Ratnam_Project_2.R

dubey

Thu Apr 06 15:46:25 2017

```
#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.

#Question 1 ) Monthly total of people on unemployed benefits in Australia
(January 1956-July 1992).


#Loading the library
library(fma)

## Warning: package 'fma' was built under R version 3.3.3

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.3.3

#Loading the Package
dole

##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 1956   4742   6128   6494   5379   6011   7003   9164  10333   9614   9545
## 1957  15711  13135  13077  15453  15995  18071  20291  20175  18975  17928
## 1958  29856  26879  24485  27745  27282  29418  29908  29278  26002  23826
## 1959  31486  28207  27669  27559  27924  27528  27410  24887  21904  19598
## 1960  23781  20020  18177  17732  16765  16310  14897  12940  11465  10364
## 1961  19257  20941  29718  35025  45110  57154  61499  62090  59561  48531
## 1962  56755  49740  45870  49136  47256  46324  45453  42333  36851  33952
## 1963  46178  40482  36394  37142  36424  38188  37174  31869  26575  21758
## 1964  28649  24226  21955  19937  18287  18129  17072  14924  12491  11160
## 1965  15831  13698  12111  12690  12585  12855  12137  10977   9993   9614
## 1966  19490  17611  16206  17560  18082  19482  19200  18918  17375  16122
## 1967  24911  21969  21956  20944  22200  24002  22951  20143  17187  15287
## 1968  26943  23735  20744  21090  21502  21275  19426  16798  14209  13357
## 1969  23460  19551  15898  16012  16054  15910  13873  11854  10138   9942
## 1970  17778  13854  12681  11328  11946  13043  12785  11937  11383  10282
## 1971  18337  16779  15504  17258  18264  19184  19453  18741  19087  18171
## 1972  37486  37303  37639  36536  35850  41581  42979  42490  37992  32454
## 1973  48622  39868  34511  37234  36675  37945  36593  31669  28682  25944
## 1974  46847  38315  32600  33349  30598  32009  37599  45999  54945  68394
## 1975 182260 184177 157547 168471 159020 160748 169631 170927 179898 176471
## 1976 248619 215342 192024 178765 182397 188423 197159 198648 195864 194125
```

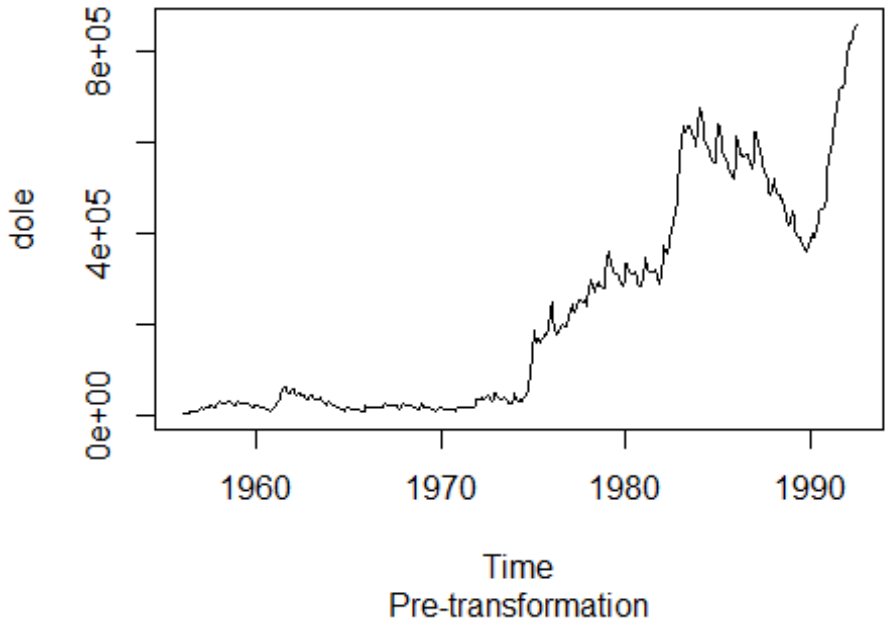
```
## 1977 229415 245395 236383 226807 239984 250309 253809 254863 249551 254085
## 1978 269896 298455 290356 283308 272384 286091 290718 285424 284642 279874
## 1979 341877 357463 334400 332572 318905 311232 310000 303800 299566 286241
## 1980 334495 334265 316776 309300 308989 311232 313943 303555 290386 283822
## 1981 339700 347400 325500 315200 314900 314500 313700 318500 306000 299500
## 1982 351425 372288 358536 356004 375626 390664 404840 421856 446341 465959
## 1983 601931 632837 622819 622162 633272 635002 634020 622103 610379 599100
## 1984 674424 667059 626653 602100 600344 584506 580347 570553 565348 555279
## 1985 636841 636342 599092 580700 568574 561400 553644 541022 534700 522587
## 1986 609987 603156 578700 568400 569966 569761 573989 573735 566245 556055
## 1987 623079 619978 592892 582102 561698 550850 536522 525650 515893 492248
## 1988 517127 511023 493993 483400 481469 475070 472806 458767 441201 428578
## 1989 448572 441100 409708 393323 391918 390001 383839 377968 368060 360246
## 1990 385727 398961 390149 391108 411171 427931 441335 450824 452304 457658
## 1991 567249 580777 596890 616326 647415 676706 701677 709801 718748 720754
## 1992 779868 816124 818102 826297 838390 851831 856505
##         Nov    Dec
## 1956  10096  13277
## 1957  19782  26055
## 1958  22302  27565
## 1959  19037  22469
## 1960  11738  17633
## 1961  47541  56756
## 1962  33392  43153
## 1963  20978  29555
## 1964  10658  15451
## 1965  10459  20509
## 1966  17269  26261
## 1967  17394  26321
## 1968  14766  25092
## 1969  11262  19601
## 1970  11316  19652
## 1971  20825  36441
## 1972  34049  47598
## 1973  28443  40232
## 1974  85735 140772
## 1975 184528 231311
## 1976 192866 215655
## 1977 238332 256223
## 1978 278829 311279
## 1979 282902 297782
## 1980 284842 299500
## 1981 289800 304700
## 1982 493446 562592
## 1983 591442 629214
## 1984 552893 576900
## 1985 520756 552300
## 1986 540527 560166
## 1987 481236 495772
## 1988 419052 420900
```

```
## 1989 357443 374530
## 1990 480083 523798
## 1991 730105 751348
## 1992
```

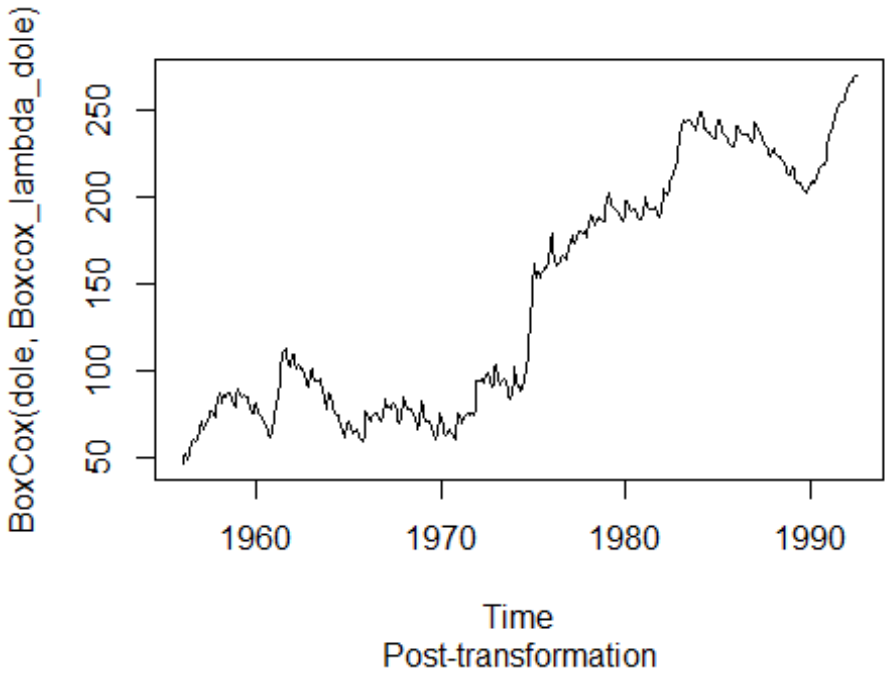
```
#Now taking the Summary of the Dole as to check the mean median and other IQR
summary(dole)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4742   20400   56760  221200  392600  856500
```

```
#plotting Dole
plot(dole , sub="Pre-transformation")
```



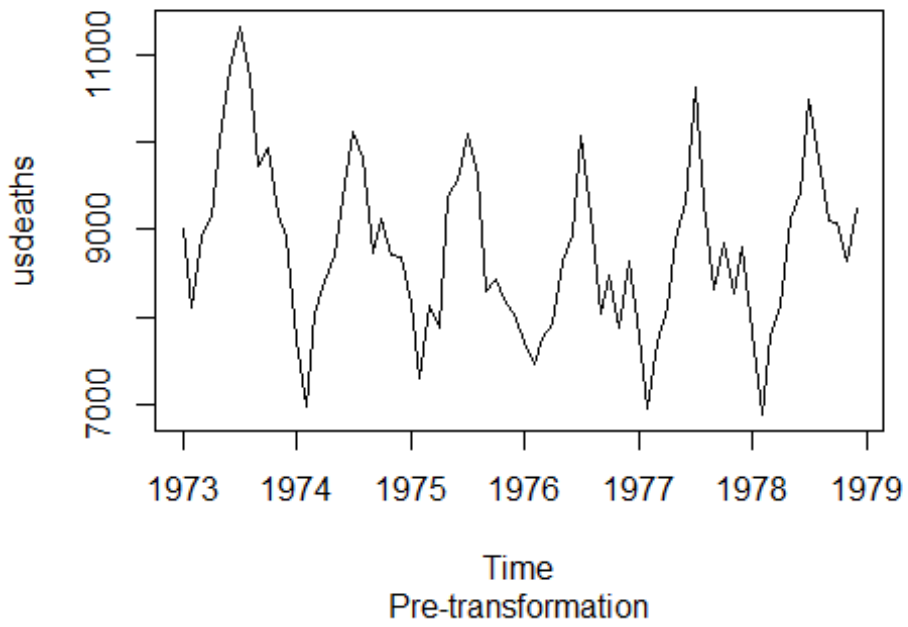

```
#Checking the Lambda for the DOle with the help of Box Cox
Boxcox_lambda_dole <- BoxCox.lambda(dole)
#based on the Lambda we will try to plot the Boxcox transformed data
plot(BoxCox(dole,Boxcox_lambda_dole),sub="Post-transformation")
```



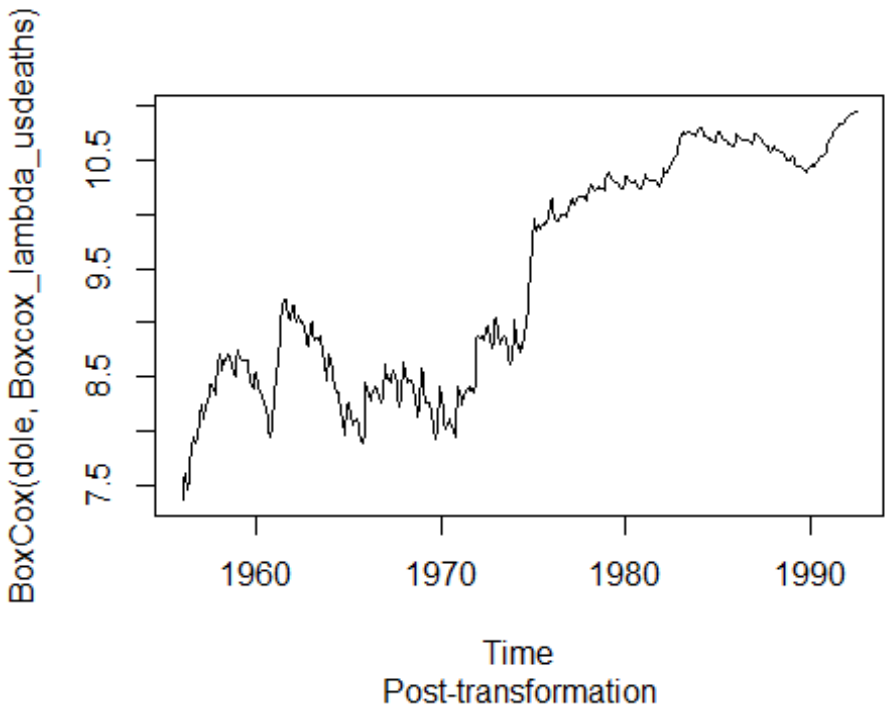

```
#Question 2 ) Monthly total of accidental deaths in the United States
(January 1973-December 1978).

#Loading the Package
usdeaths
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 1973  9007  8106  8928  9137 10017 10826 11317 10744  9713  9938  9161
## 1974  7750  6981  8038  8422  8714  9512 10120  9823  8743  9129  8710
## 1975  8162  7306  8124  7870  9387  9556 10093  9620  8285  8433  8160
## 1976  7717  7461  7776  7925  8634  8945 10078  9179  8037  8488  7874
## 1977  7792  6957  7726  8106  8890  9299 10625  9302  8314  8850  8265
## 1978  7836  6892  7791  8129  9115  9434 10484  9827  9110  9070  8633
##        Dec
## 1973  8927
## 1974  8680
## 1975  8034
## 1976  8647
## 1977  8796
## 1978  9240
```

```
#plotting the US deaths
plot(usdeaths , sub="Pre-transformation")
```



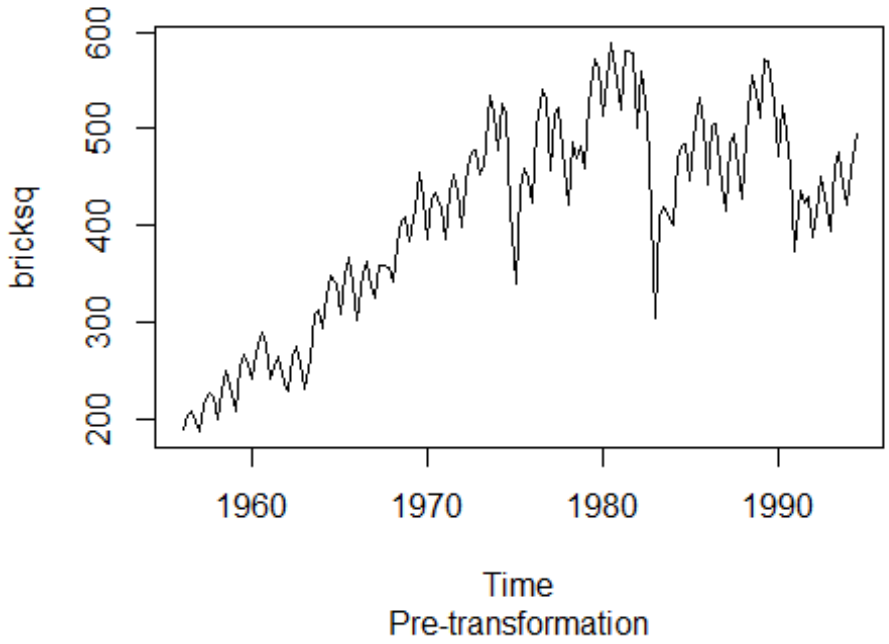

```
#Taking the Summary of the US Deaths
summary(usdeaths)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6892    8089    8728    8788    9323   11320

#Checking the Lambda for the us deaths with the help of Box Cox
Boxcox_lambda_usdeaths <- BoxCox.lambda(usdeaths)
#based on the Lambda we will try to plot the Boxcox transformed data
plot(BoxCox(dole,Boxcox_lambda_usdeaths),sub="Post-transformation")
```




```
#Question 3 ) Quarterly production of bricks (in millions of units) at
Portland, Australia (March 1956-September 1994).

#Loading the Package
bricksq

##      Qtr1 Qtr2 Qtr3 Qtr4
## 1956  189  204  208  197
## 1957  187  214  227  223
## 1958  199  229  249  234
## 1959  208  253  267  255
## 1960  242  268  290  277
## 1961  241  253  265  236
## 1962  229  265  275  258
## 1963  231  263  308  313
## 1964  293  328  349  340
## 1965  309  349  366  340
## 1966  302  350  362  337
## 1967  326  358  359  357
## 1968  341  380  404  409
## 1969  383  417  454  428
## 1970  386  428  434  417
## 1971  385  433  453  436
## 1972  399  461  476  477
## 1973  452  461  534  516
## 1974  478  526  518  417
## 1975  340  437  459  449
```

