Assignment1_Seshadri

Sri Seshadri

4/7/2018

Contents

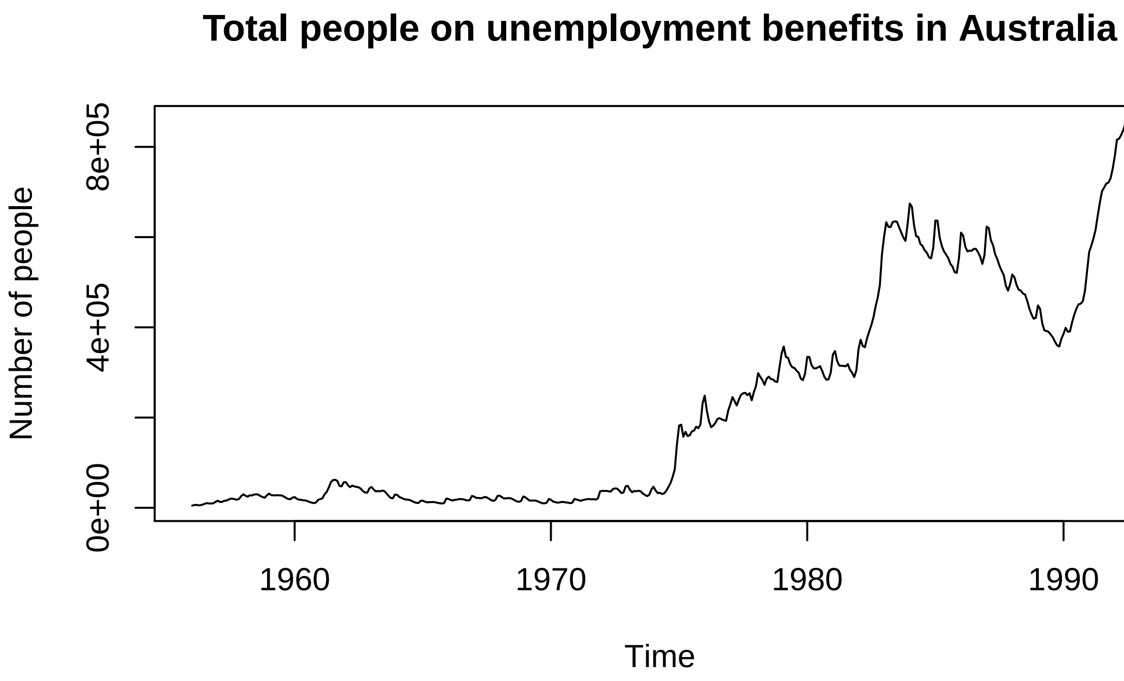


Figure 1: Monthly number of people on unemployment benefits in Australia

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.

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.

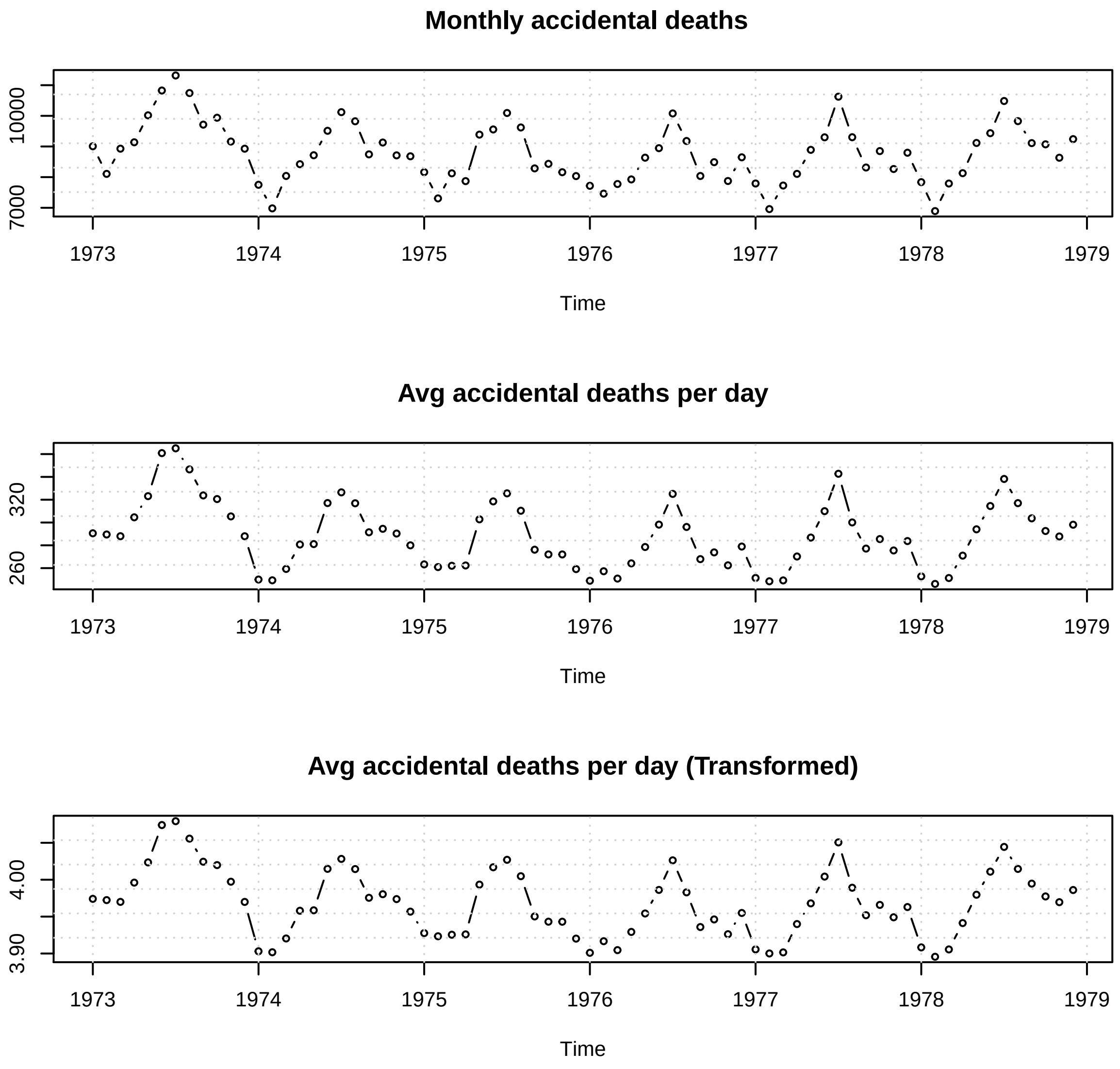
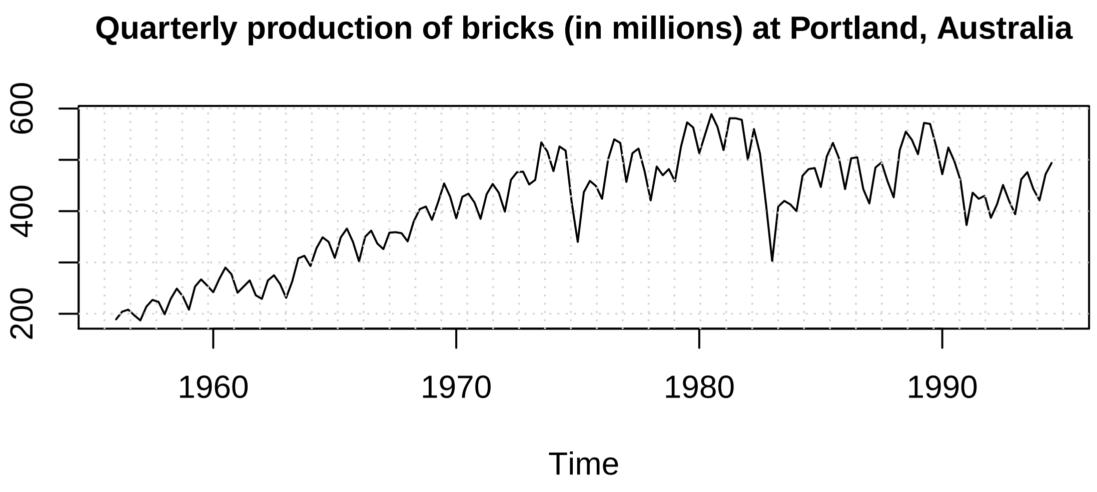


Figure 2: Top: Total accidental deaths by month in the US, Middle: Monthly Average Accidental deaths per day, Bottom: Transformed monthly average accidental deaths per day



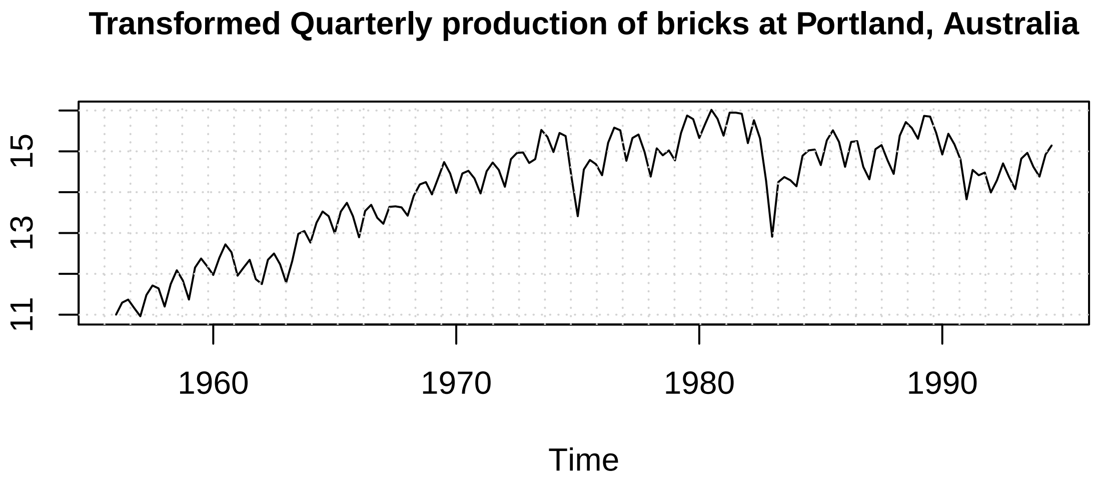


Figure 3: Top: Quarterly production of bricks (in millions) at Portland, Australia, Bottom : BoxCox transformed Quarterly production of bricks

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.

