Homework 1

Partha S Satpathy

April 10, 2017

## Library fma

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

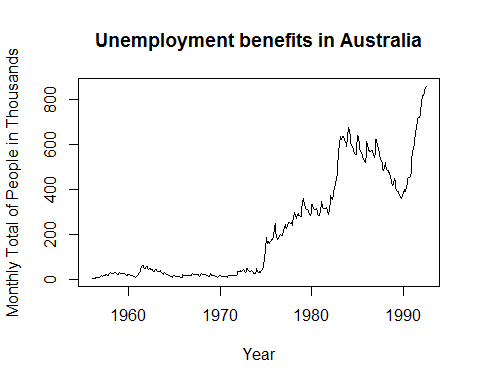
data(package='fma')

## Exercise 2.8

##Exercise 1a  
##Data is dole- 'Unemployment benefits in Australia'  
##Monthly total of people on unemployed benefits in Australia (January 1956–July 1992).  
print(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

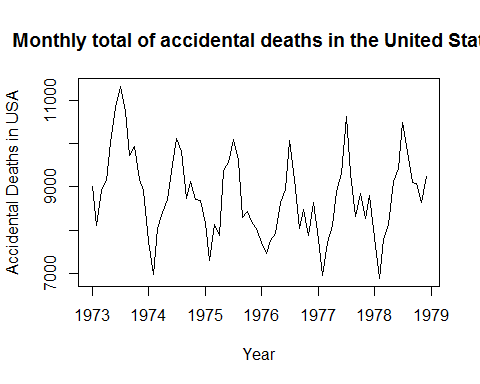
plot(dole/1000,xlab="Year",ylab="Monthly Total of People in Thousands",main="Unemployment benefits in Australia")



##Exercise 1b  
##Monthly total of accidental deaths in the United States (January 1973–December 1978).  
##Data- usdeaths  
print(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

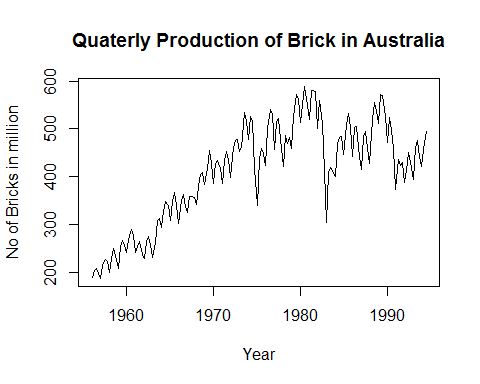
plot(usdeaths,xlab="Year",ylab="Accidental Deaths in USA",main="Monthly total of accidental deaths in the United States")



##Exercise 1c  
##bricksq- Quaterly Production of Brick in Australia  
print(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

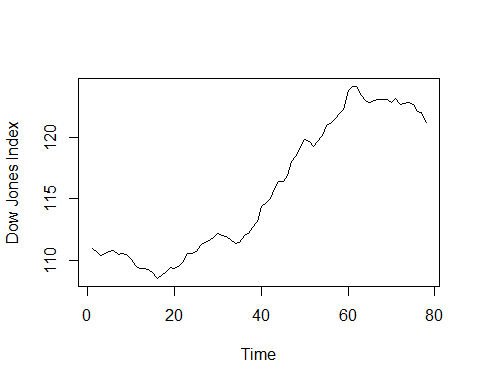
plot(bricksq,xlab="Year",ylab="No of Bricks in million",main="Quaterly Production of Brick in Australia")



##Exercise 2a  
print(dowjones)

