = "Naive method residuals")
qqline(as.numeric(naivemethod$residuals), col = "red")
stats::qqnorm(as.numeric(seasonnaivemethod$residuals), type = "p", main = "Seasonal Naive method residua
qqline(as.numeric(seasonnaivemethod$residuals), col = "red")
stats::qqnorm(as.numeric(meanmethod$residuals), type = "p", main = "Mean method residuals")
qqline(as.numeric(meanmethod$residuals), col = "red")
```



Question 2.3 IBM Stock prices

Figure 8 shows the time series plot of IBM stock price at close. The data is split into training and test set. The first 300 data points are retained as training set and the rest of the 69 data points are held out as test set.

```
data("ibmclose")
layout(matrix(c(1,1,2),nrow = 1,ncol =3))
plot(ibmclose,ylab = "Stock Price", main = "Time series of IBM closing stock price")
hist(ibmclose,col = "grey")
```

```
train <- ts(ibmclose[1:300])
test <- ts(ibmclose[-1:-300],start = 301, end = 369)
```

Time series modeling of the training set

The Drift, naive and mean method modeling was trained on the training set and the accuracy statistics are shown in table 3. Since the distribution of the data is bimodal, BoxCox transformation was deemed not

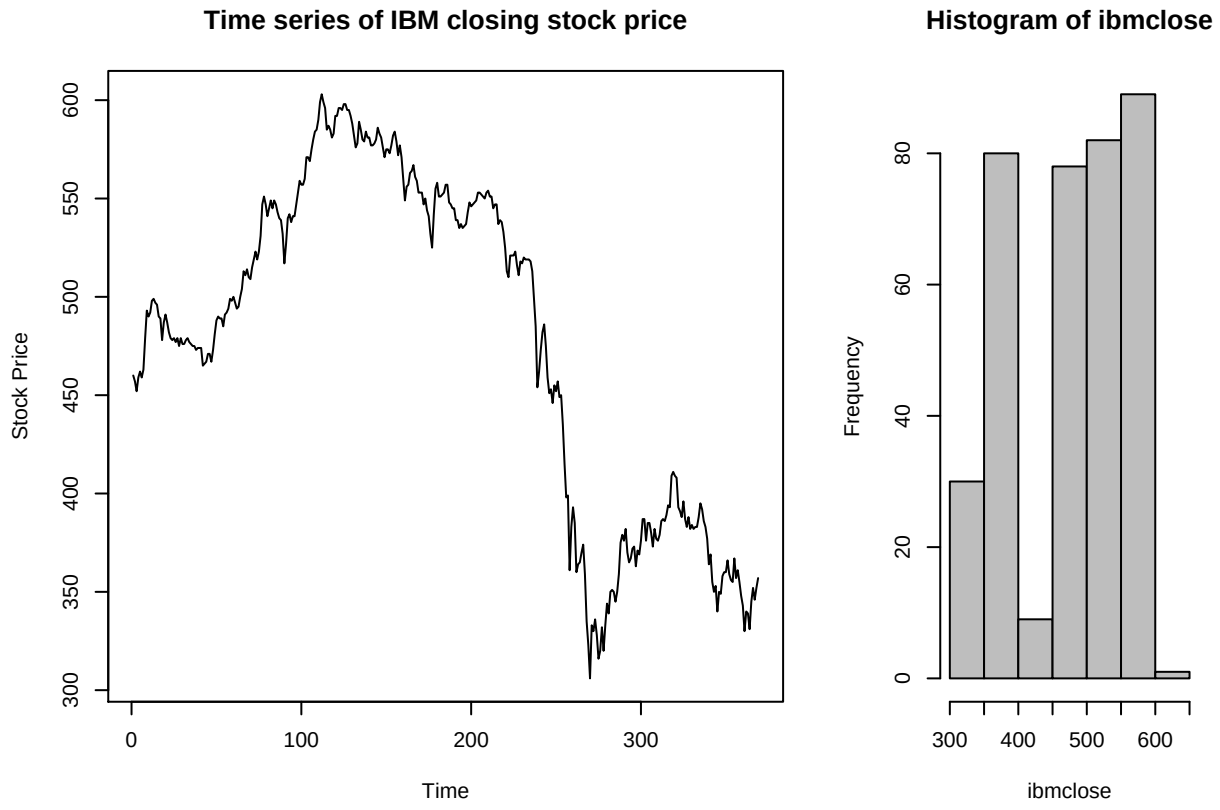


Figure 6: IBM Stock price at close

appropriate. Figure 9 shows the residual analysis. Both the naive and the drift methods are very close in their forecasts.