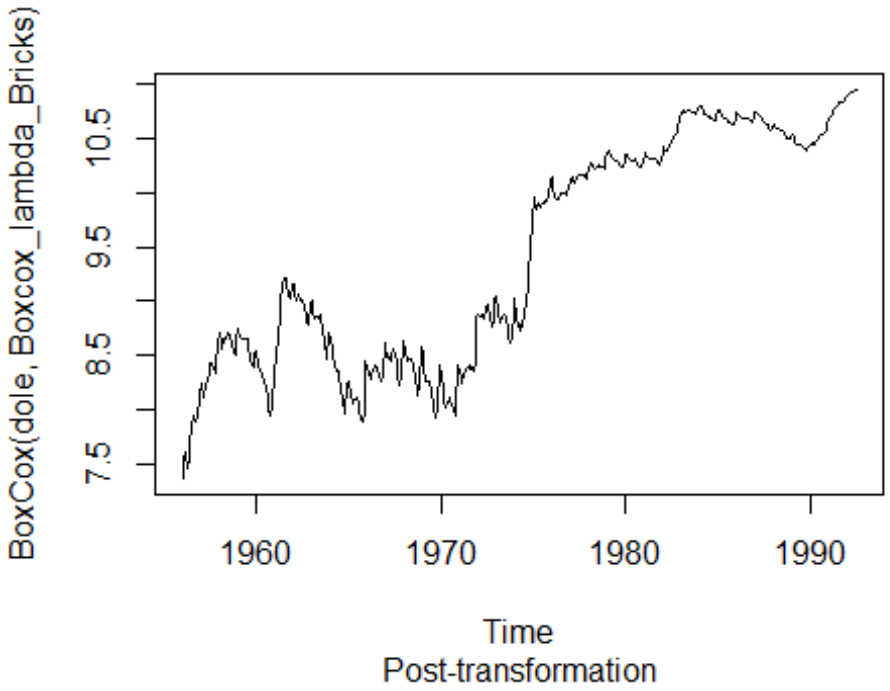
```
## 1976  424  501  540  533
## 1977  457  513  522  478
## 1978  421  487  470  482
## 1979  458  526  573  563
## 1980  513  551  589  564
## 1981  519  581  581  578
## 1982  500  560  512  412
## 1983  303  409  420  413
## 1984  400  469  482  484
## 1985  447  507  533  503
## 1986  443  503  505  443
## 1987  415  485  495  458
## 1988  427  519  555  539
## 1989  511  572  570  526
## 1990  472  524  497  460
## 1991  373  436  424  430
## 1992  387  413  451  420
## 1993  394  462  476  443
## 1994  421  472  494
```

```r
#plotting the Bricks
plot(bricksq ,sub="Pre-transformation")
```

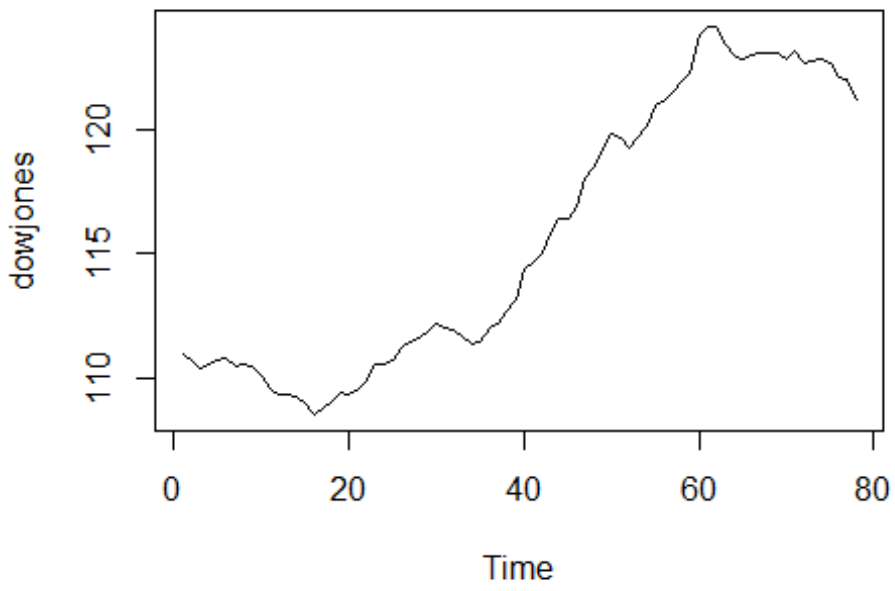



```r
#Taking the Summary of the Bricks
summary(bricksq)
```

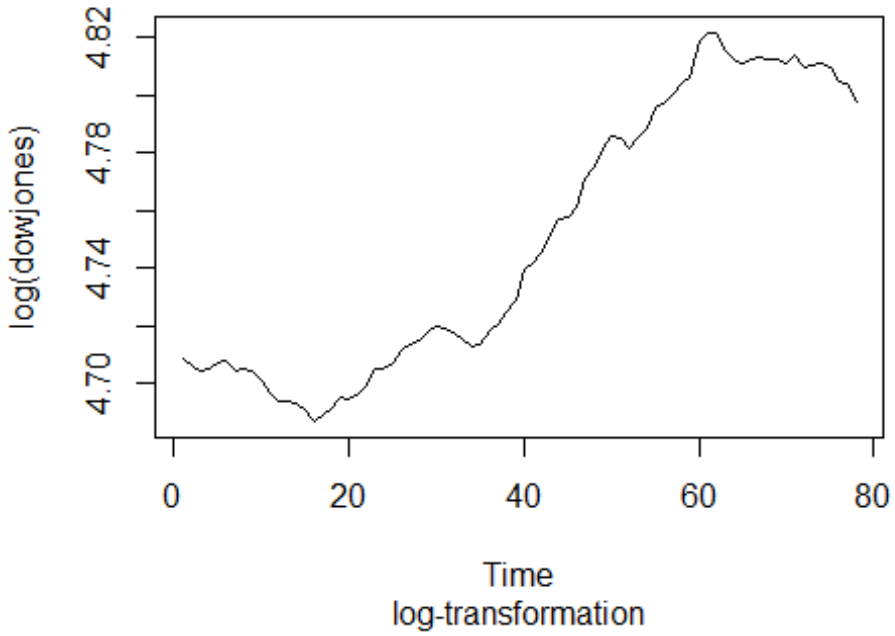
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   187.0   338.5   428.0   408.8   490.5   589.0

#Checking the Lambda for the Bricks with the help of Box Cox
Boxcox_lambda_Bricks <- BoxCox.lambda(usdeaths)
#based on the Lambda we will try to plot the Boxcox transformed data
plot(BoxCox(dole,Boxcox_lambda_Bricks) , sub="Post-transformation")
```



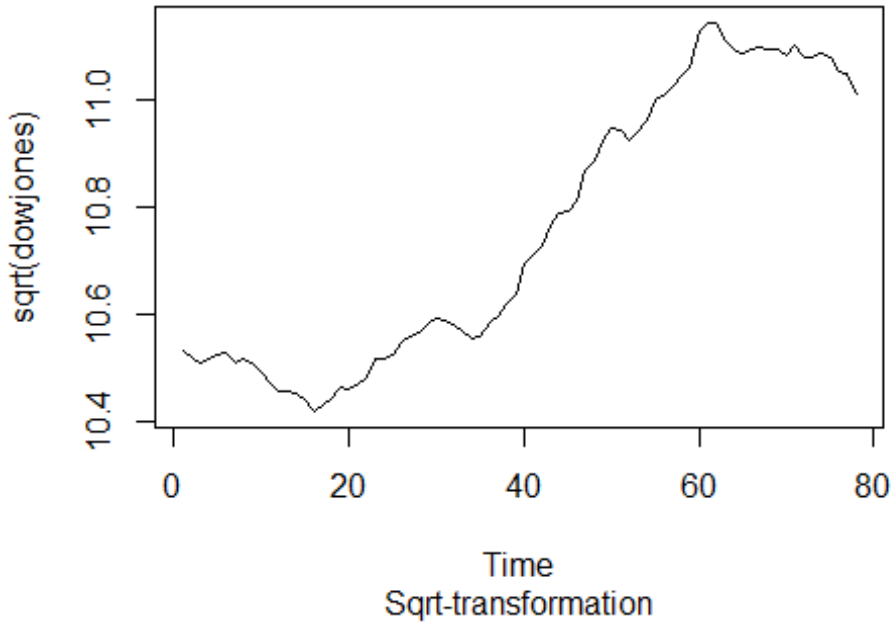

```
#############################################################################
####################

#Use the Dow Jones index (data set dowjones) to do the following:

# Question 1 ) Produce a time plot of the series.

#plotting the Dow jones
plot(dowjones)
```



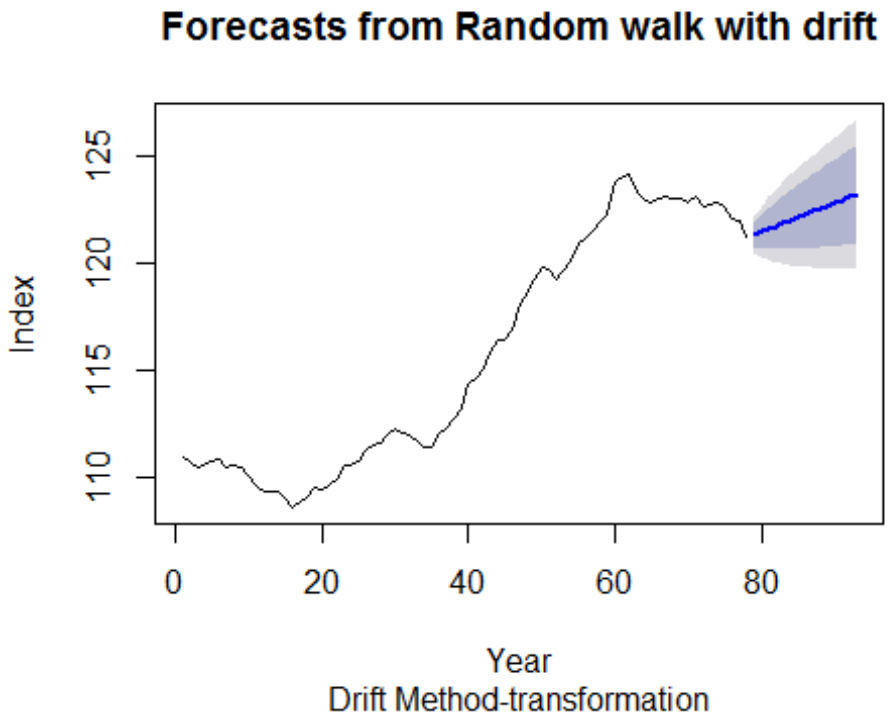

```r
#ploting the log of the DOw jones
plot(log(dowjones) , sub="log-transformation")
```



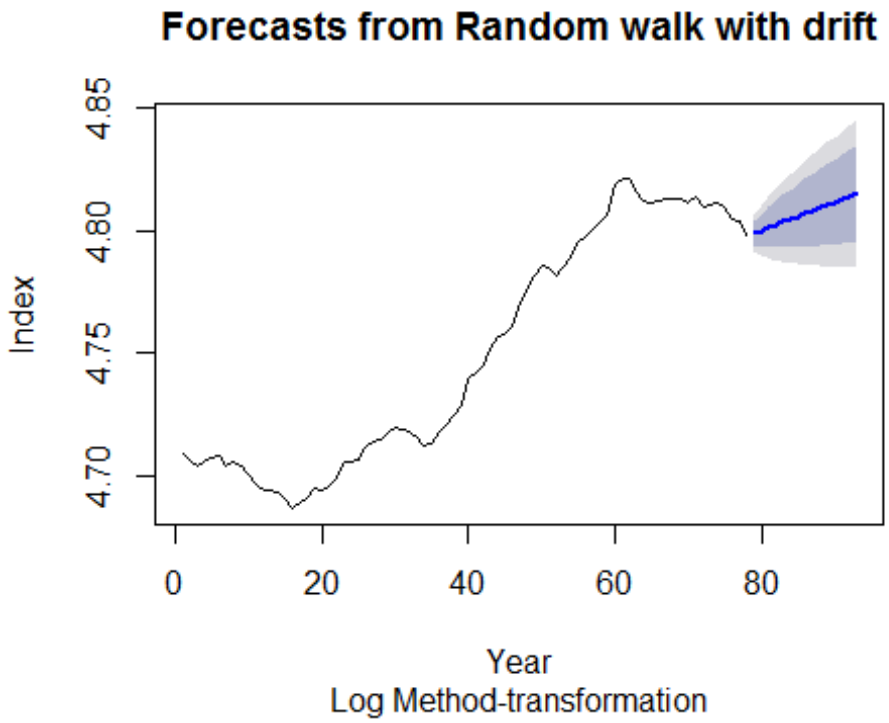
```r
#ploting the log of the Square root
plot(sqrt(dowjones) , sub="Sqrt-transformation")
```



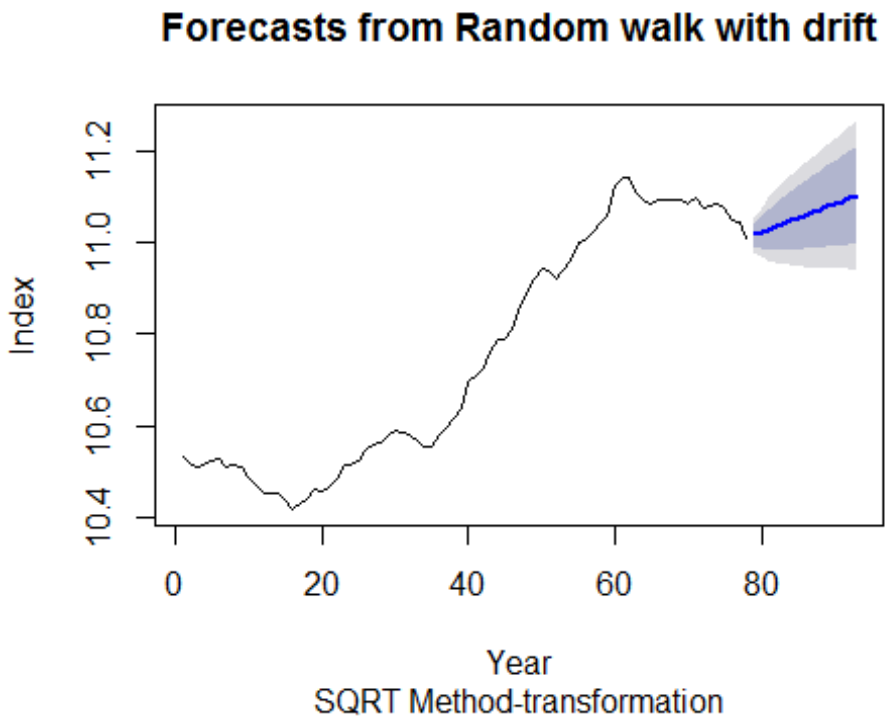