## Time Series:  
## Start = 1   
## End = 78   
## Frequency = 1   
## [1] 110.94 110.69 110.43 110.56 110.75 110.84 110.46 110.56 110.46 110.05  
## [11] 109.60 109.31 109.31 109.25 109.02 108.54 108.77 109.02 109.44 109.38  
## [21] 109.53 109.89 110.56 110.56 110.72 111.23 111.48 111.58 111.90 112.19  
## [31] 112.06 111.96 111.68 111.36 111.42 112.00 112.22 112.70 113.15 114.36  
## [41] 114.65 115.06 115.86 116.40 116.44 116.88 118.07 118.51 119.28 119.79  
## [51] 119.70 119.28 119.66 120.14 120.97 121.13 121.55 121.96 122.26 123.79  
## [61] 124.11 124.14 123.37 123.02 122.86 123.02 123.11 123.05 123.05 122.83  
## [71] 123.18 122.67 122.73 122.86 122.67 122.09 122.00 121.23

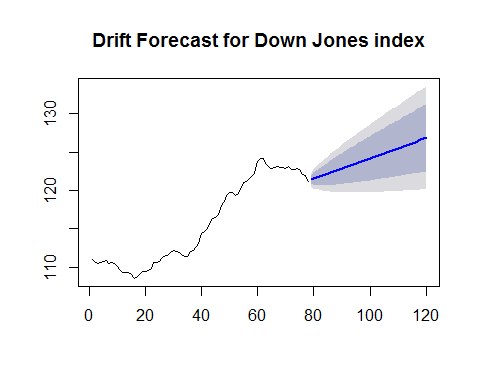
ts.plot(dowjones,ylab="Dow Jones Index")



##Exercise 2b  
rwf(dowjones, h=42, drift=TRUE)

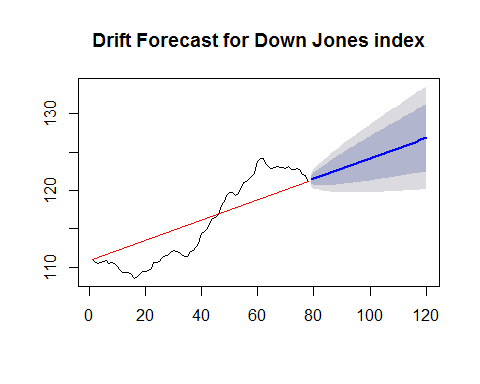
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 79 121.3636 120.8165 121.9108 120.5268 122.2004  
## 80 121.4973 120.7185 122.2761 120.3062 122.6884  
## 81 121.6309 120.6710 122.5908 120.1628 123.0990  
## 82 121.7645 120.6491 122.8800 120.0586 123.4705  
## 83 121.8982 120.6433 123.1530 119.9790 123.8173  
## 84 122.0318 120.6487 123.4149 119.9166 124.1471  
## 85 122.1655 120.6625 123.6684 119.8668 124.4641  
## 86 122.2991 120.6827 123.9155 119.8270 124.7712  
## 87 122.4327 120.7081 124.1574 119.7951 125.0703  
## 88 122.5664 120.7378 124.3950 119.7698 125.3630  
## 89 122.7000 120.7710 124.6290 119.7499 125.6501  
## 90 122.8336 120.8074 124.8599 119.7347 125.9326  
## 91 122.9673 120.8463 125.0882 119.7235 126.2110  
## 92 123.1009 120.8875 125.3143 119.7159 126.4860  
## 93 123.2345 120.9308 125.5383 119.7113 126.7578  
## 94 123.3682 120.9758 125.7605 119.7094 127.0269  
## 95 123.5018 121.0225 125.9811 119.7100 127.2936  
## 96 123.6355 121.0706 126.2003 119.7128 127.5581  
## 97 123.7691 121.1199 126.4182 119.7176 127.8206  
## 98 123.9027 121.1705 126.6350 119.7241 128.0813  
## 99 124.0364 121.2221 126.8506 119.7323 128.3404  
## 100 124.1700 121.2747 127.0653 119.7420 128.5980  
## 101 124.3036 121.3282 127.2791 119.7531 128.8542  
## 102 124.4373 121.3825 127.4920 119.7655 129.1091  
## 103 124.5709 121.4376 127.7042 119.7790 129.3628  
## 104 124.7045 121.4934 127.9157 119.7936 129.6155  
## 105 124.8382 121.5499 128.1265 119.8092 129.8672  
## 106 124.9718 121.6070 128.3367 119.8257 130.1179  
## 107 125.1055 121.6646 128.5463 119.8432 130.3677  
## 108 125.2391 121.7228 128.7554 119.8614 130.6168  
## 109 125.3727 121.7815 128.9639 119.8804 130.8650  
## 110 125.5064 121.8407 129.1720 119.9002 131.1125  
## 111 125.6400 121.9003 129.3797 119.9206 131.3594  
## 112 125.7736 121.9603 129.5870 119.9417 131.6056  
## 113 125.9073 122.0207 129.7938 119.9633 131.8512  
## 114 126.0409 122.0815 130.0003 119.9855 132.0963  
## 115 126.1745 122.1427 130.2064 120.0083 132.3408  
## 116 126.3082 122.2041 130.4122 120.0316 132.5848  
## 117 126.4418 122.2659 130.6177 120.0553 132.8283  
## 118 126.5755 122.3280 130.8229 120.0796 133.0714  
## 119 126.7091 122.3904 131.0278 120.1042 133.3140  
## 120 126.8427 122.4530 131.2324 120.1293 133.5562