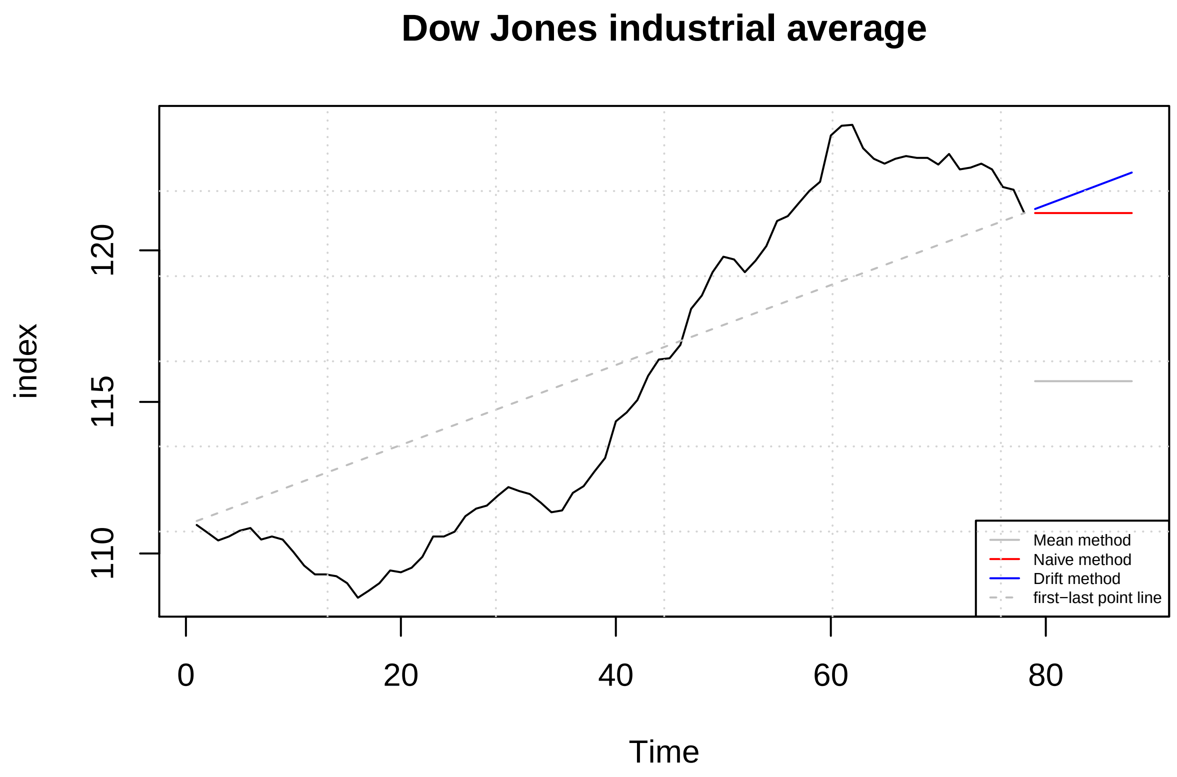


Figure 4: Time series modeling of Dow Jones industrial average

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

Table 1: Drift method forecast for 10 periods

Time	index
79	121.3636
80	121.4973
81	121.6309
82	121.7645
83	121.8982
84	122.0318
85	122.1655
86	122.2991
87	122.4327
88	122.5664

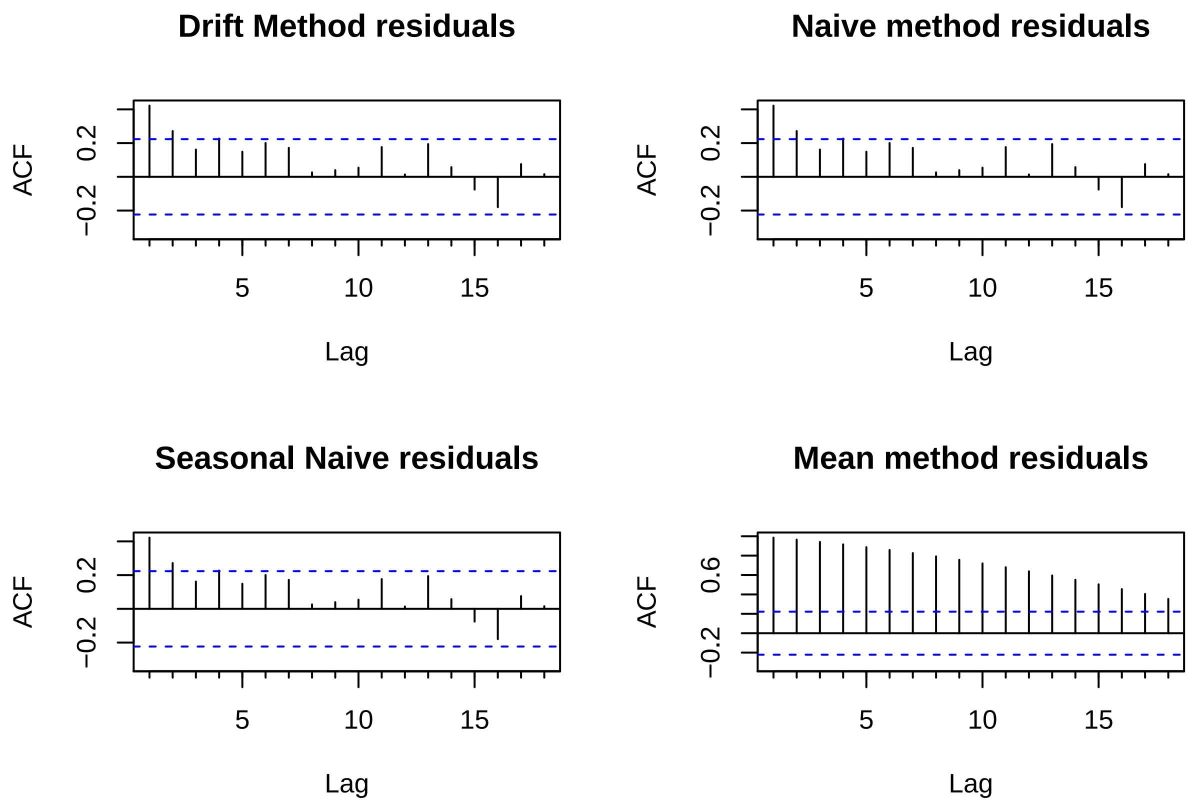


Figure 5: Autocorrelation plot of residuals

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.

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.