```r
#Question 2 ) Produce forecasts using the drift method and plot them

#Generating Random Walk Forecasts using drift menthod
drift_dowjones <- rwf(dowjones , h=15, drift=TRUE)
#Generating Random Walk Forecasts log using drift menthod
drift_dowjones_log <- rwf(log(dowjones), h=15, drift=TRUE)
#Generating Random Walk Forecasts sqrt using drift menthod
drift_dowjones_sqrt <- rwf(sqrt(dowjones), h=15, drift=TRUE)

#Plotting the Data using Drift menthod
plot(drift_dowjones , sub="Drift Method-transformation"
,ylab="Index",xlab="Year")
```



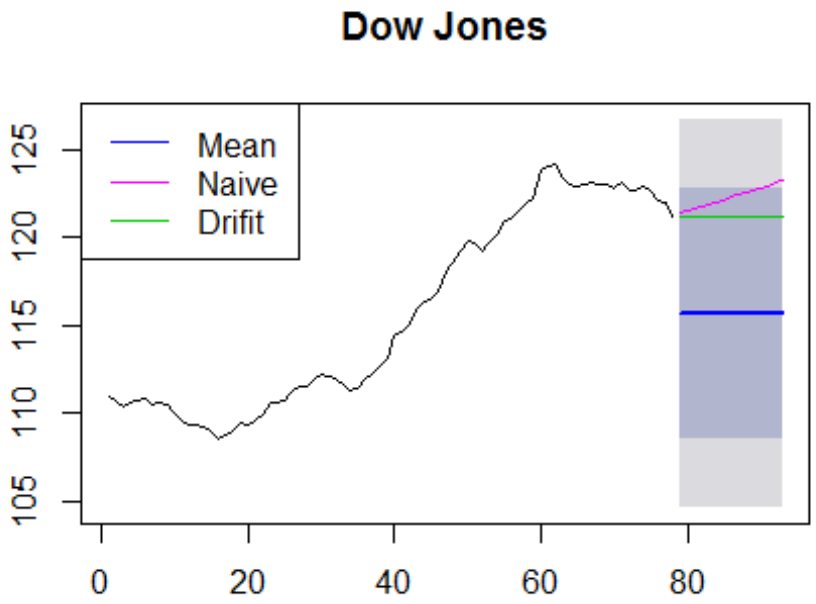

```r
#Plotting the Data using Log menthod
plot(drift_dowjones_log , sub="Log Method-transformation"
,ylab="Index",xlab="Year")
```



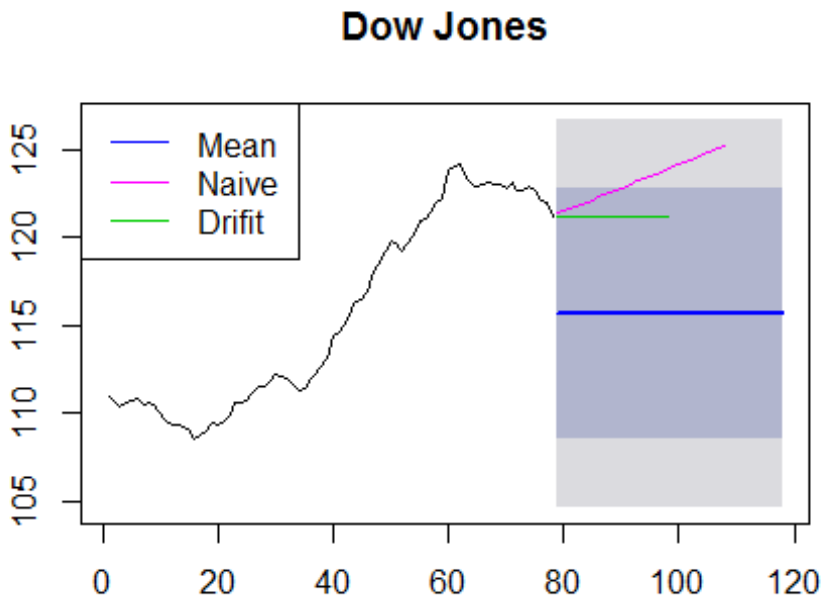
```r
#Plotting the Data using SQRT menthod
plot(drift_dowjones_sqrt , sub="SQRT Method-transformation"
,ylab="Index",xlab="Year")
```




```r
#Question 2 ) Show that the graphed forecasts are identical to extending the
line drawn between the first and last observations.
#for the First and Last observation we need to do the Mean , Naive and Drift
method in the Plot
dowjones_win <- window(dowjones)
dowjones_mean <- meanf(dowjones_win,h=15)
#Generating Random Walk Forecasts with Drift = TRUE
dowjones_drift <- rwf(dowjones_win,h=15, drift = TRUE)
#Generating Random Walk Forecasts without Drift
dowjones_no_drift <- rwf(dowjones_win,h=15)

# Generating the Plot for the same
plot(dowjones_mean, main="Dow Jones")
lines(dowjones_drift$mean,col=6)
lines(dowjones_no_drift$mean,col=3)
lines(dowjones)
legend("topleft", lty=1, col=c(4,6,3), legend=c("Mean ","Naive","Drifit"))
```




```r
#Question 3 )  Try some of the other benchmark functions to forecast the same data set.
#Which do you think is best? Why?

dowjones_nw_win <- window(dowjones)
dowjones_nw_mean <- meanf(dowjones_nw_win,h=40)
#Generating Random Walk Forecasts with Drift = TRUE
dowjones_nw_drift <- rwf(dowjones_nw_win,h=30, drift = TRUE)
#Generating Random Walk Forecasts without Drift
dowjones_no_nw_drift <- rwf(dowjones_nw_win,h=20)

# Generating the Plot for the same with modified h
plot(dowjones_nw_mean, main="Dow Jones")
lines(dowjones_nw_drift$mean,col=6)
lines(dowjones_no_nw_drift$mean,col=3)
lines(dowjones)
legend("topleft", lty=1, col=c(4,6,3), legend=c("Mean ","Naive","Drifit"))
```




```
##############################################################################
#########################
```

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

```
#Question 1 ) Produce some plots of the data in order to become familiar with
it.
```

```
#Loading the Package
ibmclose
```

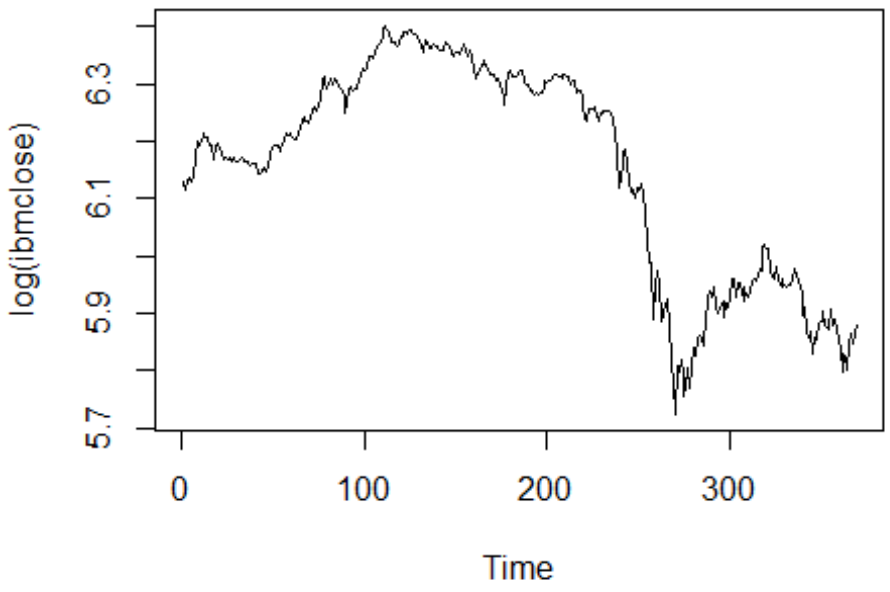
```
## Time Series:
## Start = 1
## End = 369
## Frequency = 1
##   [1] 460 457 452 459 462 459 463 479 493 490 492 498 499 497 496 490 489
##  [18] 478 487 491 487 482 479 478 479 477 479 475 479 476 476 478 479 477
##  [35] 476 475 475 473 474 474 474 465 466 467 471 471 467 473 481 488 490
##  [52] 489 489 485 491 492 494 499 498 500 497 494 495 500 504 513 511 514
##  [69] 510 509 515 519 523 519 523 531 547 551 547 541 545 549 545 549 547
##  [86] 543 540 539 532 517 527 540 542 538 541 541 547 553 559 557 557 560
## [103] 571 571 569 575 580 584 585 590 599 603 599 596 585 587 585 581 583
## [120] 592 592 596 596 595 598 598 595 595 592 588 582 576 578 589 585 580
## [137] 579 584 581 581 577 577 578 580 586 583 581 576 571 575 575 573 577
## [154] 582 584 579 572 577 571 560 549 556 557 563 564 567 561 559 553 553
```

```
## [171] 553 547 550 544 541 532 525 542 555 558 551 551 552 553 557 557 548
## [188] 547 545 545 539 539 535 537 535 536 537 543 548 546 547 548 549 553
## [205] 553 552 551 550 553 554 551 551 545 547 547 537 539 538 533 525 513
## [222] 510 521 521 521 523 516 511 518 517 520 519 519 519 518 513 499 485
## [239] 454 462 473 482 486 475 459 451 453 446 455 452 457 449 450 435 415
## [256] 398 399 361 383 393 385 360 364 365 370 374 359 335 323 306 333 330
## [273] 336 328 316 320 332 320 333 344 339 350 351 350 345 350 359 375 379
## [290] 376 382 370 365 367 372 373 363 371 369 376 387 387 376 385 385 380
## [307] 373 382 377 376 379 386 387 386 389 394 393 409 411 409 408 393 391
## [324] 388 396 387 383 388 382 384 382 383 383 388 395 392 386 383 377 364
## [341] 369 355 350 353 340 350 349 358 360 360 366 359 356 355 367 357 361
## [358] 355 348 343 330 340 339 331 345 352 346 352 357
```

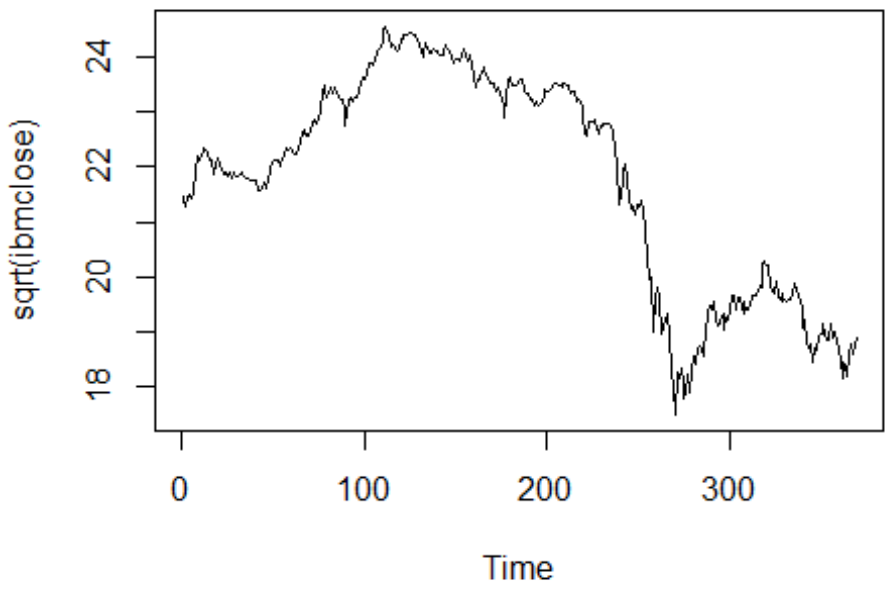
```
#Now taking the Summary of the ibm as to check the mean median and other IQR
summary(ibmclose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   306.0   387.0   494.0   478.5   549.0   603.0
```

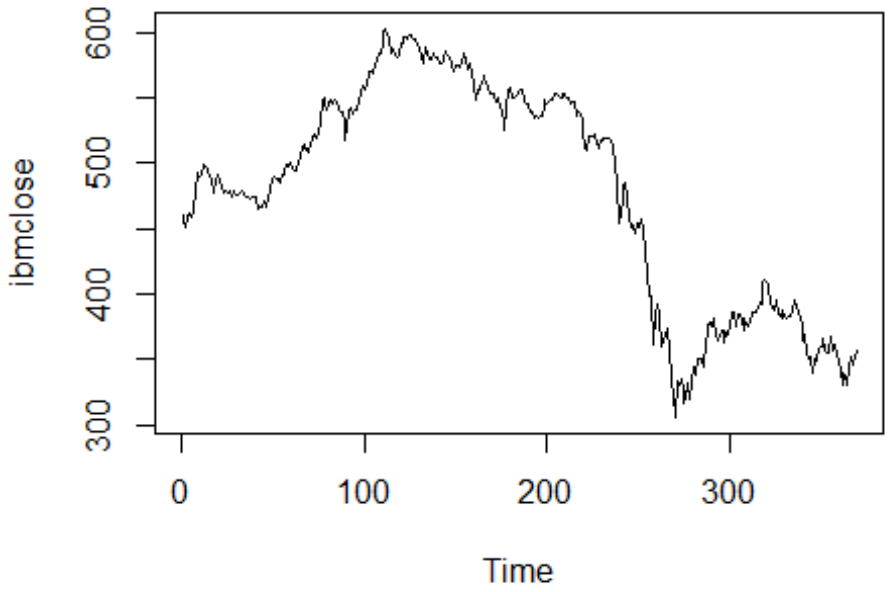
```
#Producing some plots
# Generating the Plot Log
plot(log(ibmclose))
```



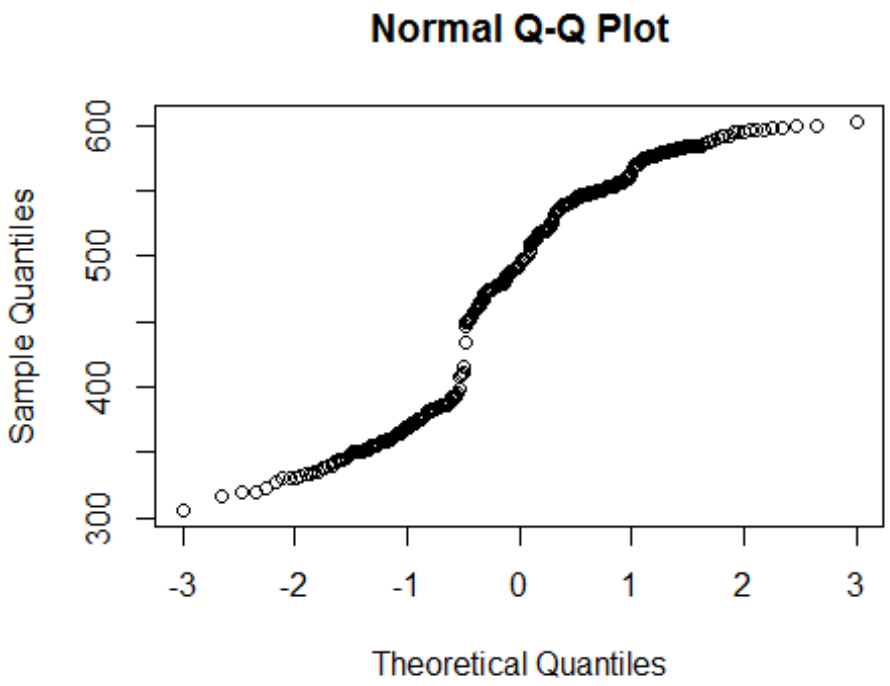