plot(rwf(dowjones, h=42, drift=TRUE),main="Drift Forecast for Down Jones index")

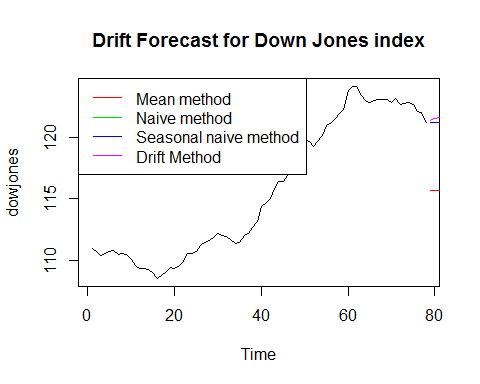


#segments()

##Exercise 2c  
plot(rwf(dowjones, h=42, drift=TRUE),main="Drift Forecast for Down Jones index")  
segments(1,110.94,78,121.23,col = "red")



##Exercise 2d  
#train = window(dowjones,start=c(1),end=c(60))  
fMean = meanf(dowjones,42)  
fNaive = naive(dowjones,42)  
fSnaive = snaive(dowjones,42)  
fDrift = rwf(dowjones, h=42, drift=TRUE)  
plot(dowjones,main="Drift Forecast for Down Jones index")  
lines(fMean$mean,col=2)  
lines(fNaive$mean,col=3)  
lines(fSnaive$mean,col=4)  
lines(fDrift$mean,col=6)  
legend("topleft",lty=1,col=c(2,3,4,6),  
 legend=c("Mean method","Naive method","Seasonal naive method","Drift Method"))



#test = window(dowjones,start=c(61))  
accuracy(fMean)

## ME RMSE MAE MPE MAPE MASE  
## Training set -3.464321e-15 5.470653 5.104103 -0.221695 4.399889 14.93789  
## ACF1  
## Training set 0.9853294

accuracy(fNaive)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.1336364 0.4447223 0.3416883 0.1144757 0.2936792 1  
## ACF1  
## Training set 0.4218786

accuracy(fSnaive)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.1336364 0.4447223 0.3416883 0.1144757 0.2936792 1  
## ACF1  
## Training set 0.4218786

accuracy(fDrift)

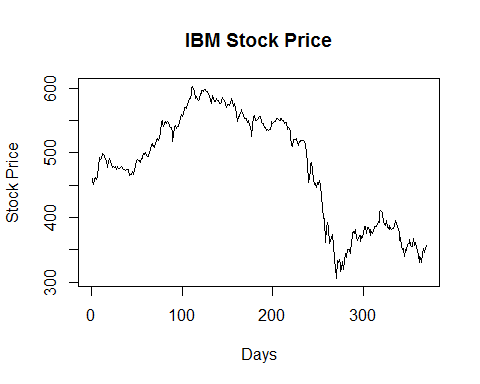
## ME RMSE MAE MPE MAPE  
## Training set -6.274917e-15 0.4241689 0.3253365 -0.001238663 0.2796548  
## MASE ACF1  
## Training set 0.952144 0.4218786