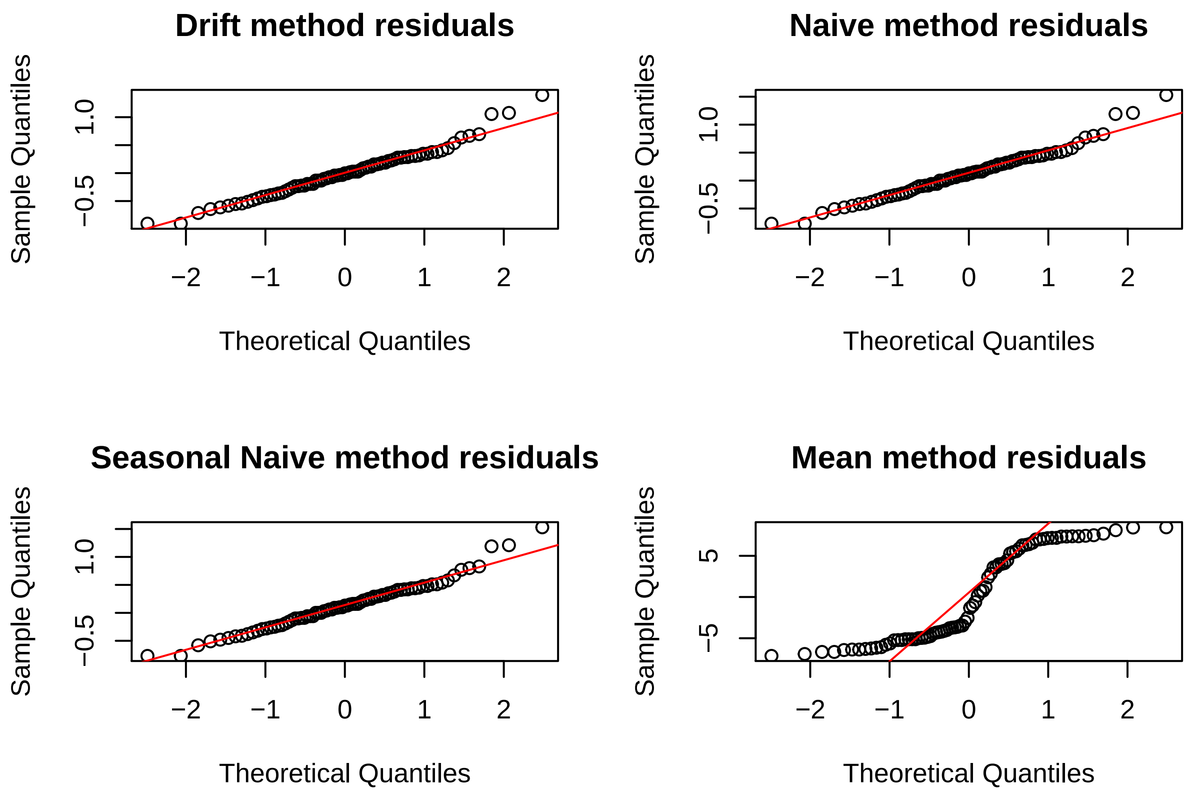


Figure 6: Residual Analysis

Question 2.3 IBM Stock prices

Figure 7 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.

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

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

Figure 9 shows the actuals and the forecasts along with the time series of the forecast errors. Both the naive and drift methods are very identical.