```
# Generating the Plot SQRT
plot(sqrt(ibmclose))
```




```r
# Generating the Plot Simple
plot(ibmclose)
```

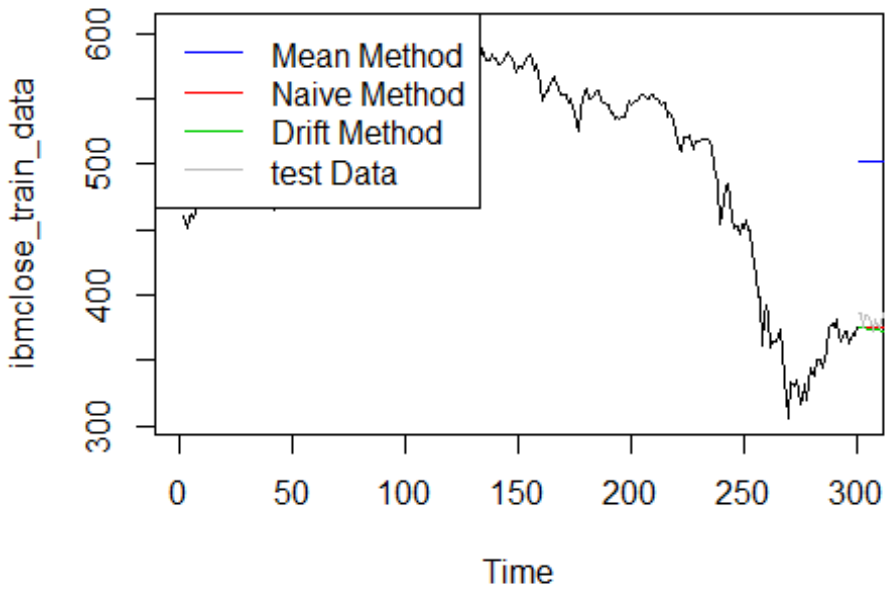


```r
## Generating the Plot QQ NORM  PLOT
qqnorm(ibmclose)
```




```r
#Question 2 ) Split the data into a training set of 300 observations and a
test set of 69 observations.

ibmclose_train_data <- window(ibmclose ,Start= 1 ,end=300)
ibmclose_test_data <- window(ibmclose ,start=301, end=369)

#Question 3 ) Try various benchmark methods to forecast the training set and
compare the
#results on the test set. Which method did best?


ibmclose_avg <- meanf(ibmclose_train_data,h=64)$mean
ibmclose_naive <- naive(ibmclose_train_data ,h=64)$mean
ibmclose_drift <- rwf(ibmclose_train_data ,drift=TRUE,h=64)$mean

#plotting the Data

plot(ibmclose_train_data)
lines(ibmclose_naive,col=2)
lines(ibmclose_avg,col=4)
lines(ibmclose_drift,col=3)
lines(ibmclose_test_data,col=8)
```

```
legend("topleft",lty=1,col=c(4,2,3,8), legend=c("Mean Method","Naive
Method","Drift Method","test Data"))
```




```
##############################################################################
#####################
```

#Consider the sales of new one-family houses in the USA, Jan 1973 - Nov 1995 (data set hsales).

#Question 1) Produce some plots of the data in order to become familiar with it.

#Loading the Package

```
hsales
```

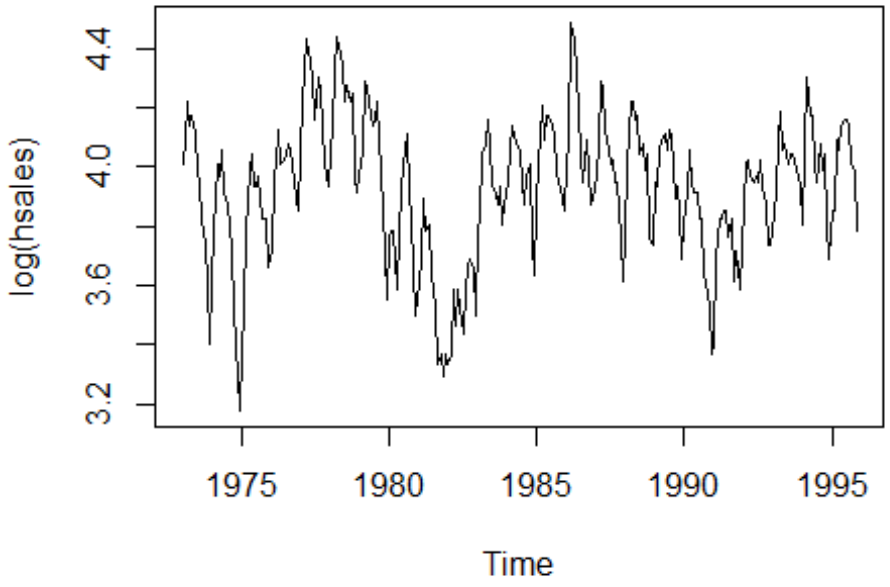
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1973  55  60  68  63  65  61  54  52  46  42  37  30
## 1974  37  44  55  53  58  50  48  45  41  34  30  24
## 1975  29  34  44  54  57  51  51  53  46  46  46  39
## 1976  41  53  55  62  55  56  57  59  58  55  49  47
## 1977  57  68  84  81  78  74  64  74  71  63  55  51
## 1978  57  63  75  85  80  77  68  72  68  70  53  50
## 1979  53  58  73  72  68  63  64  68  60  54  41  35
```

```
## 1980  43  44  44  36  44  50  55  61  50  46  39  33
## 1981  37  40  49  44  45  38  36  34  28  29  27  29
## 1982  28  29  36  32  36  34  31  36  39  40  39  33
## 1983  44  46  57  59  64  59  51  50  48  51  45  48
## 1984  52  58  63  61  59  58  52  48  53  55  42  38
## 1985  48  55  67  60  65  65  63  61  54  52  51  47
## 1986  55  59  89  84  75  66  57  52  60  54  48  49
## 1987  53  59  73  72  62  58  55  56  52  52  43  37
## 1988  43  55  68  68  64  65  57  59  54  57  43  42
## 1989  52  51  58  60  61  58  62  61  49  51  47  40
## 1990  45  50  58  52  50  50  46  46  38  37  34  29
## 1991  30  40  46  46  47  47  43  46  37  41  39  36
## 1992  48  55  56  53  52  53  52  56  51  48  42  42
## 1993  44  50  60  66  58  59  55  57  57  56  53  51
## 1994  45  58  74  65  65  55  52  59  54  57  45  40
## 1995  47  47  60  58  63  64  64  63  55  54  44
```

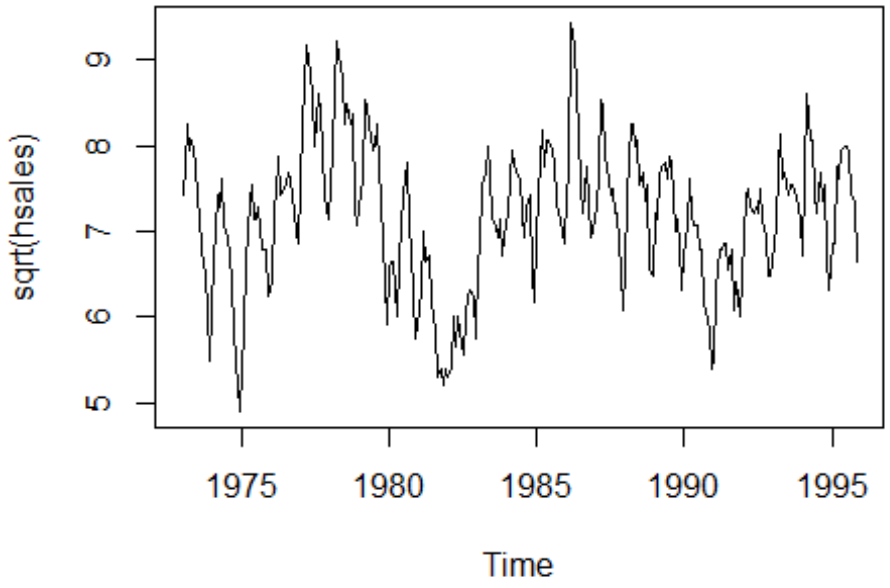
```
#Now taking the Summary of the ibm as to check the mean median and other IQR
summary(hsales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   44.00   53.00   52.29   59.00   89.00
```

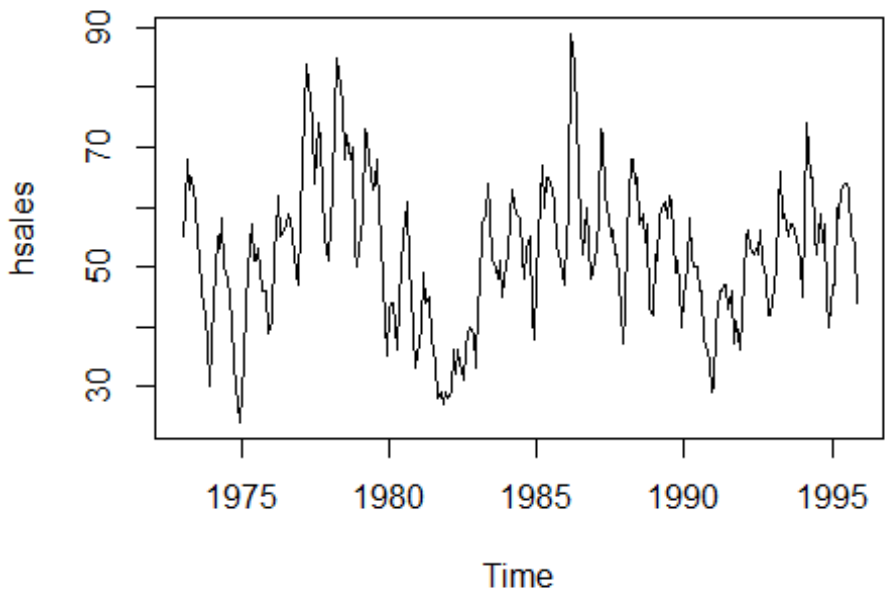
```
#Producing some plots
# Generating the Plot Log
plot(log(hsales))
# Generating the Plot Log
plot(log(hsales))
```



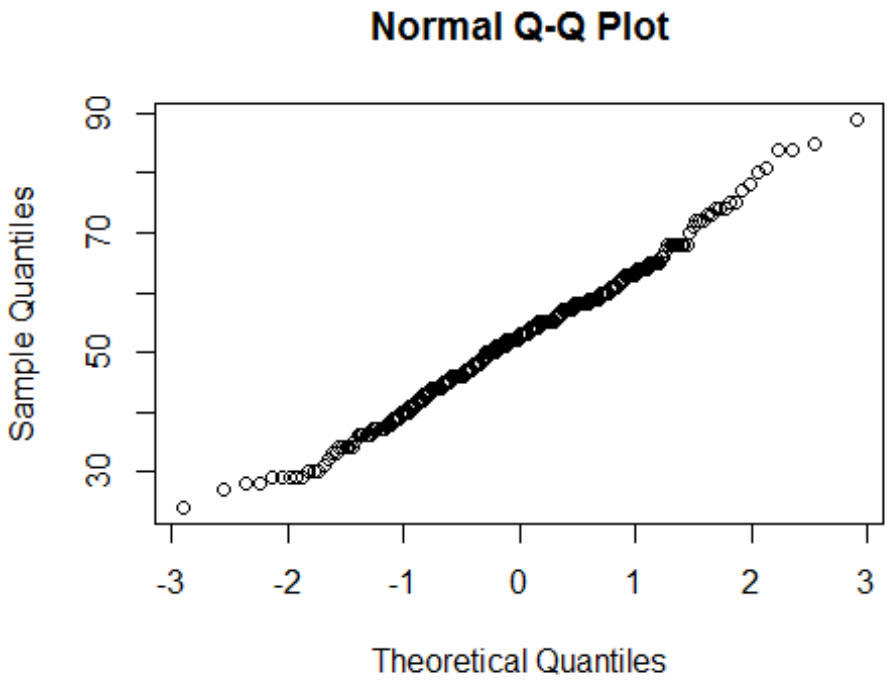

```r
# Generating the Plot SQRT
plot(sqrt(hsales))
```



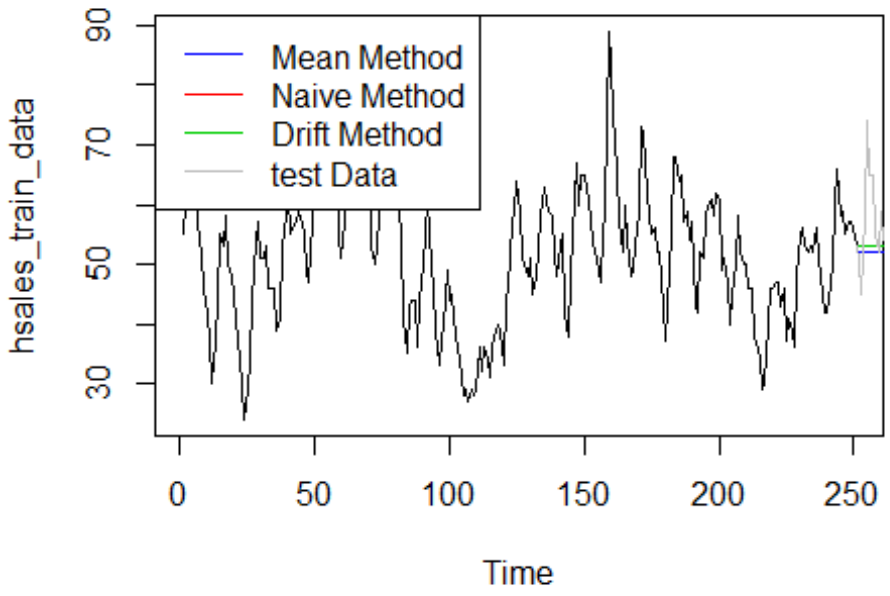
```r
# Generating the Plot Simple
plot(hsales)
```




```r
## Generating the Plot QQ NORM  PLOT
qqnorm(hsales)
```




```
#Question 2 ) Split the hsales data set into a training set and a test set,
where the test set is the last two years of data.
hsales_ts <- ts(hsales,start=1,end=275)
hsales_train_data <-  window(hsales_ts,start=1,end=251)
hsales_test_data <- window(hsales_ts,start=251)

#Question 3 ) Try various benchmark methods to forecast the training set and
compare the results on the test set. Which method did best?


hsales_avg <- meanf(hsales_train_data,h=34)$mean
hsales_naive <- naive(hsales_train_data ,h=34)$mean
hsales_drift <- rwf(hsales_train_data ,drift=TRUE,h=34)$mean

#plotting the Data

plot(hsales_train_data)
lines(hsales_naive,col=2)
lines(hsales_avg,col=4)
lines(hsales_drift,col=3)
lines(hsales_test_data,col=8)
legend("topleft",lty=1,col=c(4,2,3,8), legend=c("Mean Method","Naive
Method","Drift Method","test Data"))
```