##The Drift method works best as it has the least error

##Exercise 3a  
print(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

plot(ibmclose,xlab="Days",ylab="Stock Price",main="IBM Stock Price")



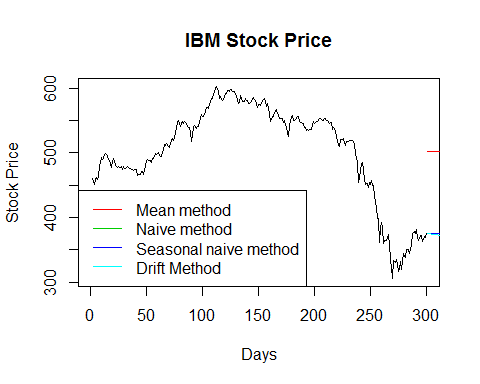
##Exercise 3b  
train\_ibmclose = window(ibmclose,start=1,end = 300)  
test\_ibmclose = window(ibmclose,start=301)  
print(train\_ibmclose)

## Time Series:  
## Start = 1   
## End = 300   
## 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

print(test\_ibmclose)

## Time Series:  
## Start = 301   
## End = 369   
## Frequency = 1   
## [1] 387 387 376 385 385 380 373 382 377 376 379 386 387 386 389 394 393  
## [18] 409 411 409 408 393 391 388 396 387 383 388 382 384 382 383 383 388  
## [35] 395 392 386 383 377 364 369 355 350 353 340 350 349 358 360 360 366  
## [52] 359 356 355 367 357 361 355 348 343 330 340 339 331 345 352 346 352  
## [69] 357

##Exercise 3b  
train\_ibmclose = window(ibmclose,start=1,end = 300)  
test\_ibmclose = window(ibmclose,start=301)  
  
fMean = meanf(train\_ibmclose,h=42)  
fNaive = naive(train\_ibmclose,h=42)  
fSnaive = snaive(train\_ibmclose,h=42)  
fdrift = rwf(train\_ibmclose,h=42,drift = T)  
  
plot(train\_ibmclose,xlab="Days",ylab="Stock Price",main="IBM Stock Price")  
lines(fMean$mean,col=2)  
lines(fNaive$mean,col=3)  
lines(fSnaive$mean,col=4)  
lines(fdrift$mean,col=5)  
legend("bottomleft",lty=1,col=c(2,3,4,5),  
 legend=c("Mean method","Naive method","Seasonal naive method","Drift Method"))



print(accuracy(fMean,test\_ibmclose))

## ME RMSE MAE MPE MAPE  
## Training set 1.660438e-14 73.61532 58.72231 -2.642058 13.03019  
## Test set -1.169886e+02 117.49988 116.98857 -30.420483 30.42048  
## MASE ACF1 Theil's U  
## Training set 11.52098 0.9895779 NA  
## Test set 22.95248 0.7296952 17.9457

print(accuracy(fNaive,test\_ibmclose))

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.2809365 7.302815 5.09699 -0.08262872 1.115844 1.000000  
## Test set 9.9047619 14.764823 11.95238 2.48806561 3.055375 2.344988  
## ACF1 Theil's U  
## Training set 0.1351052 NA  
## Test set 0.7296952 2.205498

print(accuracy(fSnaive,test\_ibmclose))

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.2809365 7.302815 5.09699 -0.08262872 1.115844 1.000000  
## Test set 9.9047619 14.764823 11.95238 2.48806561 3.055375 2.344988  
## ACF1 Theil's U  
## Training set 0.1351052 NA  
## Test set 0.7296952 2.205498