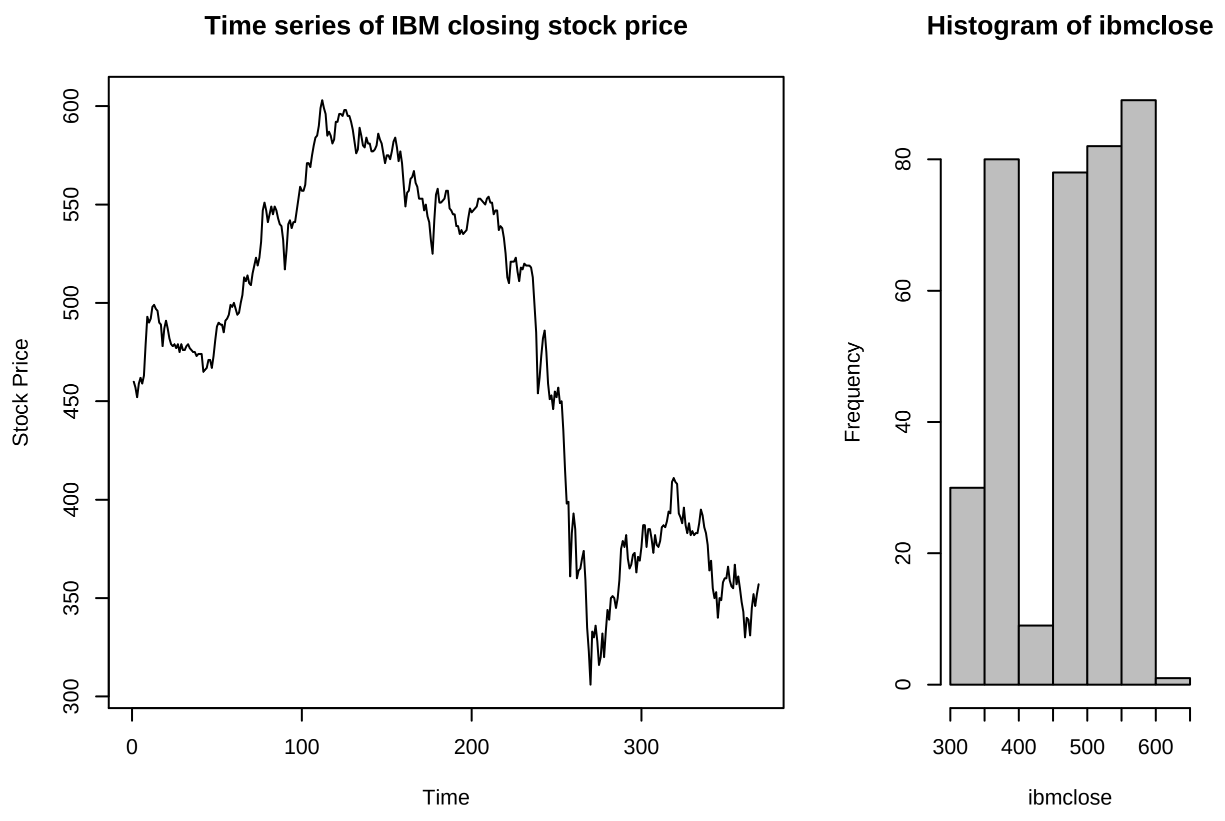


Figure 7: IBM Stock price at close

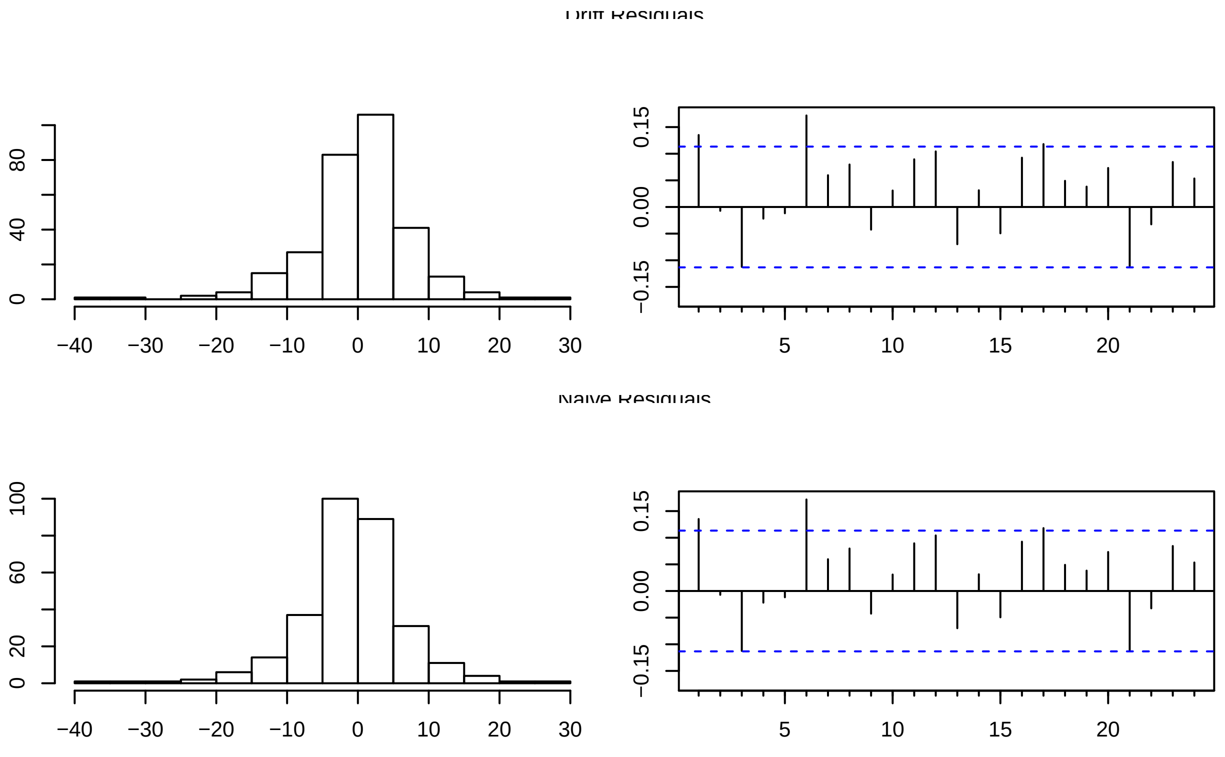


Figure 8: Residual Analysis

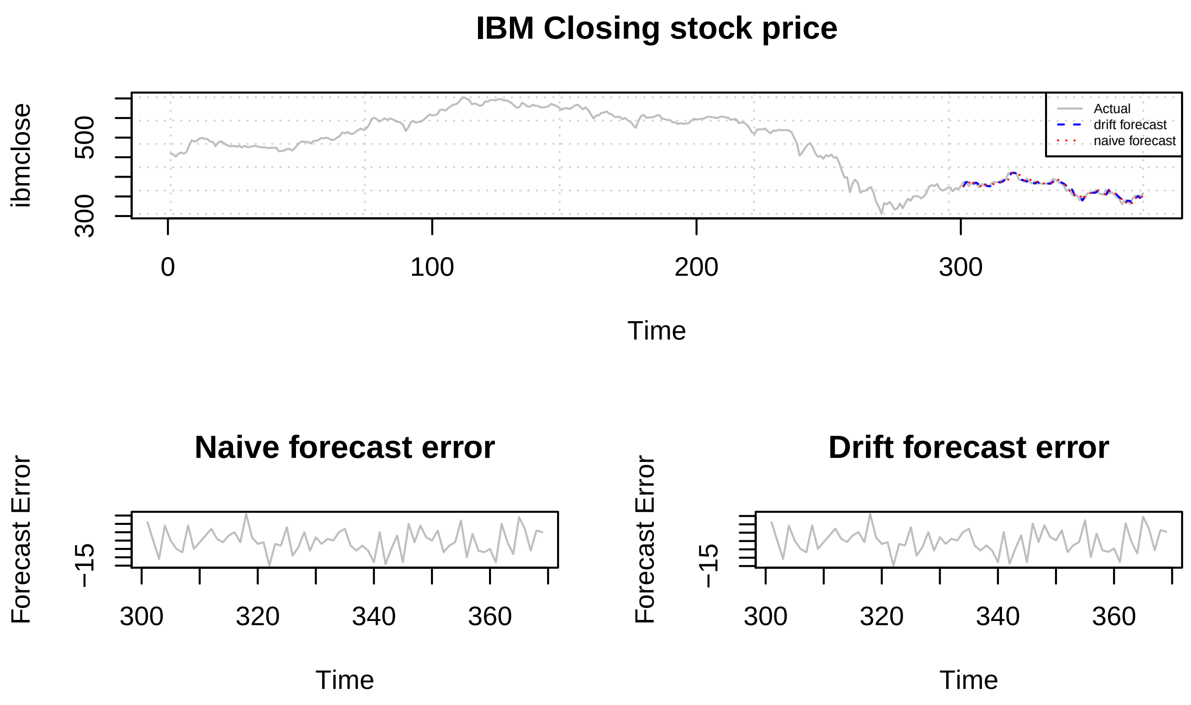


Figure 9: IBM stock price forecast

2.4 Single family house sales in the US

Figure 10 shows the time series plot of home sales in the US. The top right panel shows that there is seasonality in the time series. The bottom left panel shows the variation of sales by month over years. The data is attempted to be smoothed by plotting the sales per day over years.

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.

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.

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.

Table 6: Accuracy on Transformed training data

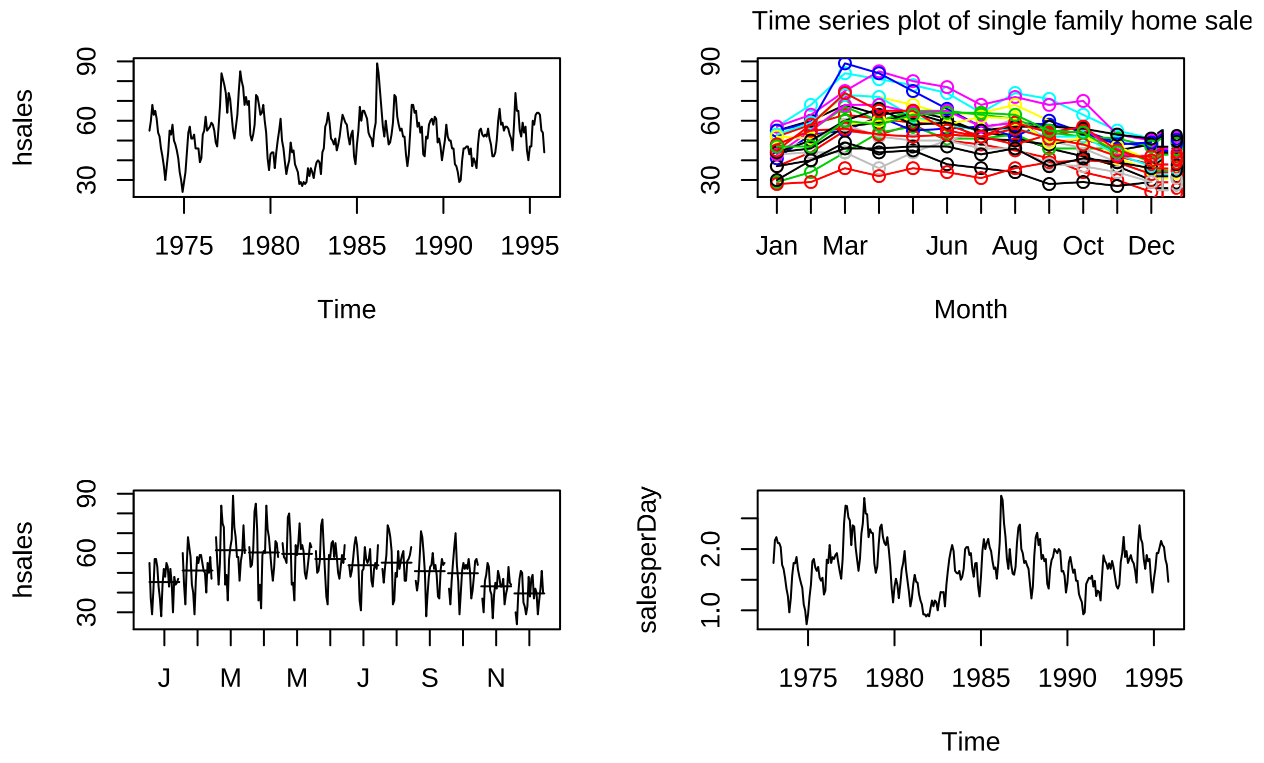


Figure 10: Time series plot of single family home sales in the US

Table 8: Forecast error on test data - Sales per day per month

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Naive	0.4	0.48	0.41	19.71	20.67	0.5	1.83
Drift	0.4	0.48	0.41	19.79	20.73	0.5	1.84

Section 4.10 Question 4.1

a) Figure 11 shows rthe relationship between temperature and electricity consumption. The relationship is negative probabaly because the usage is for heating purposes.