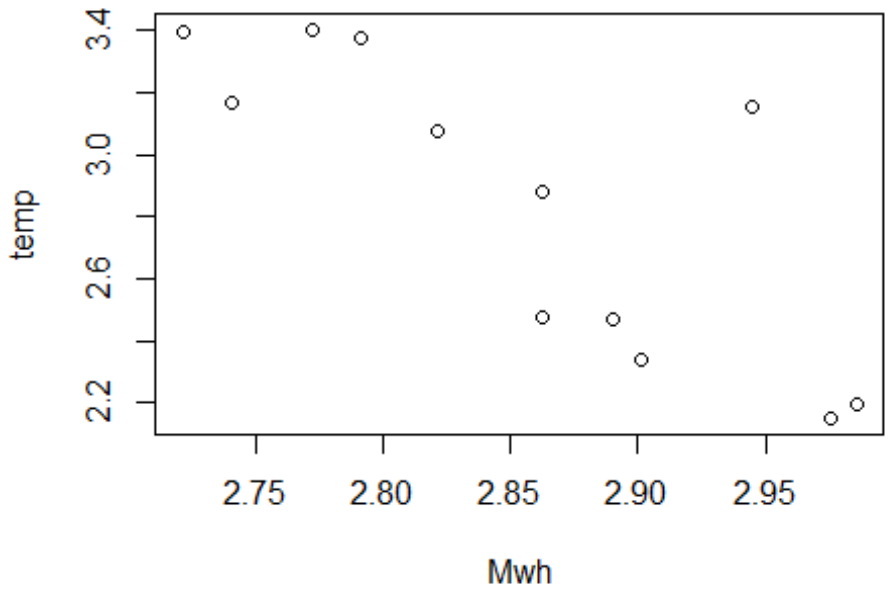

```
##############################################################################
#########################
#
##
#                              QUESTION 4
##
#
##
##############################################################################
#########################


#Day    1   2   3   4   5   6   7   8   9   10  11  12
#Mwh    16.3    16.8    15.5    18.2    15.2    17.5    19.8    19.0    17.5
16.0    19.6    18.0
#temp   29.3    21.7    23.7    10.4    29.7    11.9    9.0 23.4    17.8
30.0    8.6 11.8

#Question 1 ) Plot the data and find the regression model for Mwh with
temperature as an explanatory variable. Why is there a negative relationship?

#Loading the Package
econsumption

##     Mwh temp
## 1  16.3 29.3
```

```
## 2  16.8 21.7
## 3  15.5 23.7
## 4  18.2 10.4
## 5  15.2 29.7
## 6  17.5 11.9
## 7  19.8  9.0
## 8  19.0 23.4
## 9  17.5 17.8
## 10 16.0 30.0
## 11 19.6  8.6
## 12 18.0 11.8
```

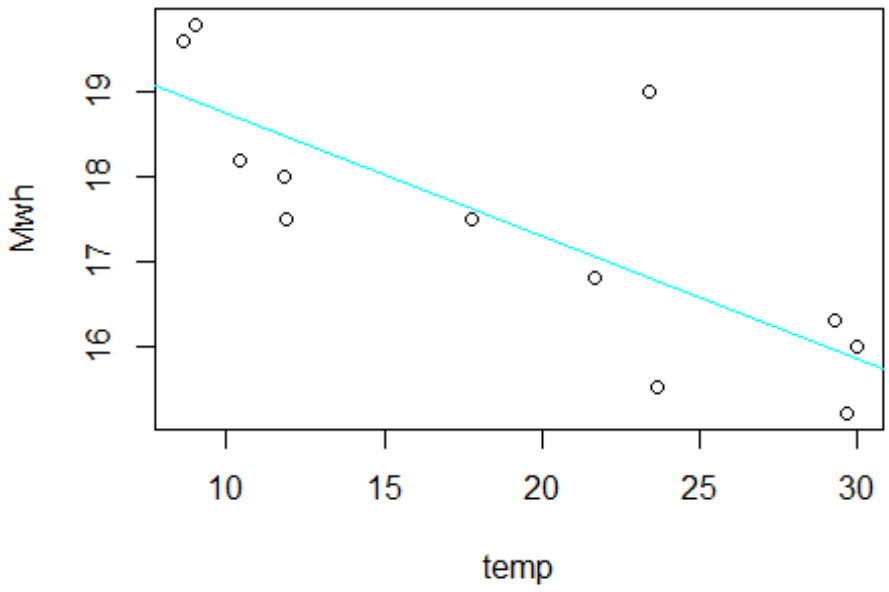
```r
#Now taking the Summary of the ibm as to check the mean median and other IQR
summary(econsumption)
```

```
##       Mwh             temp
##  Min.   :15.20   Min.   : 8.60
##  1st Qu.:16.23   1st Qu.:11.45
##  Median :17.50   Median :19.75
##  Mean   :17.45   Mean   :18.94
##  3rd Qu.:18.40   3rd Qu.:25.10
##  Max.   :19.80   Max.   :30.00
```

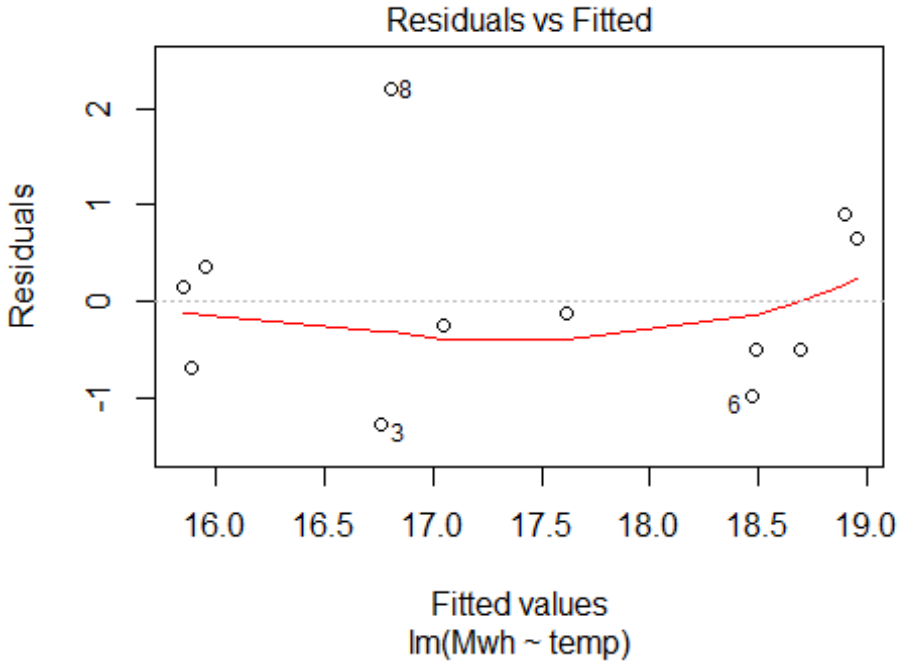
```r
#Producing some plots
# Generating the Plot Log
plot(log(econsumption))
```

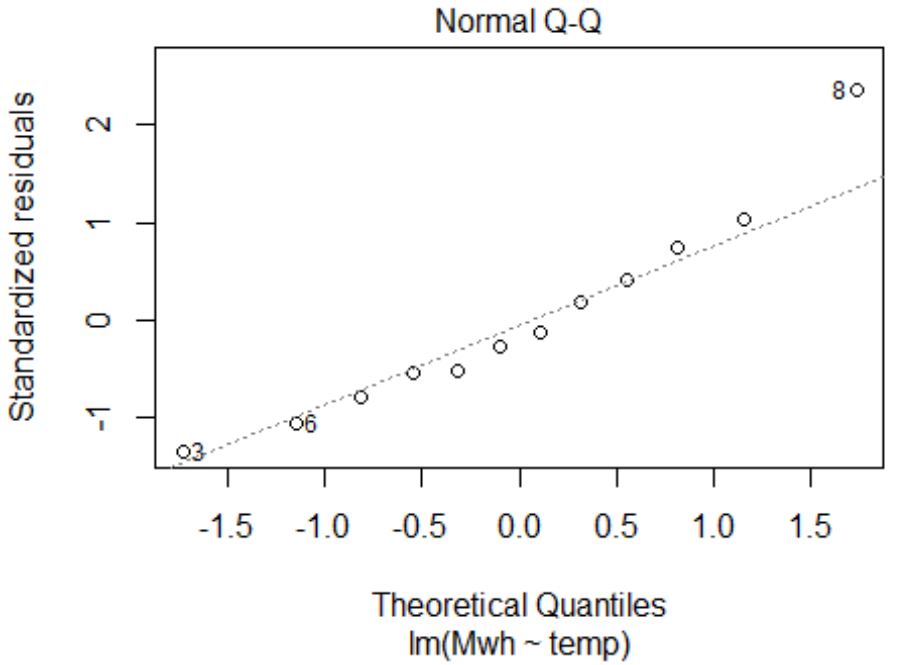


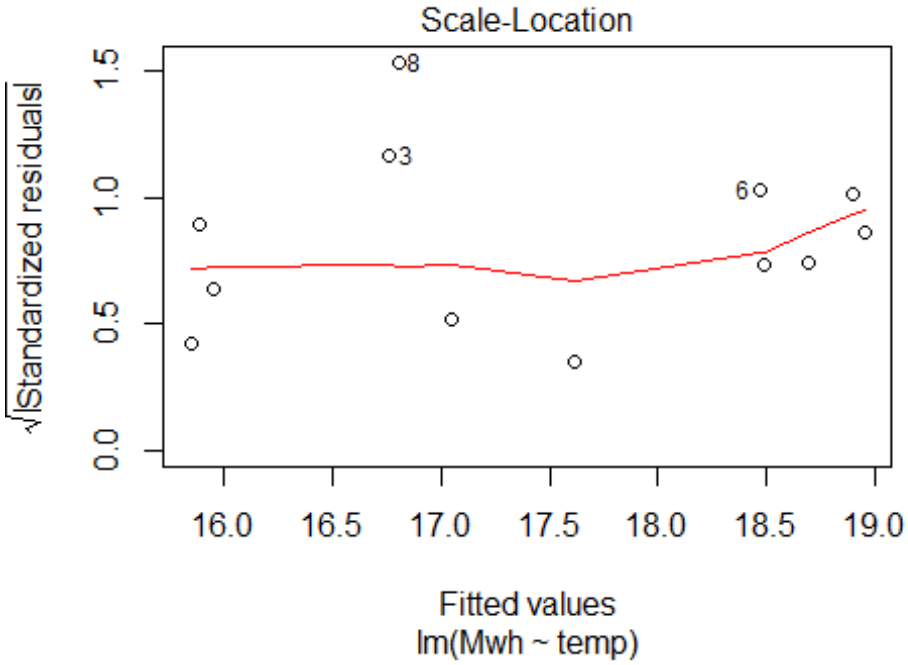
```r
#question 2) a. Plot the data and find the regression model for Mwh with
temperature as an explanatory variable. Why is there a negative relationship?
plot(Mwh ~ temp, data = econsumption)
fit_eco = lm(formula = Mwh  ~ temp, data = econsumption)
abline(fit_eco, col=5)
```

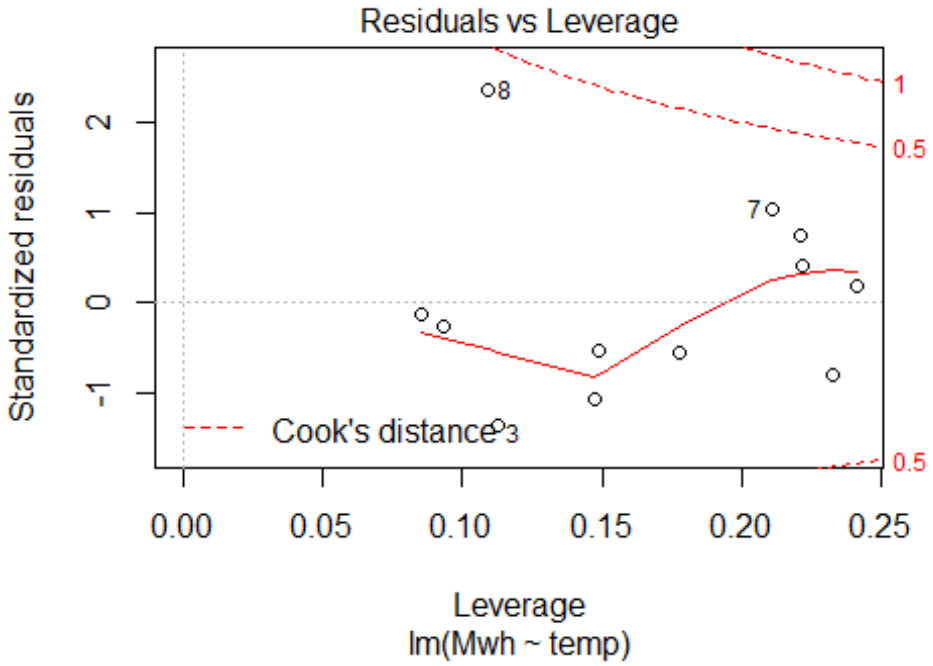



```r
#Question 3 ) Produce a residual plot. Is the model adequate? Are there any
outliers or influential observations?

plot(fit_eco)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Mwh ~ temp)
8
3
6
16.0
16.5
17.0
17.5
18.0
18.5
19.0
-1
0
1
2


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(Mwh ~ temp)
8
6
3
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
-1
0
1
2



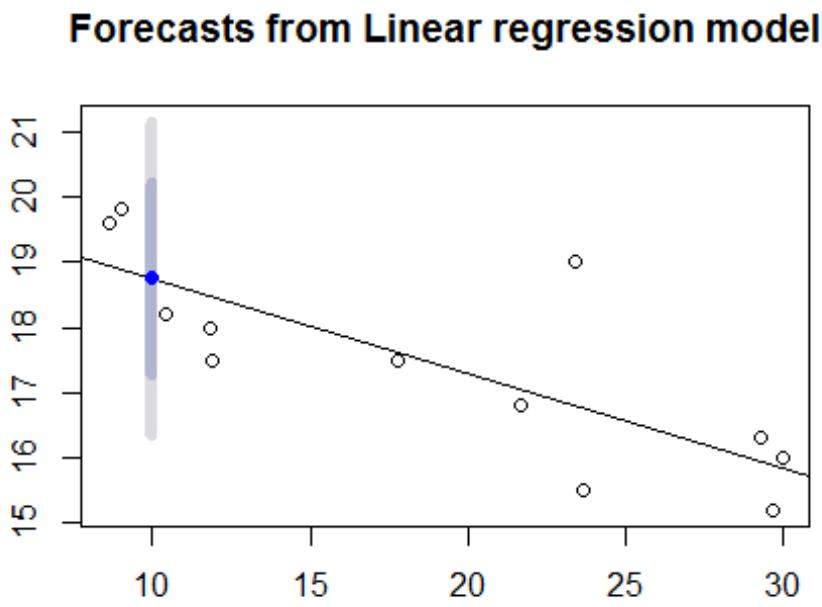



```
#Question 4) Use the model to predict the electricity consumption that you
would expect for a day with maximum temperature 10 and a day with maximum
temperature 35. Do you believe these predictions?
coeffs = coefficients(fit_eco)
```

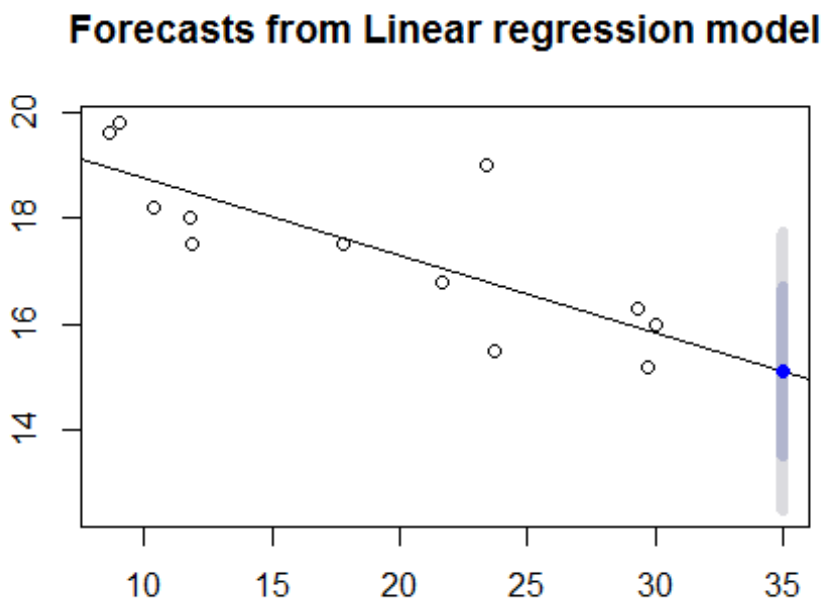
```r
pred_temp = c(10, 35)
p_temp = coeffs[1] + coeffs[2]*pred_temp
p_temp
```