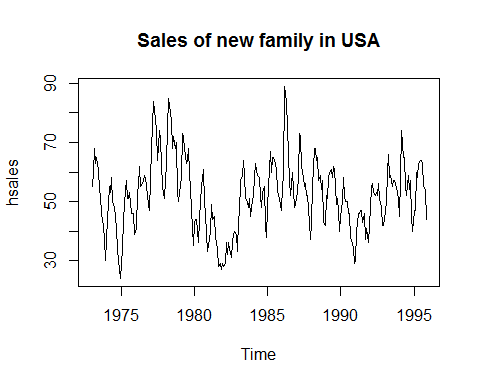
print(accuracy(fdrift,test\_ibmclose))

## ME RMSE MAE MPE MAPE MASE  
## Training set 2.87048e-14 7.297409 5.127996 -0.02530123 1.121650 1.006083  
## Test set 1.59449e+01 19.323121 16.468546 4.05898695 4.205572 3.231034  
## ACF1 Theil's U  
## Training set 0.1351052 NA  
## Test set 0.7596624 2.898499

##Exercise 4a  
print(hsales)

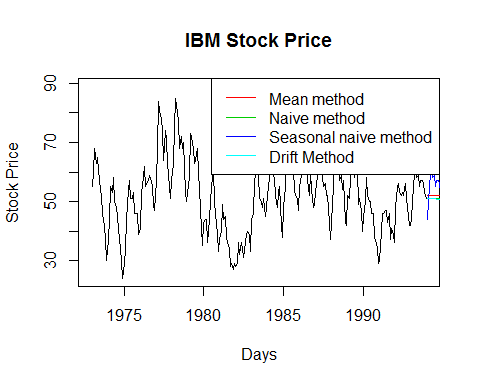
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1973 55 60 68 63 65 61 54 52 46 42 37 30  
## 1974 37 44 55 53 58 50 48 45 41 34 30 24  
## 1975 29 34 44 54 57 51 51 53 46 46 46 39  
## 1976 41 53 55 62 55 56 57 59 58 55 49 47  
## 1977 57 68 84 81 78 74 64 74 71 63 55 51  
## 1978 57 63 75 85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

plot(hsales,main="Sales of new family in USA")



train\_hsales = window(hsales,start=c(1973,1),end=c(1993,12))  
test\_hsales = window(hsales,start=c(1994,1))

fMean = meanf(train\_hsales,h=42)  
fNaive = naive(train\_hsales,h=42)  
fSnaive = snaive(train\_hsales,h=42)  
fdrift = rwf(train\_hsales,h=42,drift = T)  
  
plot(train\_hsales,xlab="Days",ylab="Stock Price",main="IBM Stock Price")  
lines(fMean$mean,col=2)  
lines(fNaive$mean,col=3)  
lines(fSnaive$mean,col=4)  
lines(fdrift$mean,col=5)  
legend("topright",lty=1,col=c(2,3,4,5),  
 legend=c("Mean method","Naive method","Seasonal naive method","Drift Method"))



print(accuracy(fMean,test\_hsales))

## ME RMSE MAE MPE MAPE MASE  
## Training set 2.480763e-15 12.138802 9.498898 -6.120182 20.30851 1.119163  
## Test set 4.051587e+00 9.216133 7.850759 5.074990 13.75973 0.924979  
## ACF1 Theil's U  
## Training set 0.8661515 NA  
## Test set 0.5095178 1.13105

print(accuracy(fNaive,test\_hsales))

## ME RMSE MAE MPE MAPE MASE  
## Training set -0.01593625 6.289813 4.988048 -0.7800232 9.880157 0.5876934  
## Test set 5.00000000 9.670664 8.304348 6.8080182 14.381673 0.9784210  
## ACF1 Theil's U  
## Training set 0.1829708 NA  
## Test set 0.5095178 1.179633

print(accuracy(fSnaive,test\_hsales))