```
#plot(train, ylab = "Stock Price", main = "IBM closing stock price \n Training set")
drift.fit <- rwf(y = train,drift = T,h =1)
#plot(drift.fit)
mean.fit <- meanf(train,h = 1)
naive.fit <- naive(train,h=1)
knitr::kable(rbind.data.frame(Drift = round(accuracy(drift.fit),2),Naive = round(accuracy(naive.fit),2)
```

Table 3: Forecast Accuracy

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Drift	0.00	7.30	5.13	-0.03	1.12	1.01	0.14
Naive	-0.28	7.30	5.10	-0.08	1.12	1.00	0.14
Mean	0.00	73.62	58.72	-2.64	13.03	11.52	0.99

```
par(mar=c(2.5,2.5,1,1))
layout(matrix(c(1,2,3,4,1,5,3,6),ncol=2),heights = c(1,3,1,3))
plot.new()
text(0.5,0.5,"Drift Residuals",cex = 1)
hist(drift.fit$residuals, xlab = "Drift residuals", main = "")
plot.new()
text(0.5,0.5,"Naive Residuals",cex = 1)
hist(naive.fit$residuals, xlab = "Naive residuals", main ="")

Acf(drift.fit$residuals, main = "")
Acf(naive.fit$residuals, main = "")
```


Figure 10 shows the actuals and the forecasts along with the time series of the forecast errors. Both the naive and drift methods are very identical.

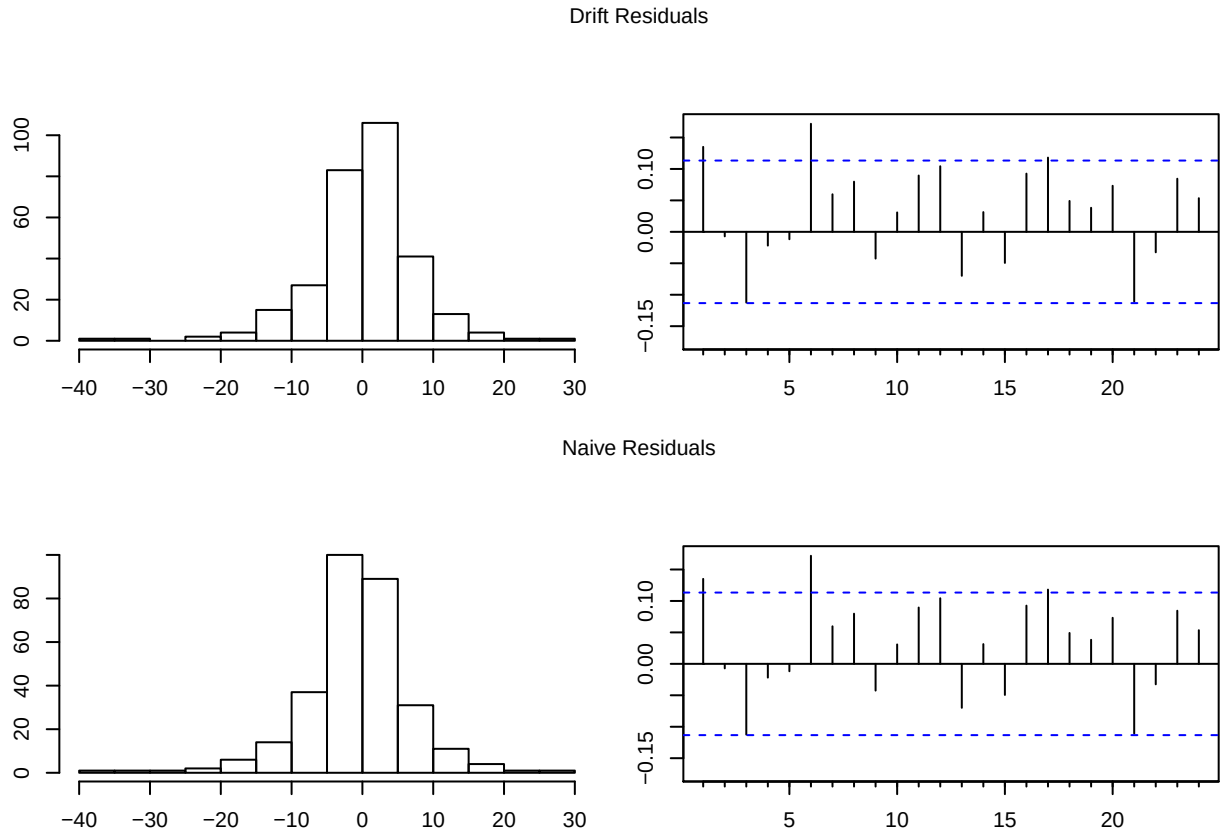


Figure 7: Residual Analysis

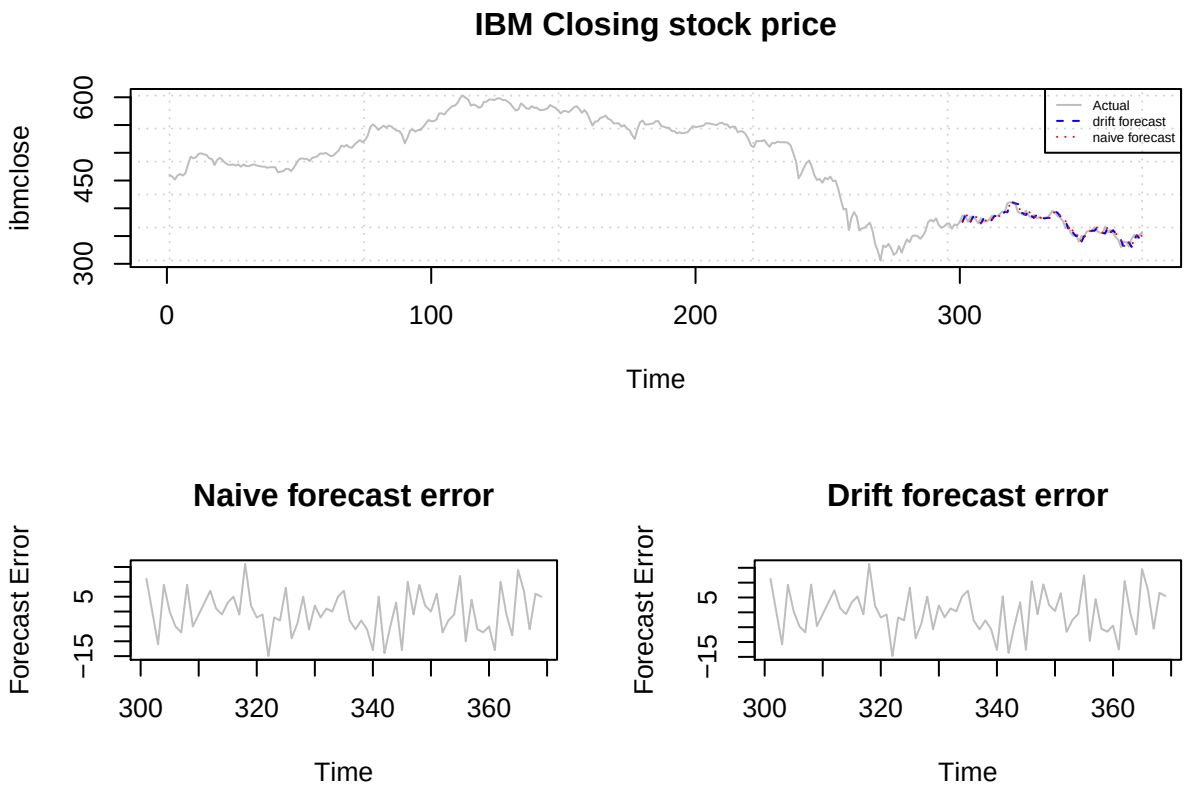


Figure 8: IBM stock price forecast