```
## [1] 18.74795 15.11902
```

```r
#Question 5 Give prediction intervals for your forecasts. The following R
code will get you started:

forecast <- forecast(fit_eco , newdata=data.frame(temp=10))
plot(forecast)
```




```r
forecast2 <- forecast(fit_eco, newdata=data.frame(temp=35))
plot(forecast2)
```




```
temp10 = data.frame(temp=10)
temp35 = data.frame(temp=35)
predict(fit_eco, temp10, interval="predict")
```

```
##        fit      lwr      upr
## 1 18.74795 16.34824 21.14766
```

```
predict(fit_eco, temp35, interval="predict")
```

```
##        fit      lwr      upr
## 1 15.11902 12.49768 17.74035
```

```
################################################################################
##############################
```

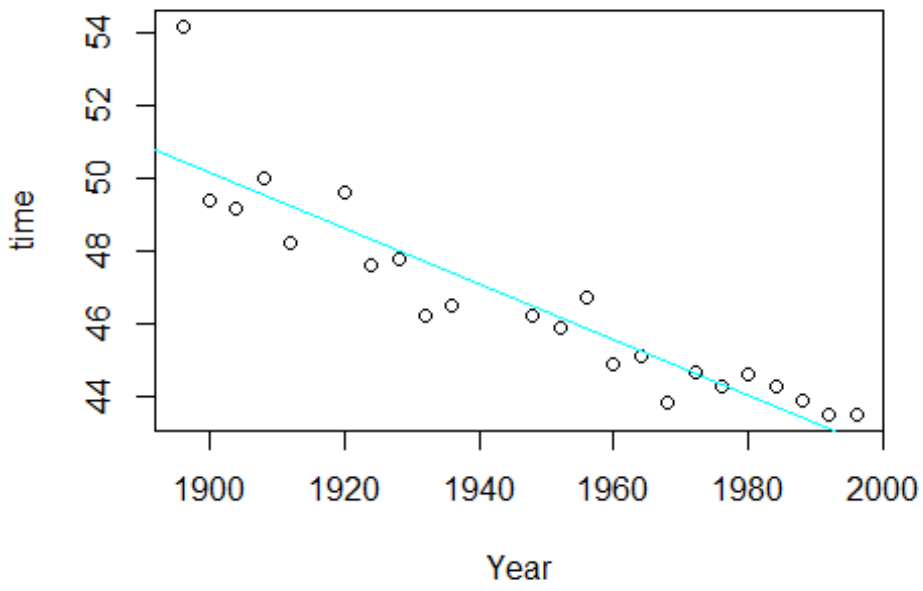
```
#The following table gives the winning times (in seconds) for the men's 400
meters final in each Olympic Games from 1896 to 2012 (data set `olympic`).

#question 1) Update the data set `olympic` to include the winning times from
the last few Olympics.
```

```
olympic
```

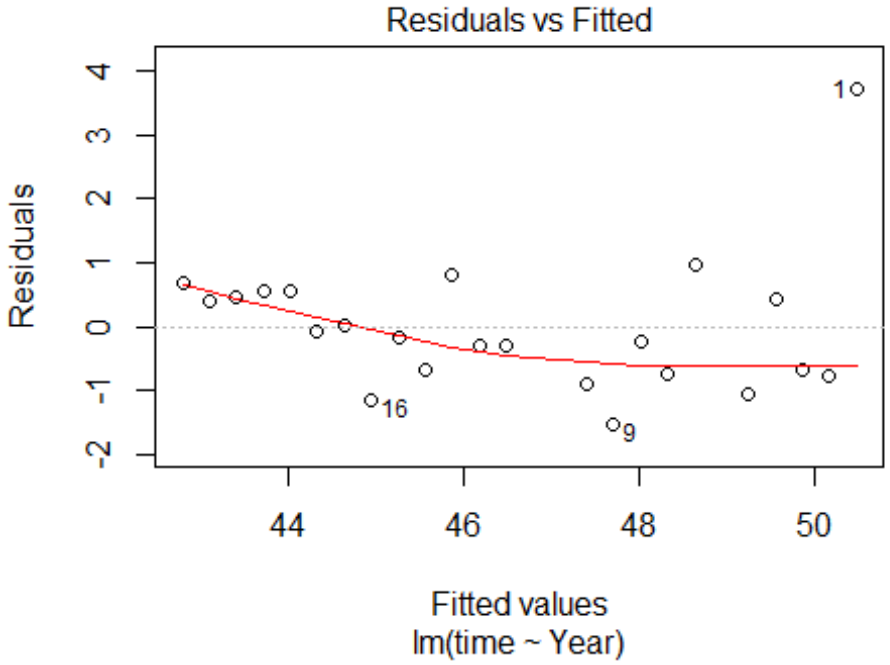
```
##    Year  time
## 1  1896 54.20
## 2  1900 49.40
## 3  1904 49.20
```

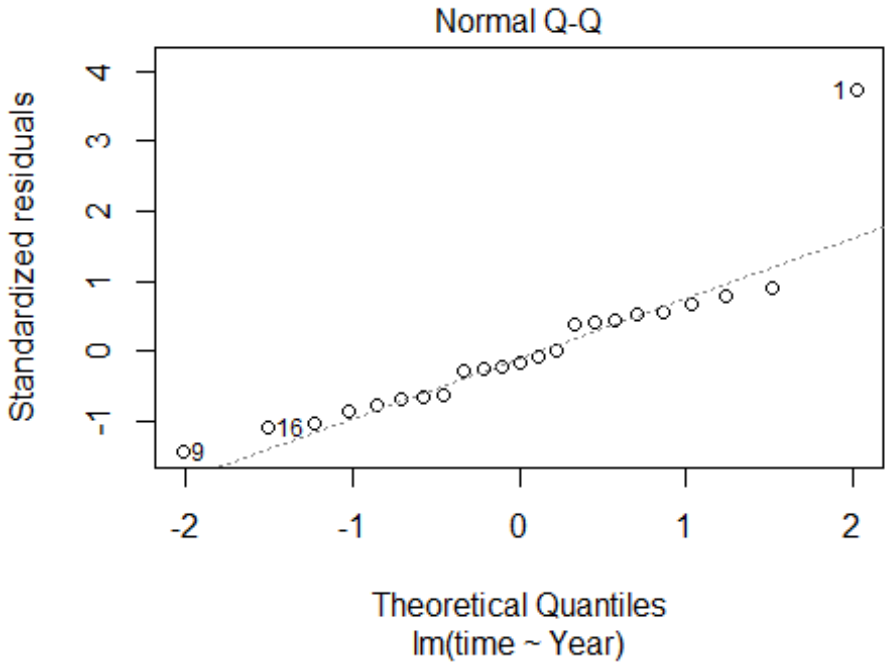
```
## 4  1908 50.00
## 5  1912 48.20
## 6  1920 49.60
## 7  1924 47.60
## 8  1928 47.80
## 9  1932 46.20
## 10 1936 46.50
## 11 1948 46.20
## 12 1952 45.90
## 13 1956 46.70
## 14 1960 44.90
## 15 1964 45.10
## 16 1968 43.80
## 17 1972 44.66
## 18 1976 44.26
## 19 1980 44.60
## 20 1984 44.27
## 21 1988 43.87
## 22 1992 43.50
## 23 1996 43.49
```

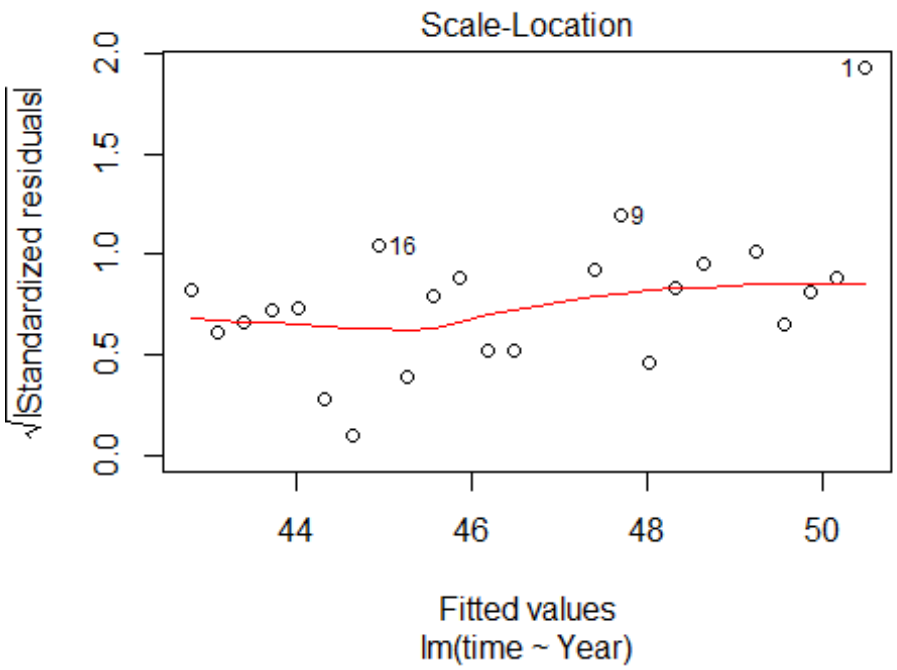
```r
#question 2) Plot the winning time against the year. Describe the main features of the scatterplot.

plot(time ~ Year, data = olympic)

#Question 3) Fit a regression line to the data. Obviously the winning times have been decreasing, but at what average rate per year?

oly_fit = lm(formula = time  ~ Year, data = olympic)
plot(time ~ Year, data = olympic)
abline(oly_fit, col=5)
```

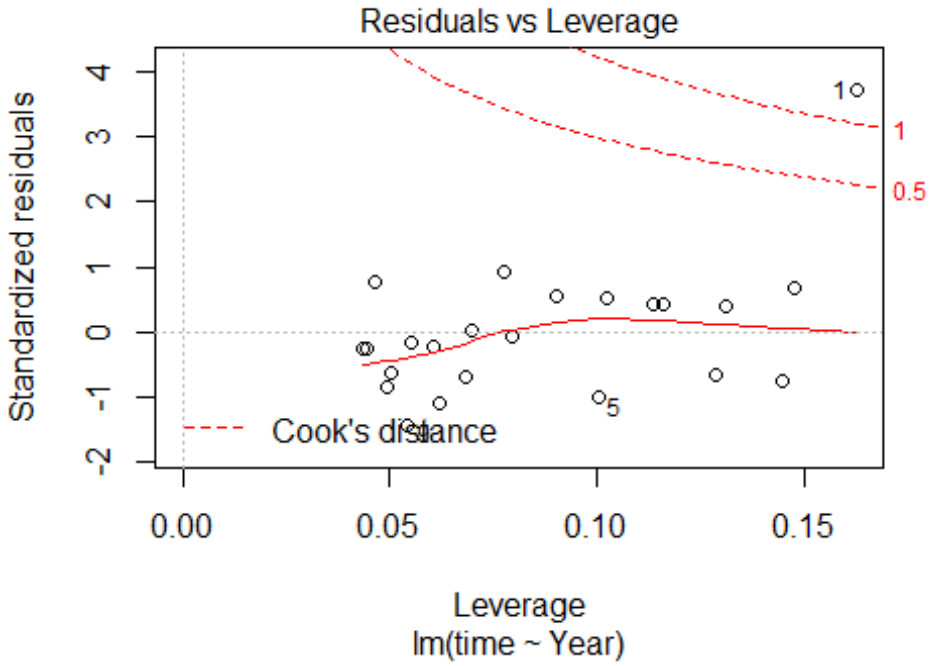



```r
olympic_ts <- ts(olympic,start=1,end=28)

#Question 4) Plot the residuals against the year. What does this indicate
about the suitability of the fitted line?

plot(oly_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(time ~ Year)
1
16
9
-2
-1
0
1
2
3
4
44
46
48
50

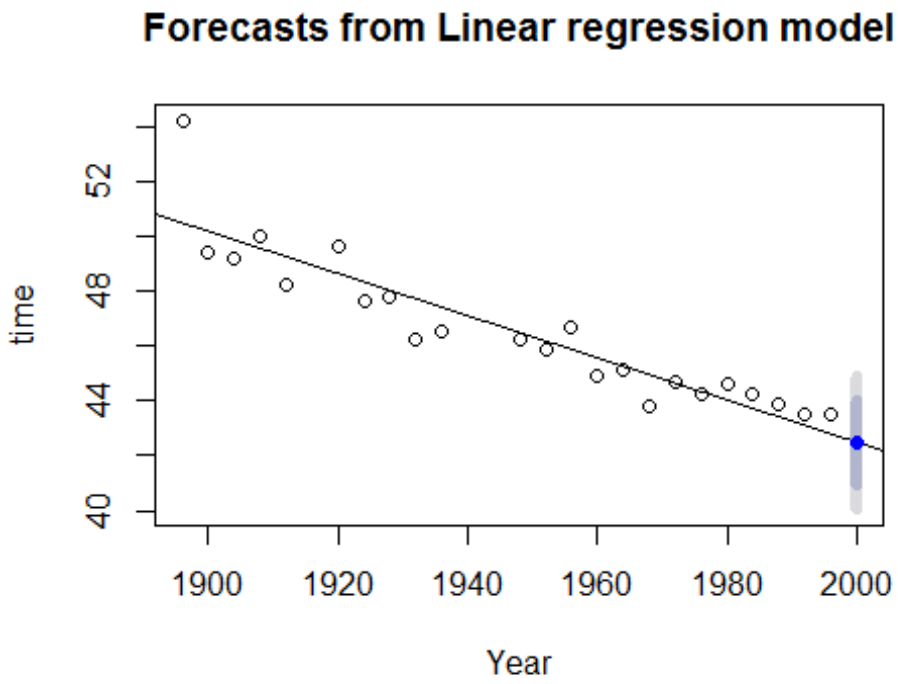

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(time ~ Year)
1
16
9
-2
-1
0
1
2
3
4



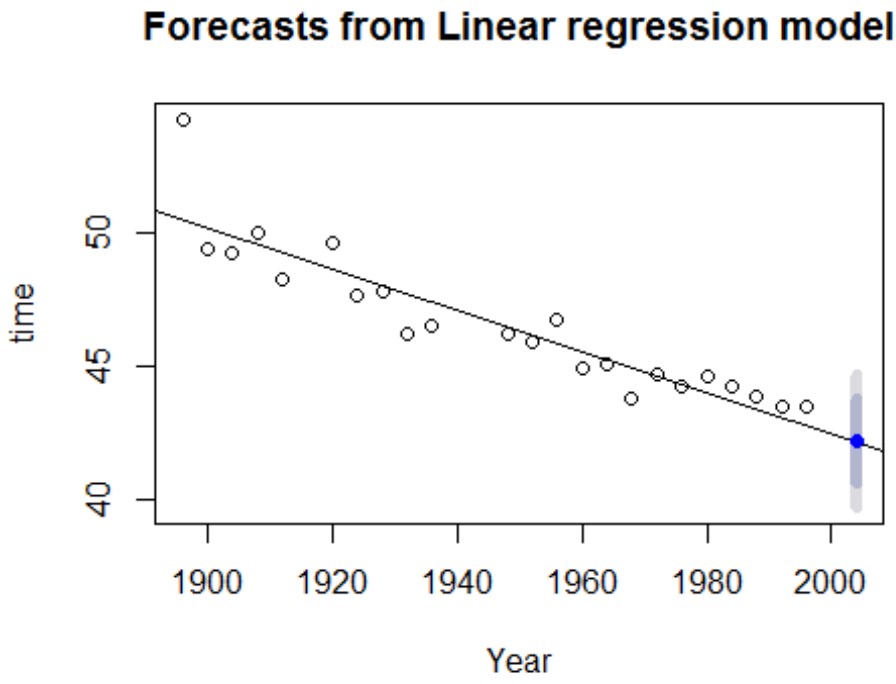



```
#Question 5) Predict the winning time for the men's 400 meters final in the
2000, 2004, 2008 and 2012 Olympics. Give a prediction interval for each of
your forecasts. What assumptions have you made in these calculations?
```