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.1375000 10.576113 8.4875 -2.1016380 17.63375 1.0000000  
## Test set 0.3043478 6.160886 5.0000 -0.7312374 9.12828 0.5891016  
## ACF1 Theil's U  
## Training set 0.838108 NA  
## Test set 0.224307 0.8031005

print(accuracy(fdrift,test\_hsales))

## ME RMSE MAE MPE MAPE MASE  
## Training set 2.377739e-15 6.289793 4.987730 -0.7474544 9.87819 0.5876560  
## Test set 5.191235e+00 9.761548 8.393037 7.1599507 14.50303 0.9888703  
## ACF1 Theil's U  
## Training set 0.1829708 NA  
## Test set 0.5083059 1.188562

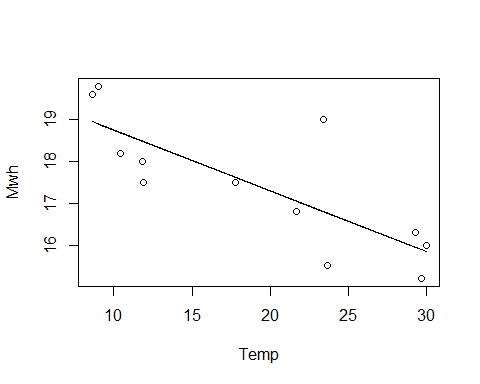
##SNaive works best as it has the least test error

## Exercise 4.10

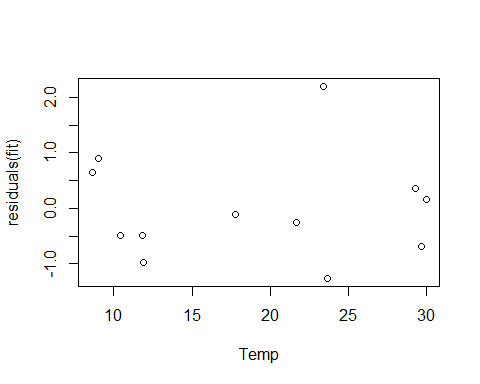
mwh <- c(16.3,16.8,15.5,18.2,15.2,17.5,19.8,19.0,17.5,16.0,19.6,18.0)  
temp <- c(29.3,21.7,23.7,10.4,29.7,11.9,9.0,23.4,17.8,30.0,8.6,11.8)  
econsumtion <- data.frame(cbind(mwh,temp))  
names(econsumtion) <- c('Mwh','Temp')  
attach(econsumtion)  
  
plot(Mwh~Temp)  
fit <- lm(Mwh~Temp)  
summary(fit)

##   
## Call:  
## lm(formula = Mwh ~ Temp)  
##   
## 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  
##   
## Residual standard error: 0.9888 on 10 degrees of freedom  
## Multiple R-squared: 0.6258, Adjusted R-squared: 0.5884   
## F-statistic: 16.73 on 1 and 10 DF, p-value: 0.00218

lines(Temp,predict(fit))



plot(residuals(fit)~Temp)



##1. The R square value is 0.62 which shows that it is not a good fit  
##2. There are outliers present as can be seen from the Residual plot

#library('fma')  
#fitted(fit)[1]  
forecast(fit,newdata = data.frame(Temp=c(10,35)))

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1 18.74795 17.27010 20.22579 16.34824 21.14766  
## 2 15.11902 13.50469 16.73335 12.49768 17.74035

fcast <- forecast(fit,newdata = data.frame(Temp=c(10,35)))  
fcast

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1 18.74795 17.27010 20.22579 16.34824 21.14766  
## 2 15.11902 13.50469 16.73335 12.49768 17.74035

head(olympic)