b) The model though is statistically significant has an R-Squared of only 59%. Figure 12 shows the residual plots. The residuals are fairly random. Observation 8 has the maximum residual error.

```
##
## Call:
## lm(formula = Mwh ~ temp, data = econsumption)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2593 -0.5395 -0.1827  0.4274  2.1972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.19952    0.73040   27.66 8.86e-11 ***
## temp        -0.14516    0.03549   -4.09  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

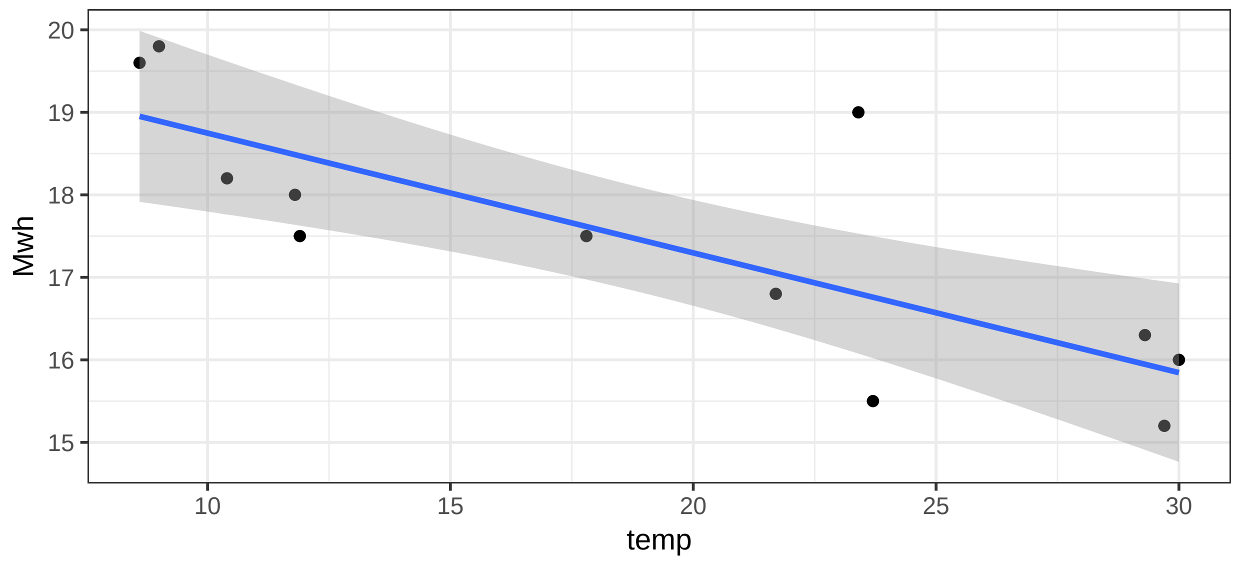


Figure 11: Electricity consumption by temperature

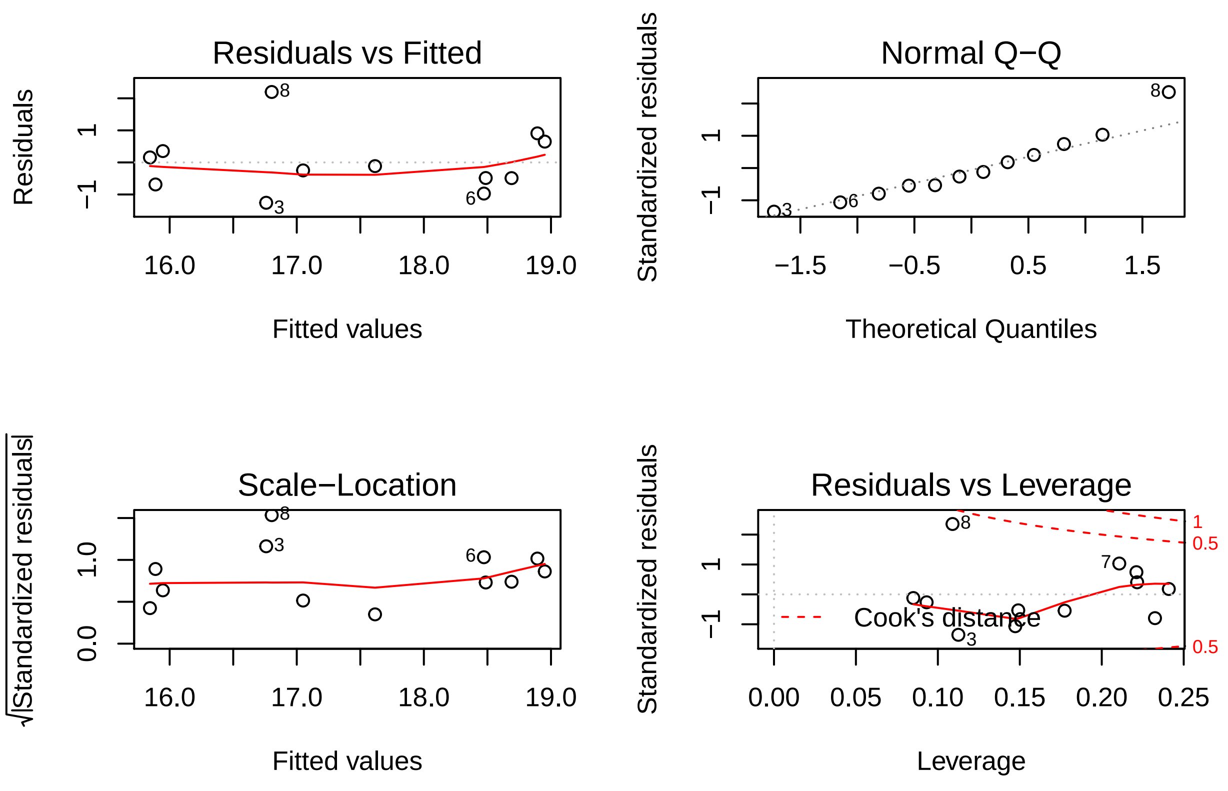


Figure 12: Residual plots

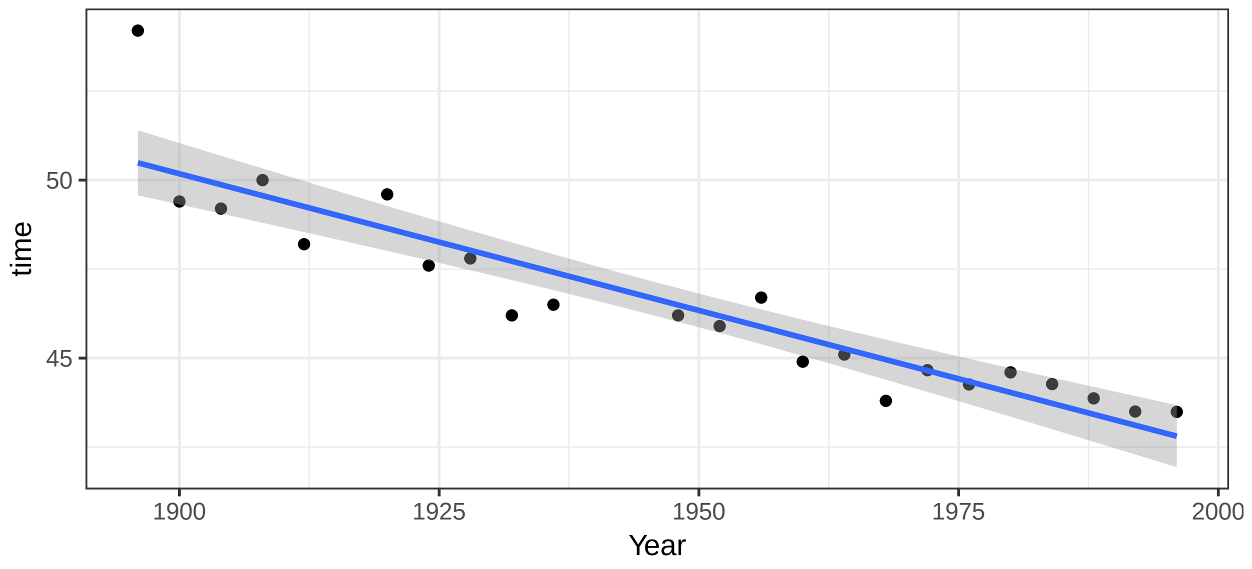


Figure 13: Winning times of 400 meters dash

Section 4.10 Question 4.2

a & b) Figure 13 shows the winning times of 400m dash in olympic finals between 1900 to 2000. There is a decreasing trend over years.

c) The decrease is at the rate of 7.6%

d) The residual point shows there is one influential point - observation 1. This point can lead to bias in the model fit. However otherwise the model has a 83% R-Squared.

e and f) The table below shows the prediction and the actual times. The assumption made in using this model is :

- There is no human limitation for running;
- the location of the race does not contribute to performance of the athletes.
- The residuals follow a normal distribution.

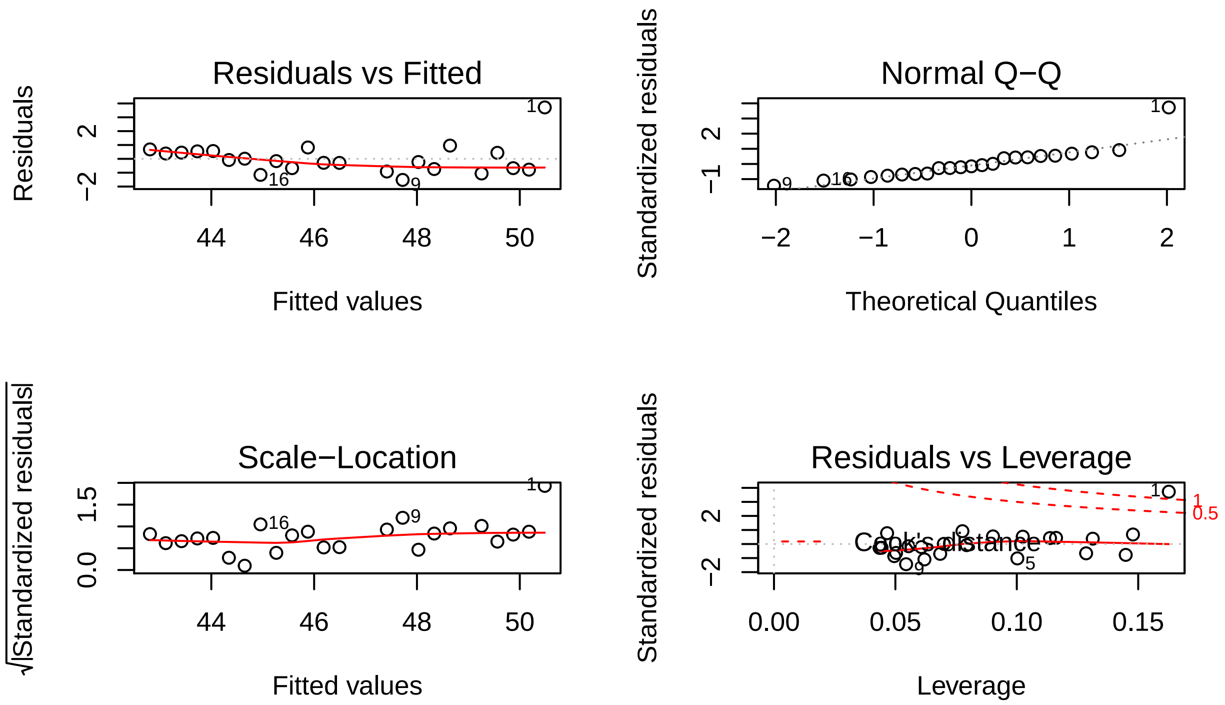


Figure 14: Residual Analysis