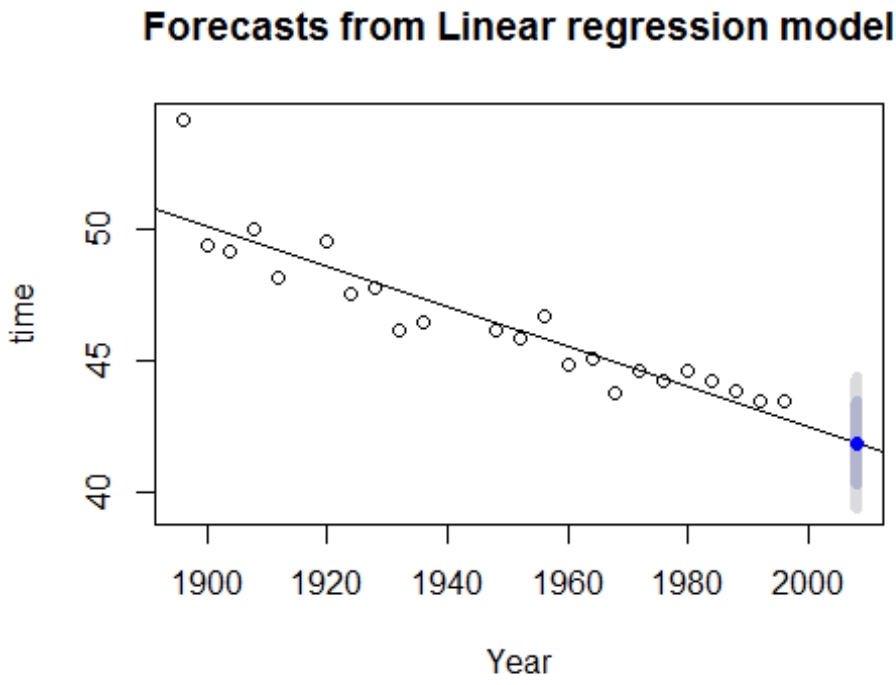
```
coeff = coefficients(oly_fit)
pred_time = c(2000, 2004, 2008, 2012)
p_time = coeff[1] + coeff[2]*pred_time
p_time

## [1] 42.49977 42.19261 41.88545 41.57829

fcast3 <- forecast(oly_fit, newdata=data.frame(Year=2000))
plot(fcast3, xlab="Year", ylab="time")
```



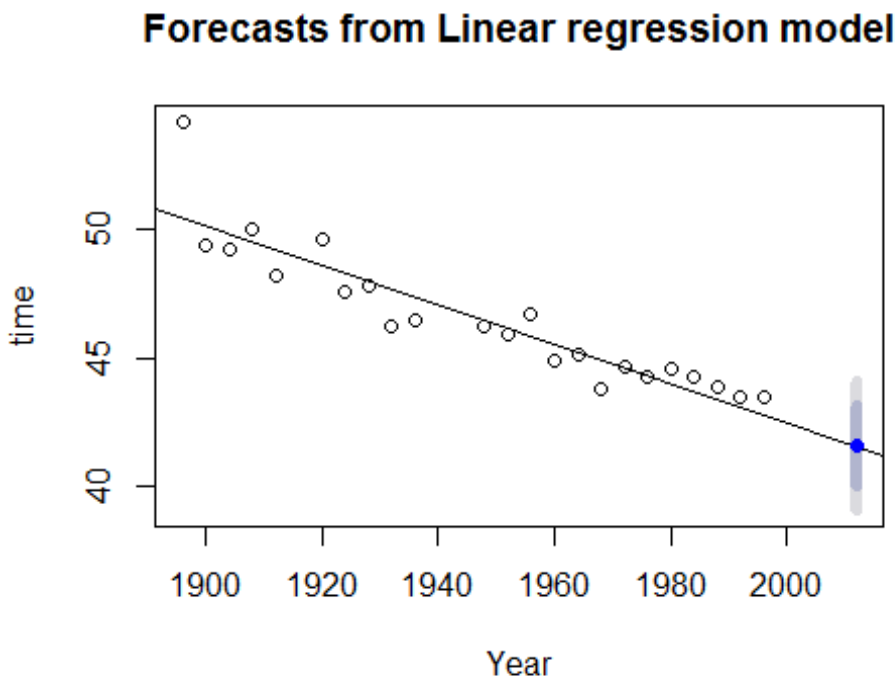

```
fcast4 <- forecast(oly_fit, newdata=data.frame(Year=2004))
plot(fcast4, xlab="Year", ylab="time")
```




```
fcast5 <- forecast(oly_fit, newdata=data.frame(Year=2008))
plot(fcast5, xlab="Year", ylab="time")
```



```
fcast6 <- forecast(oly_fit, newdata=data.frame(Year=2012))
plot(fcast6, xlab="Year", ylab="time")
```




```
time2000 = data.frame(Year=2000)
time2004 = data.frame(Year=2004)
time2008 = data.frame(Year=2008)
time2012 = data.frame(Year=2012)

predict(oly_fit, time2000, interval="predict")

##        fit      lwr      upr
## 1 42.49977 40.05401 44.94554

predict(oly_fit, time2004, interval="predict")

##        fit      lwr      upr
## 1 42.19261 39.72657 44.65866

predict(oly_fit, time2008, interval="predict")

##        fit      lwr      upr
## 1 41.88545 39.39782 44.37308

predict(oly_fit, time2012, interval="predict")

##        fit     lwr      upr
## 1 41.57829 39.0678 44.08879
```

```r
#Question 6 ) Find out the actual winning times for these Olympics (see
www.databaseolympics.com). How good were your forecasts and prediction
intervals?

coeffs1 = coefficients(oly_fit)
pred_time = c(2000, 2004, 2008, 2012)
p_time = coeffs1[1] + coeffs1[2]*pred_time
p_time
```

```
## [1] 42.49977 42.19261 41.88545 41.57829
```

```r
##############################################################################
#########################

#An elasticity coefficient is the ratio of the percentage change in the
forecast variable (yy) to the percentage change in the predictor variable
(xx). Mathematically, the elasticity is defined as
(dy/dx)×(x/y)(dy/dx)×(x/y). Consider the log-log model,
## logy=??0+??1logx+??.
#Express yy as a function of xx and show that the coefficient ??1??1 is the
elasticity coefficient.


# logy=B0+B1logx+e
#Taking Diffrential

#dy/y = dx/x*B1 (dy/y / dx/x) = B1

# change in y divided by y over the the change in x divided by x

#100(dy/y) = 100(dx/x)B1 % change y = % change x B1

##############################################################################
##################

###The data below (data set fancy) concern the monthly sales figures of a
shop which opened in January 1987 and sells gifts, souvenirs, and novelties.
The shop is situated on the wharf at a beach resort town in Queensland,
Australia. The sales volume varies with the seasonal population of tourists.
There is a large influx of visitors to the town at Christmas and for the
local surfing festival, held every March since 1988. Over time, the shop has
expanded its premises, range of products, and staff.

#Question 1 ) ) Produce a time plot of the data and describe the patterns in
the graph.
#Identify any unusual or unexpected fluctuations in the time series.
```

```
library(fma)
library(fpp)
```

```
## Warning: package 'fpp' was built under R version 3.3.3
```

```
## Loading required package: expsmooth
```

```
## Warning: package 'expsmooth' was built under R version 3.3.3
```

```
## Loading required package: lmtest
```

```
## Warning: package 'lmtest' was built under R version 3.3.3
```

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

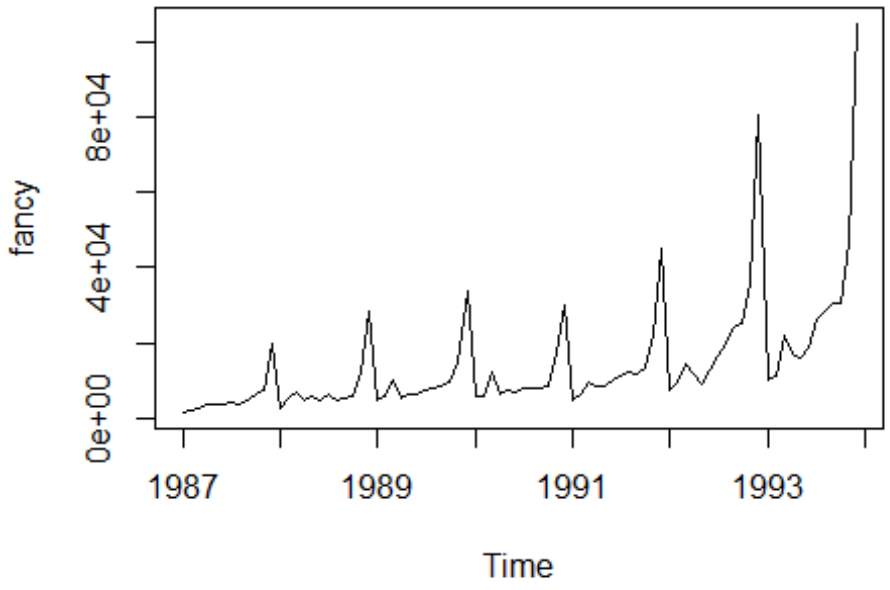
```
##
## Attaching package: 'zoo'
```

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

```
## Loading required package: tseries
```

```
## Warning: package 'tseries' was built under R version 3.3.3
```

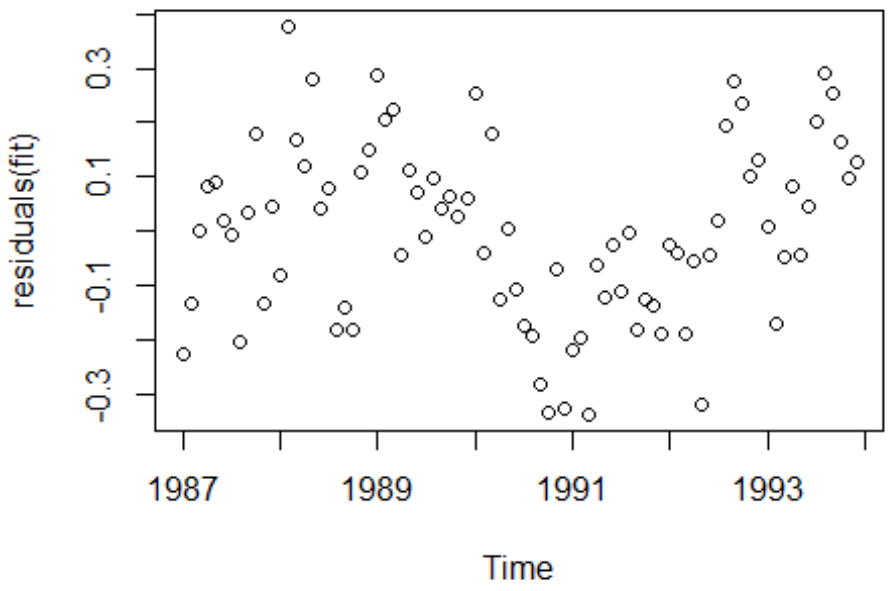
```
plot(fancy)
```



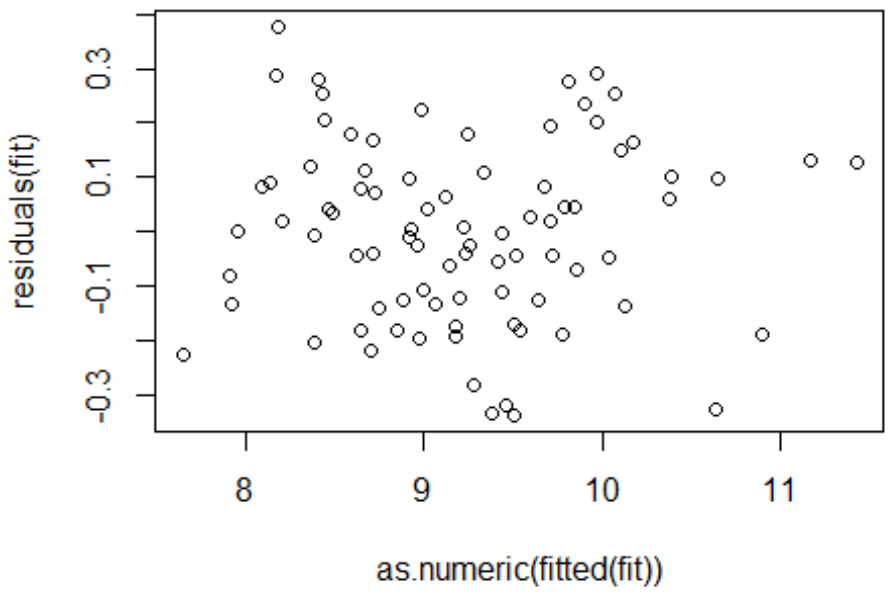
```r
#Question 2 ) Explain why it is necessary to take logarithms of these data
before fitting a model.

# Answer 2 ) It is because of the increasing seasonal fluctuations
log_fancy <- log(fancy)

#question 3 ) c) Use R to fit a regression model to the logarithms of these
sales data with a linear trend, seasonal dummies and a "surfing festival"
dummy variable.

log_fancy <- log(fancy)
dummy_fest = rep(0, length(fancy))
dummy_fest[seq_along(dummy_fest)%%12 == 3] <- 1
dummy_fest[3] <- 0
dummy_fest <- ts(dummy_fest, freq = 12, start=c(1987,1))
my_data <- data.frame(
  log_fancy,
  dummy_fest
)

fit <- tslm(log_fancy ~ trend + season + dummy_fest, data=my_data)

future_data <- data.frame(
  dummy_fest = rep(0, 12)
)
future_data[3,] <- 1
forecast(fit, newdata=future_data)
```

```
##          Point Forecast     Lo 80     Hi 80     Lo 95    Hi 95
## Jan 1994       9.491352  9.238522  9.744183  9.101594  9.88111
## Feb 1994       9.764789  9.511959 10.017620  9.375031 10.15455
## Mar 1994      10.302990 10.048860 10.557120  9.911228 10.69475
## Apr 1994       9.941465  9.688635 10.194296  9.551707 10.33122
## May 1994       9.988919  9.736088 10.241749  9.599161 10.37868
## Jun 1994      10.050280  9.797449 10.303110  9.660522 10.44004
## Jul 1994      10.233926  9.981095 10.486756  9.844168 10.62368
## Aug 1994      10.233456  9.980625 10.486286  9.843698 10.62321
## Sep 1994      10.336841 10.084010 10.589671  9.947083 10.72660
## Oct 1994      10.436923 10.184092 10.689753 10.047165 10.82668
## Nov 1994      10.918299 10.665468 11.171129 10.528541 11.30806
## Dec 1994      11.695812 11.442981 11.948642 11.306054 12.08557
```