```
seasonplot(hsales,year.labels = T,col = 1:10,main = "")
monthplot(hsales)
salesperDay <- hsales/monthdays(hsales)
plot(salesperDay)
```

```
lambda <- BoxCox.lambda(salesperDay)
#plot(BoxCox(salesperDay,lambda))


# plot(decompose(BoxCox(salesperDay,lambda),"additive"))
# plot(decompose(BoxCox(salesperDay,lambda),"multiplicative"))
```

Model training

The time series data is split into training and test set. On the training set the naive, seasonal naive and the drift methods are fit. The drift and Naive models have identical error statistics.

```
training.hsales <- window(hsales,start = 1973, end = 1993-0.01)
test.hsales <- window(hsales, start = 1994,frequency = 12)

naive.fit2 <- naive(training.hsales,h=1)
snaive.fit2 <- snaive(training.hsales,h=1)
drift.fit2 <- rwf(training.hsales,h = 1, drift = T)

knitr::kable(round(rbind.data.frame(Naive = accuracy(naive.fit2), SeasonalNaive = accuracy(snaive.fit2)
```

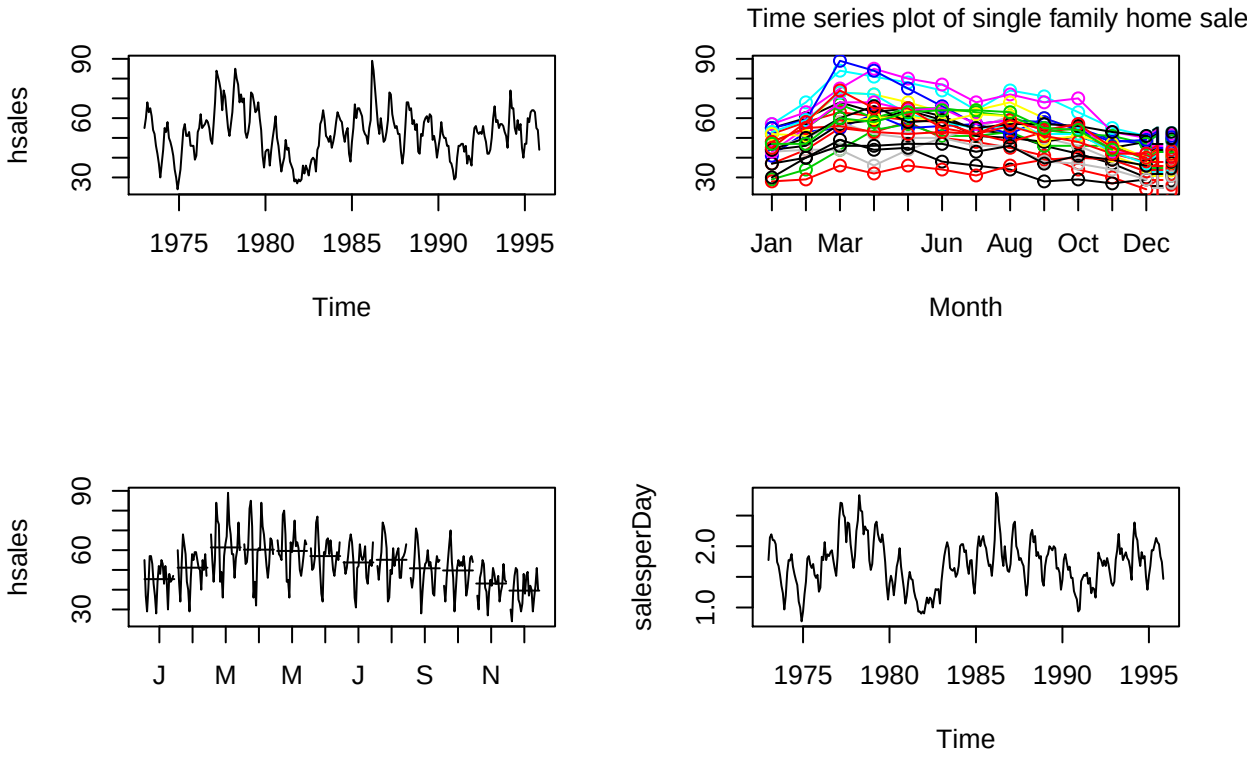


Figure 9: Time series plot of single family home sales in the US

Table 4: Accuracy statistics on training data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Naive	-0.05	6.36	5.05	-0.88	10.04	0.59	0.18
SeasonalNaive	-0.11	10.73	8.60	-2.64	17.96	1.00	0.84
Drift	0.00	6.36	5.05	-0.77	10.03	0.59	0.18

The same methods are fit on the sales normalized by the number of days in a month i.e Sales per day. The accuracy of the training set is seen in the table below. Again the Naive and the drift methods are identical.

```
training.salesPerDay <- window(salesperDay, start = 1973,end = 1993-0.01)
naive.fit3 <- naive(training.salesPerDay,h=10)
snaive.fit3 <- snaive(training.salesPerDay)
drift.fit3 <- rwf(training.salesPerDay,h = 10, drift = T)

knitr::kable(round(rbind.data.frame(Naive = accuracy(naive.fit3),
Seasonal = accuracy(snaive.fit3),
Drift = accuracy(drift.fit3)),2),caption = "Accuracy statistics on Sales per data- training data")
```

Table 5: Accuracy statistics on Sales per data- training data

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Naive	0	0.21	0.17	-0.91	10.09	0.59	0.25
Seasonal	0	0.35	0.28	-2.65	17.93	1.00	0.85
Drift	0	0.21	0.17	-0.80	10.08	0.59	0.25

The models are fit on a tranformed data set. While transformed data points of values zero make the MAPE; infinite. The MAE and other statistics convey the same message of drift and Naive being equal.

```
training.salesPerDayTrans <- BoxCox(training.salesPerDay,BoxCox.lambda(training.salesPerDay))
```

Test errors

test.hsalesPerMonth <- test.hsales/monthdays(test.hsales) test.hsalesPerMonth.Naive <- ts(sapply(1:23,function(x) naive(hsales/monthdays(hsales)[1:240+x-1], h = 1)$mean),start = 1994,frequency = 12) accuracy(test.hsalesPerMonth.Naive,test.hsalesPerMonth)

plot(test.hsales/monthdays(test.hsales) - ts(sapply(1:23,function(x) rwf(hsales/monthdays(hsales)[1:240+x-1], h = 1,drift = T)$mean),start = 1994,frequency = 12))

accuracy(test.hsales/monthdays(test.hsales),ts(sapply(1:23,function(x) naive(hsales/monthdays(hsales)[1:240+x-1], h = 1)$mean),start = 1994,frequency = 12)) “‘