```
##
## Call:
## lm(formula = time ~ Year, data = olympic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5215 -0.7037 -0.1642  0.4952  3.7141
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 196.079876  14.177031   13.83 5.09e-12 ***
## Year         -0.076790   0.007278  -10.55 7.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.089 on 21 degrees of freedom
## Multiple R-squared:  0.8413, Adjusted R-squared:  0.8337
## F-statistic: 111.3 on 1 and 21 DF,  p-value: 7.515e-10
```

Table 9: Prediction of run times

Year	Prediction	Actuals
2000	42.49977	43.84
2004	42.19261	44.00
2008	41.88545	43.75
2012	41.57829	43.94

Section 4.10 Question 3

Elesticity is defined as $(dy/dx)(x/y)$. The Regression model is as follows:

$$logy = \beta_0 + log\beta_1 x + \varepsilon$$

$$y = e^{\beta_0 + log\beta_1 x + \varepsilon}$$

Differentiating y by x, we get . . .

$$dy/dx = (e^{\beta_0 + log\beta_1 x + \varepsilon})\frac{\beta_1}{x}$$

$$(\frac{dy}{dx})(\frac{x}{y}) = (e^{\beta_0 + log\beta_1 x + \varepsilon})(\frac{\beta_1}{x})(\frac{x}{y})$$

$$(\frac{dy}{dx})(\frac{x}{y}) = (y)(\frac{\beta_1}{x})(\frac{x}{y})$$

$$(\frac{dy}{dx})(\frac{x}{y}) = \beta_1$$

Section 6.7 Question 6.1

Prove 3 X 5 MA = weighted 7 MA with (0.067,0.133,0.2,0.2,0.2,0.133,0.067)

The weights can be converted into fraction (1/15, 2/15, 1/5, 1/5,1/5, 2/15, 1/15)

$$3X5MA = \frac{1}{15}[y_{t-3}+y_{t-2}+y_{t-1}+y_t+y_{t+1}]+\frac{1}{15}[y_{t-2}+y_{t-1}+y_t+y_{t+1}+y_{t+2}]+\frac{1}{15}[y_{t-1}+y_t+y_{t+1}+y_{t+2}+y_{t+3}]$$

$$= \frac{1}{15}y_{t-3} + \frac{2}{15}y_{t-2} + \frac{1}{5}y_{t-1} + \frac{1}{5}y_t + \frac{1}{5}y_{t+1} + \frac{2}{15}y_{t+2} + \frac{1}{15}y_{t+3}$$

The above equation is identical to that of 7 MA with weights (1/15, 2/15, 1/5, 1/5,1/5, 2/15, 1/15)

Section 6.7 Question 6.2

a) Figure 15 shows the time series of sales of manufacturer A. It has a seasonality of period 1 year and an increasing trend.

b & c) The classical multiplicative decomposing supports the interpretation of figure 16.