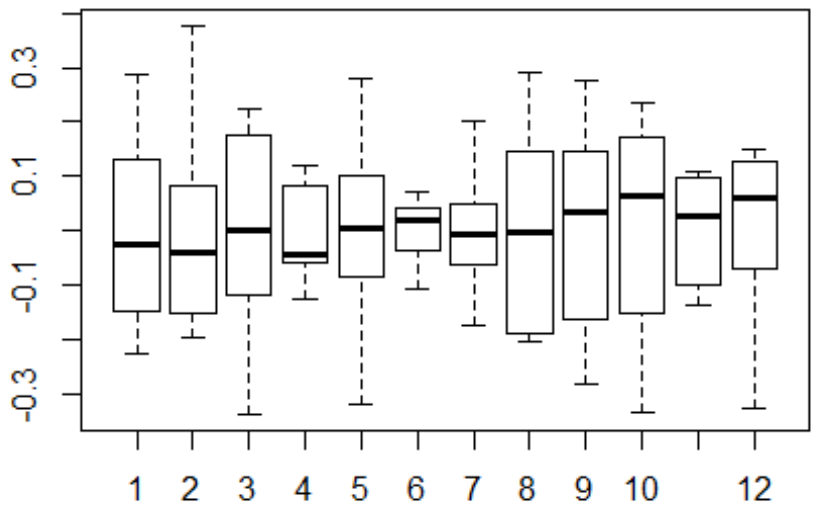
```r
#question 4  d)  Plot the residuals against time and against the fitted
values. Do these plots reveal any problems with the model?


plot(residuals(fit), type='p')
```




```r
plot(as.numeric(fitted(fit)), residuals(fit), type='p')
```

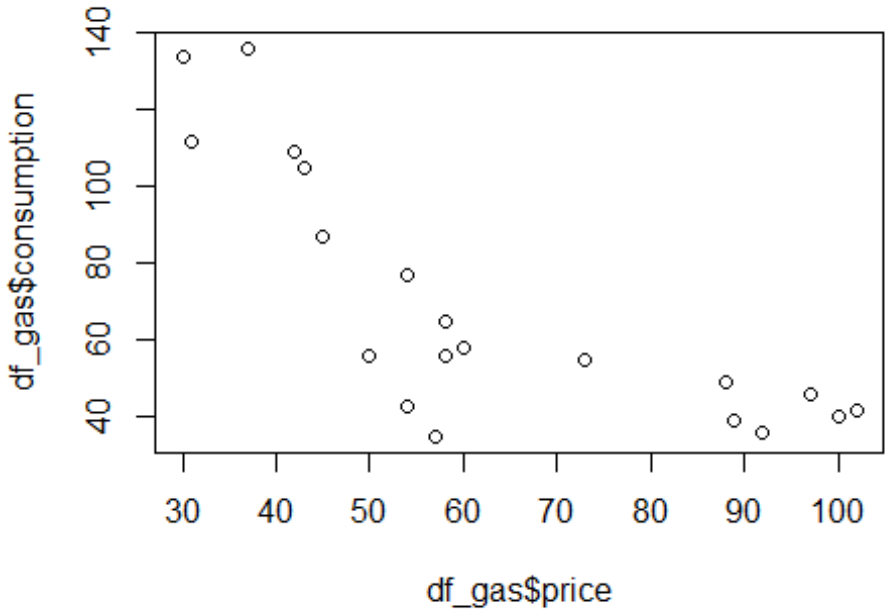


```
#Question 5 ) e)  Do boxplots of the residuals for each month. Does this
reveal any problems with the model?

boxplot(resid(fit) ~ cycle(resid(fit)))
```




```
#Question 5 )  ;  f) What do the values of the coefficients tell you about
each variable?

# The value of the coefficients show how much the model thinks each month
contributes
# to the conditional mean of the model.

#Question 5 g) What does the Durbin-Watson statistic tell you about your
model?


dwtest(fit)

##
##  Durbin-Watson test
##
## data:  fit
## DW = 0.88889, p-value = 9.78e-08
## alternative hypothesis: true autocorrelation is greater than 0

# Question 6 h ) Regardless of your answers to the above questions, use your
regression model to predict the monthly sales for 1994, 1995, and 1996.
```

```r
Produce prediction intervals for each of your forecasts.


future_data <- data.frame(
  dummy_fest = rep(0, 36)
)
preds <- forecast(fit, newdata=future_data)


# Question 5 i ) Transform your predictions and intervals to obtain
predictions and intervals for the raw data.


df_pred <- as.data.frame(preds)
df_pred <- exp(df_pred)

# Question 5 j ) How could you improve these predictions by modifying the
model?

#ANswers 5 J) We Could use  consider using a dynamic-regression model



################################################################################
##########################


# Question 5b ) (1) The data below (data set texasgas) shows the demand for
natural gas and the price of natural gas for 20 towns in Texas in 1969.

# a) Do a scatterplot of consumption against price. The data are clearly not
linear. Three possible nonlinear models for the data are given below

library(fma)
library(fpp)
library(segmented)

## Warning: package 'segmented' was built under R version 3.3.2

df_gas <- (texasgas)
plot(df_gas$price, df_gas$consumption)
```




```
# b) Can you explain why the slope of the fitted line should change with
price?

# Ans The data is not linear so the slope needs to change in order to get the
Data from our model

# c)  Fit the three models and find the coefficients, and residual variance
in each case.

# First Model

fit_gas <- lm(consumption ~ exp(price), df_gas)

fit_gas

##
## Call:
## lm(formula = consumption ~ exp(price), data = df_gas)
##
## Coefficients:
## (Intercept)   exp(price)
##   7.086e+01   -1.642e-43

# Residual Variance

(summary(fit_gas)$sigma)**2
```

```
## [1] 1101.359
```

```
# Second model - piecewise linear regression


lin.mod <- lm(consumption ~ price, df_gas)

lin.mod
```

```
##
## Call:
## lm(formula = consumption ~ price, data = df_gas)
##
## Coefficients:
## (Intercept)        price
##     138.561       -1.104
```

```
segmented.mod <- segmented(lin.mod, seg.Z = ~price, psi=60)

slope(segmented.mod)
```

```
## $price
##           Est. St.Err. t value CI(95%).l CI(95%).u
## slope1 -3.1470  0.5102  -6.169   -4.2290   -2.0660
## slope2 -0.3075  0.2220  -1.385   -0.7782    0.1632
```

```
# Residual variance
(summary(segmented.mod)$sigma)**2
```

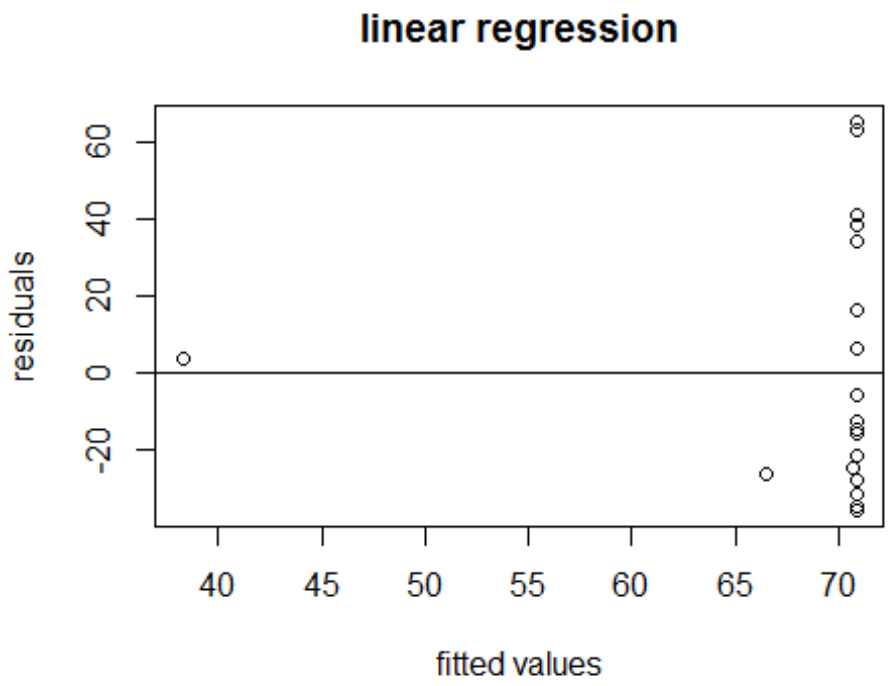
```
## [1] 167.8511
```

```
# Third model - polynomial regression

poly_fit <- lm(consumption ~ poly(price, 2), df_gas)

# Residual variance
(summary(poly_fit)$sigma)**2
```

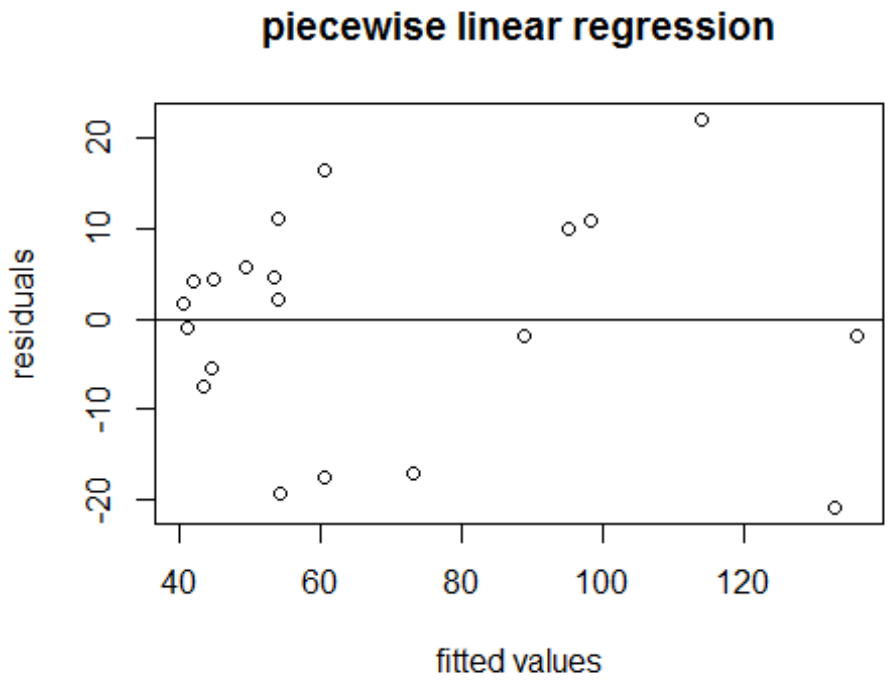
```
## [1] 206.5276
```

```
#d) For each model, find the value of R2 and AIC, and produce a residual
plot. Comment on the adequacy of the three models.

# First model - basic linear regression

# Adjusted R-squared: -0.004
# AIC: 200.736

resid <- residuals(fit_gas)
plot(fit_gas$fitted.values, resid, ylab='residuals', xlab='fitted values',
```

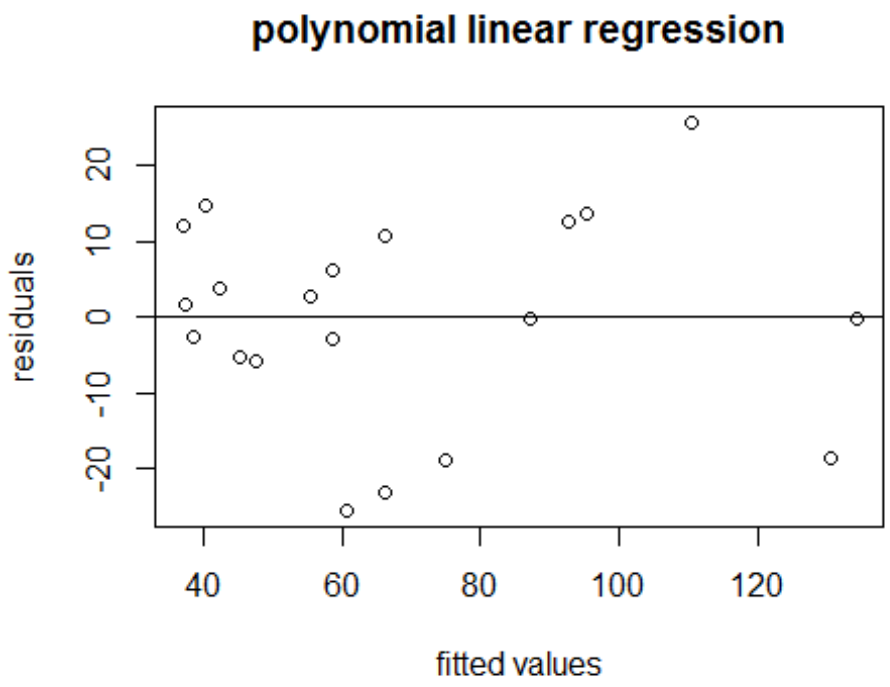
```
main='linear regression')
abline(0,0)
```



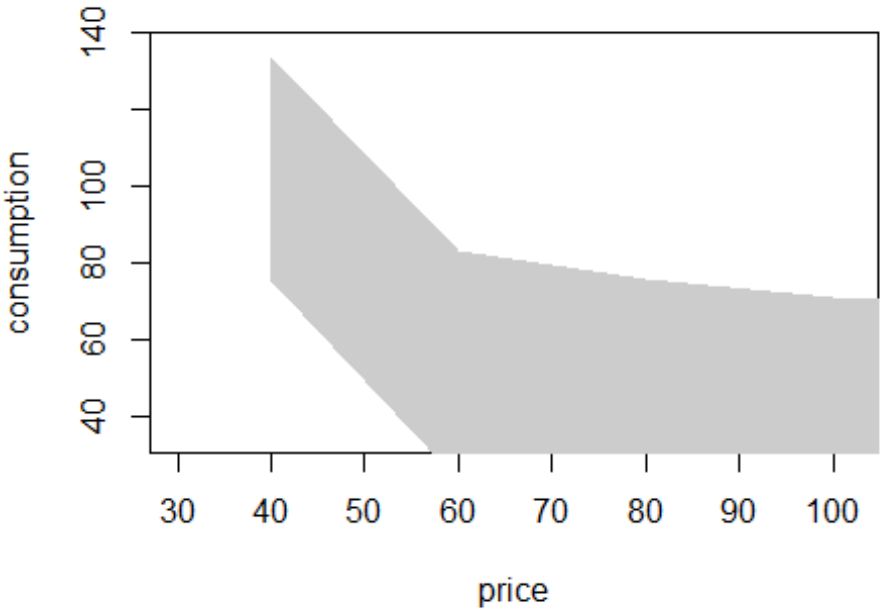

```
# Second model - piecewise linear regression

# Adjusted R-squared:  0.847
# AIC: 164.756

resid <- residuals(segmented.mod)
plot(segmented.mod$fitted.values, resid, ylab='residuals', xlab='fitted
values',
     main='piecewise linear regression')
abline(0,0)
```




```r
# Third model - polynomial regression.

# Adjusted R-squared:  0.812
# AIC: 168.116

resid <- residuals(poly_fit)
plot(poly_fit$fitted.values, resid, ylab='residuals', xlab='fitted values',
     main='polynomial linear regression')
abline(0,0)
```




```
# e) For prices 40, 60, 80, 100, and 120 cents per 1,000 cubic feet, compute
the forecasted per capita demand using the best model of the three above.


new.data <- data.frame(price=c(40, 60, 80, 100, 120))
predict(segmented.mod, new.data)

##         1         2         3         4         5
## 104.53618  53.34514  47.19593  41.04673  34.89752

# F ) Compute 95% prediction intervals. Make a graph of these prediction
intervals and discuss their interpretation.

newx <- seq(min(new.data), max(new.data), length.out=5)
intervals <- predict(segmented.mod, new.data, interval="predict")

plot(consumption ~ price, data = df_gas, type = 'n')

polygon(c(rev(newx), newx), c(rev(intervals[ ,3]), intervals[ ,2]), col =
'grey80', border = NA)
```


consumption
140
100
80
60
40
30
40
50
60
70
80
90
100
price