d,g and h) Figure 17 shows the time series plot of the seasonally adjusted data.

The drift method forecasts from the seasonally adjusted data is multiplied with the seasonal components and the mean of the random error to produce the forecasts for the next 2 years. The plot of the data is shownn figure 17.

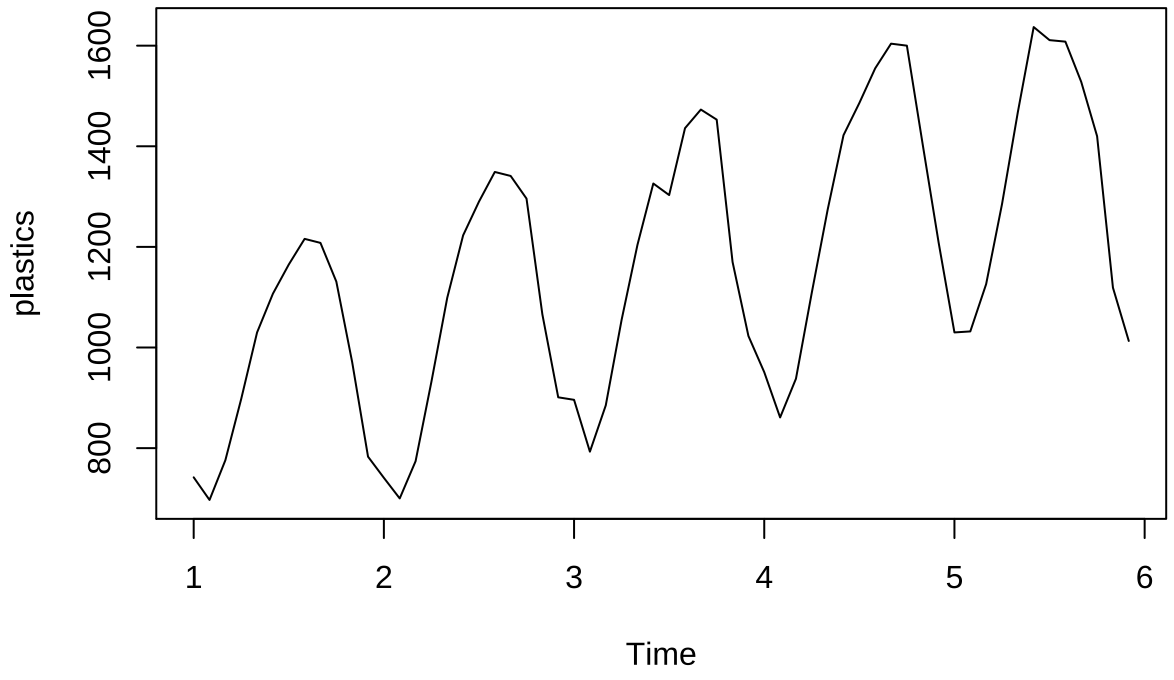


Figure 15: Time series plot of sales

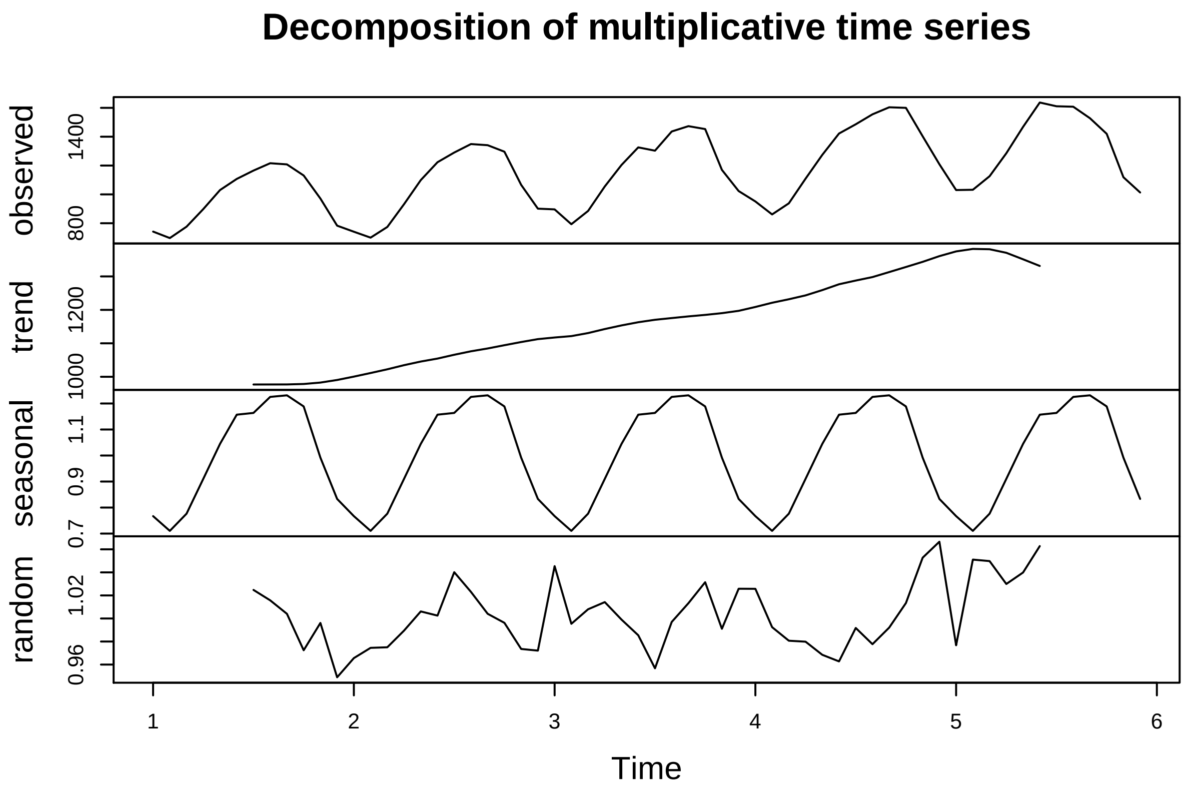


Figure 16: Classical multiplicative decomposing

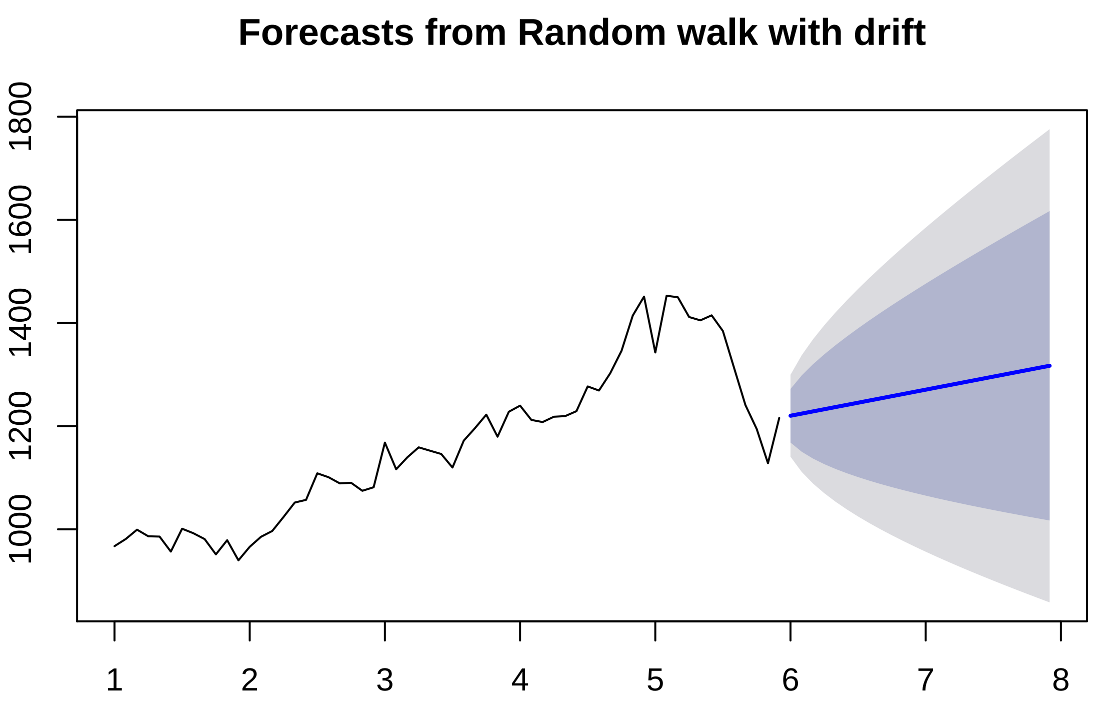


Figure 17: Seasonally Adj Data

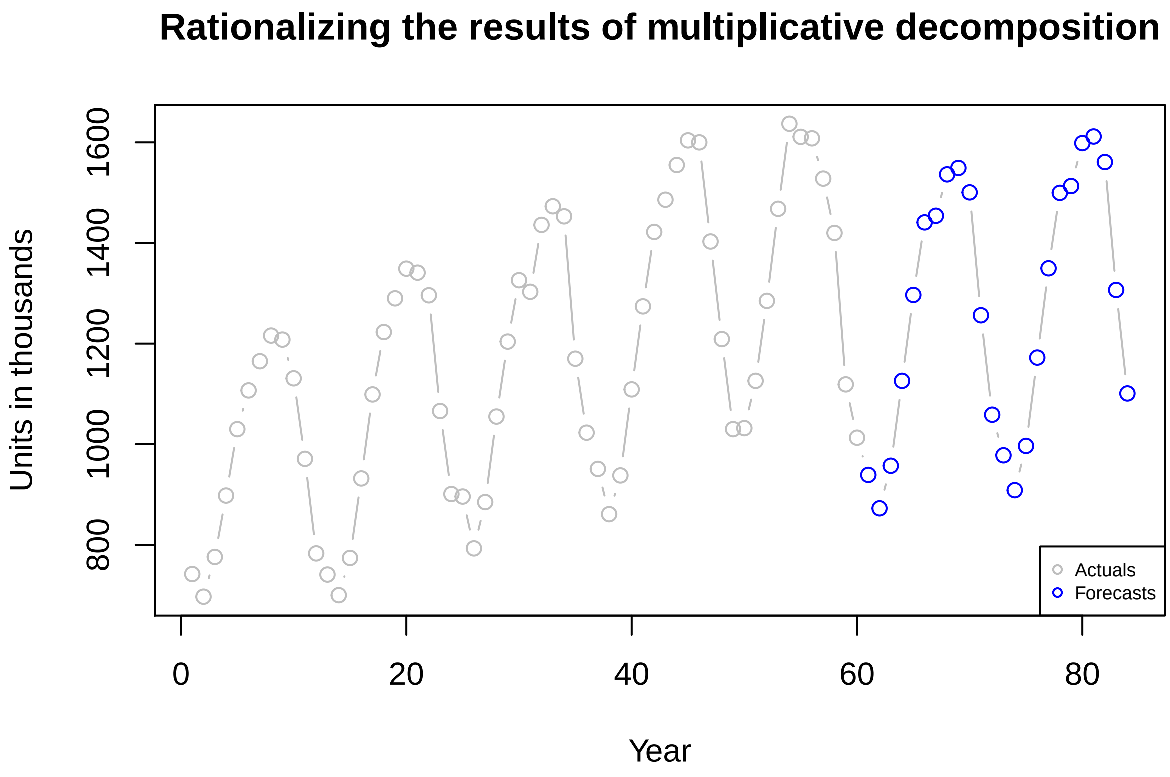


Figure 18: Rationalizing results

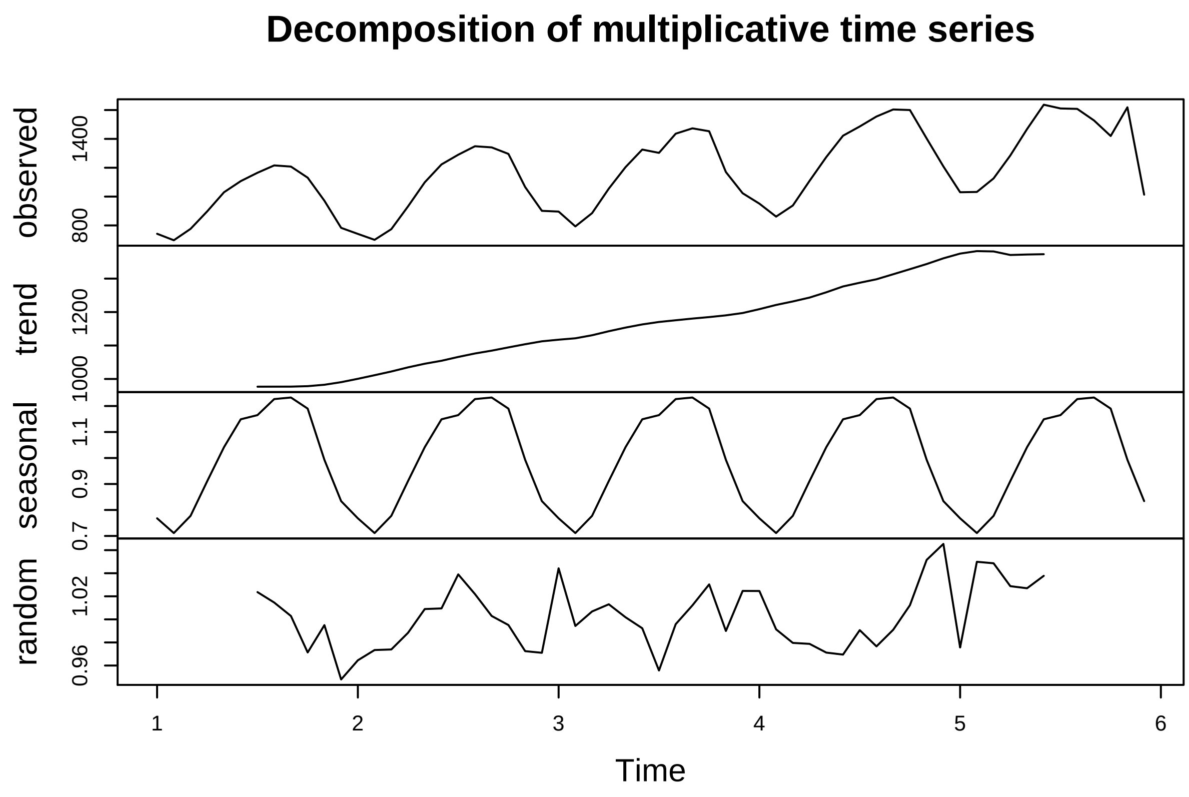


Figure 19: Added 500 to december of year 5

e, f) Figure 19 shows the decomposition when 500 units is added to Nov of year 5. The effect is none, this is because of the classical decomposition that has a disadvantage of losing predictions in the begining and ending of the time series (loss of half year each).

Figure 20 Shows the decomposition when 500 units is added to June of year 4. This has a slight effect on the predictions and it can be seen in the random error chart as a spike. The moving average smoothes over the spike to minimize its effect.

Section 6.7 Question 6.3

a) The trend component is the major source of variation in the data. There is a steady increase (constant rate) for the most part from 1978 to 1995. The seasonal component and the random error each take up only about 8% (16% in total) of the varaition in the data. The month of November and Feb has not changed year over year. The month of March has seen much volatility. Likely because of the recession in 91' & 92'.

b) the recession is slightly visible in the trend, but not much. However the random error spikes in the negative direction shows that the model did not pick up the recession.

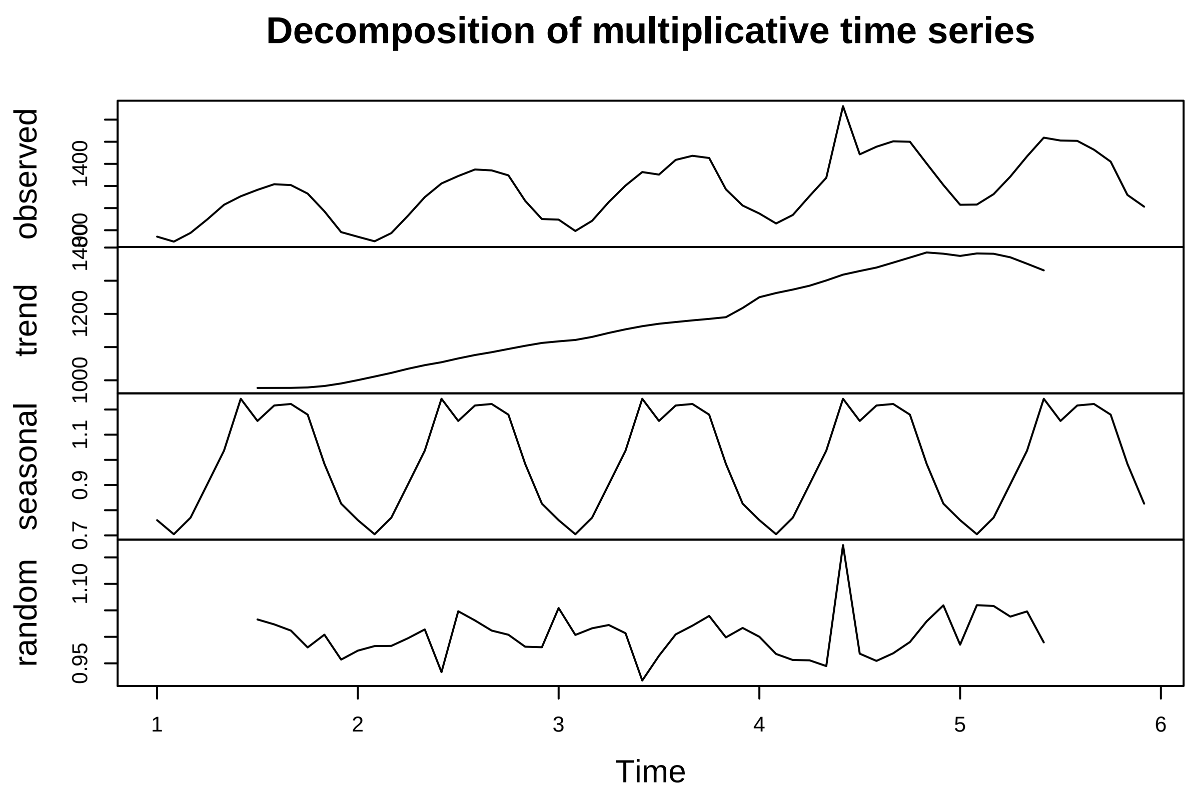


Figure 20: Added 500 to Jun of Year 4

Appendix

```
knitr::opts_chunk$set(echo = F)
library(magrittr)
library(dplyr)
library(fpp)
library(forecast)
library(fma)
data(package = "fma", dole)
plot(dole, main = "Total people on unemployment benefits in Australia",
    ylab = "Number of people")
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 per day (Transformed)", cex.lab = 0.4,
    ylab = "")
```

```r
grid(nx = NULL, ny = 72/12, col = "lightgray", lty = "dotted",
    lwd = par("lwd"), equilogs = TRUE)
data("bricksq")
par(mfrow = c(2, 1))
plot(bricksq, main = "Quarterly production of bricks (in millions) at Portland, Australia",
    cex.main = 1, ylab = "")
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.main = 1, ylab = "")
grid(ny = NULL, nx = 156/4, col = "lightgray", lty = "dotted",
    lwd = par("lwd"), equilogs = TRUE)
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"), col = c("grey",
    "red", "blue", "grey"), lty = c(1, 1, 1, 2), cex = 0.5)
grid(nx = 6)

knitr::kable(data.frame(Time = 79:88, index = driftmeth$mean),
    caption = "Drift method forecast for 10 periods")
Metrics <- rbind.data.frame(Drift = accuracy(driftmeth), Mean = accuracy(meanmethod),
    Naive = accuracy(naivemethod), SeasonalNaive = accuracy(seasonnaivemethod))
knitr::kable(round(Metrics, 3), caption = "Accuracy metrics of benchmark models")

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$residuals), type = "p",
    main = "Seasonal Naive method residuals")
qqline(as.numeric(seasonnaivemethod$residuals), col = "red")
```

```r
stats::qqnorm(as.numeric(meanmethod$residuals), type = "p", main = "Mean method residuals")
qqline(as.numeric(meanmethod$residuals), col = "red")
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)
# 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), Mean = round(accuracy(mean.fit),
    2)), caption = "Forecast Accuracy")
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 = "")
naive.generatedForecasts <- ts(sapply(1:69, function(x) naive(ibmclose[1:300 +
    x - 1], h = 1)$mean), start = 301, end = 369)
drift.generatedForecasts <- ts(sapply(1:69, function(x) rwf(ibmclose[1:300 +
    x - 1], h = 1, drift = TRUE)$mean), start = 301, end = 369)
layout(matrix(c(1, 1, 2, 3), byrow = T, ncol = 2), heights = c(3,
    2.5))
plot(ibmclose, col = "grey", panel.first = grid(), main = "IBM Closing stock price")
lines(drift.generatedForecasts, col = "blue", lty = 2)
lines(naive.generatedForecasts, col = "red", lty = 3)
legend("topright", legend = c("Actual", "drift forecast", "naive forecast"),
    lty = c(1:3), col = c("grey", "blue", "red"), cex = 0.5)
plot(test - naive.generatedForecasts, ylab = "Forecast Error",
    main = "Naive forecast error", col = "grey")
plot(test - drift.generatedForecasts, ylab = "Forecast Error",
    main = "Drift forecast error", col = "grey")

par(mfrow = c(2, 2))
plot(hsales)
mtext("Time series plot of single family home sales in the US",
    adj = -3, padj = 0.5, cex = 0.8, line = 1)
seasonplot(hsales, year.labels = T, col = 1:10, main = "")
monthplot(hsales)
salesperDay <- hsales/monthdays(hsales)
plot(salesperDay)
```

```r
lambda <- BoxCox.lambda(salesperDay)
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), Drift = accuracy(drift.fit2)),
    2), caption = "Accuracy statistics on training data")
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")
training.salesPerDayTrans <- BoxCox(training.salesPerDay, BoxCox.lambda(training.salesPerDay))
naive.fit4 <- naive(training.salesPerDayTrans, h = 10)
snaive.fit4 <- snaive(training.salesPerDayTrans)
drift.fit4 <- rwf(training.salesPerDayTrans, h = 10, drift = T)

knitr::kable(round(rbind.data.frame(Naive = accuracy(naive.fit4),
    Seasonal = accuracy(snaive.fit4), Drift = accuracy(drift.fit4)),
    2), caption = "Accuracy on Transformed training data")
Naive.test.forecast <- ts(sapply(1:23, function(x) naive(hsales[1:240 +
    x - 1], h = 1)$mean), start = 1994, frequency = 12)
Drift.test.forecast <- ts(sapply(1:23, function(x) rwf(hsales[1:240 +
    x - 1], h = 1, drift = T)$mean), start = 1994, frequency = 12)

knitr::kable(round(rbind.data.frame(Naive = accuracy(Naive.test.forecast,
    test.hsales), Drift = accuracy(Drift.test.forecast, test.hsales)),
    2), caption = "Forecast error on test data")
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)$mean), start = 1994, frequency = 12)
test.hsalesPerMonth.Drift <- ts(sapply(1:23, function(x) rwf(hsales/monthdays(hsales)[1:240 +
    x - 1], h = 1, drift = T)$mean), start = 1994, frequency = 12)

knitr::kable(round(rbind.data.frame(Naive = accuracy(test.hsalesPerMonth.Naive,
    test.hsalesPerMonth), Drift = accuracy(test.hsalesPerMonth.Drift,
    test.hsalesPerMonth)), 2), caption = "Forecast error on test data - Sales per day per month")

library(ggplot2)
ggplot(data = econsumption, mapping = aes(x = temp, y = Mwh)) +
    geom_point() + geom_smooth(method = "lm") + theme_bw()

lm.fit <- lm(Mwh ~ temp, data = econsumption)
summary(lm.fit)
```

```r
par(mfrow = c(2, 2))
plot(lm.fit)
forecast(lm.fit, newdata = data.frame(temp = c(10, 35)))
data("olympic")
ggplot(data = olympic, mapping = aes(x = Year, y = time)) + geom_point() +
    geom_smooth(method = "lm") + theme_bw()

lmfit.oly <- lm(time ~ Year, data = olympic)
summary(lmfit.oly)

par(mfrow = c(2, 2))
plot(lmfit.oly)
knitr::kable(data.frame(Year = seq(2000, 2012, length.out = 4),
    Prediction = predict(lmfit.oly, newdata = data.frame(Year = seq(2000,
        2012, length.out = 4))), Actuals = c(43.84, 44, 43.75,
        43.94)), caption = "Prediction of run times")

data("plastics")
plot(plastics)
DecomposeMulti <- decompose(plastics, type = "multiplicative")
plot(DecomposeMulti)
SeasonallyAdj <- seasadj(DecomposeMulti)
TrendComponent <- rwf(SeasonallyAdj, drift = T, h = 24)
plot(rwf(SeasonallyAdj, drift = T, h = 24))
SeasonalComponent <- snaive(DecomposeMulti$seasonal, h = 24)
randomComponent <- naive(DecomposeMulti$random, h = 24)
Fcts <- SeasonalComponent$mean * TrendComponent$mean * rep(mean(DecomposeMulti$random,
    na.rm = T), 24)
plot(c(plastics, Fcts), type = "b", col = c(rep("grey", 60),
    rep("blue", 24)), main = "Rationalizing the results of multiplicative decomposition",
    xlab = "Year", ylab = "Units in thousands")
legend("bottomright", legend = c("Actuals", "Forecasts"), col = c("grey",
    "blue"), pch = 1, cex = 0.6)

plastics.rev <- plastics
plastics.rev[59] <- plastics.rev[59] + 500
plot(decompose(plastics.rev, type = "multiplicative"))
plastics.rev2 <- plastics
plastics.rev2[42] <- plastics.rev[42] + 500
plot(decompose(plastics.rev2, type = "multiplicative"))
```