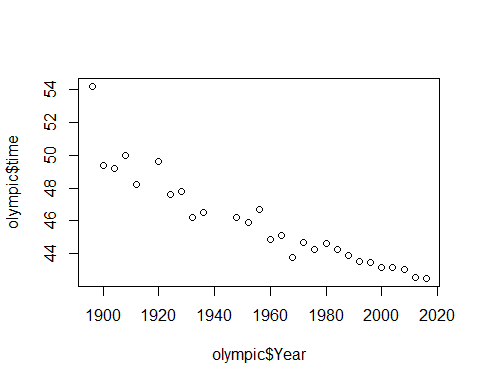
## Year time  
## 1 1896 54.2  
## 2 1900 49.4  
## 3 1904 49.2  
## 4 1908 50.0  
## 5 1912 48.2  
## 6 1920 49.6

year <- c(2000,2004,2008,2012,2016)  
time <- c(43.18,43.18,43.02,42.58,42.50)  
newrec <- data.frame(cbind(year,time))  
names(newrec)=c('Year','time')  
olympic <- rbind(olympic,newrec)

attach(olympic)

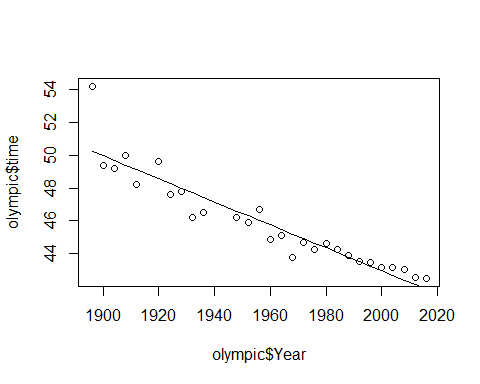
## The following object is masked \_by\_ .GlobalEnv:  
##   
## time

plot(olympic$time~olympic$Year)



##The sctterplot shows that olympic record time is decreasing linearly with each year

fit <- lm(time ~ Year, data = olympic)  
plot(olympic$time~olympic$Year)  
lines(Year,predict(fit))

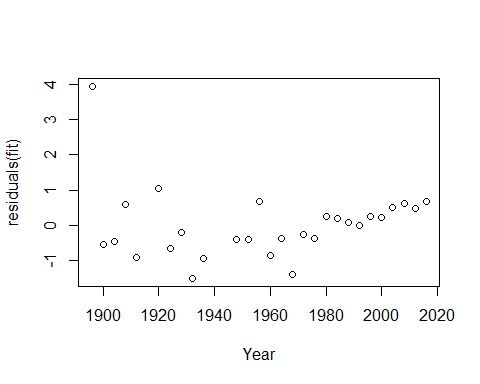


print(fit$coefficients[2])

## Year   
## -0.07002136

##The rate of dcreasing can be found from the slope of the regression lines  
##which is here --- -0.07002136

plot(residuals(fit)~Year)



##There is no pattern on the residuals. This shows that the model is fit.

forecast(fit,newdata = data.frame(Year=c(2000,2004,2008,2010)))

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 1 42.96090 41.55356 44.36823 40.76099 45.16081  
## 2 42.68081 41.26745 44.09418 40.47147 44.89015  
## 3 42.40073 40.98080 43.82065 40.18114 44.62031  
## 4 42.26068 40.83729 43.68408 40.03567 44.48570

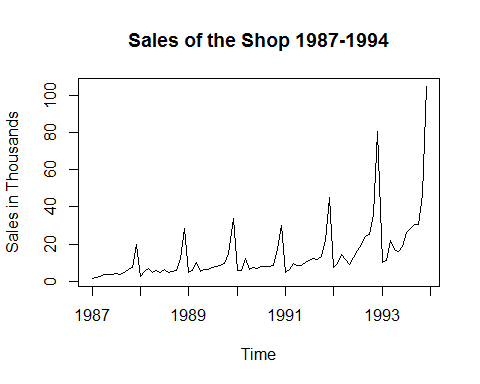
##The real data for years 2000,2004,2008,2010 are as follows:  
## 2000 43.18  
## 2004 43.18  
## 2008 43.02  
## 2012 42.58  
#This is very near to the expectations

## Exercise 5.8

print(fancy)

## Jan Feb Mar Apr May Jun Jul  
## 1987 1664.81 2397.53 2840.71 3547.29 3752.96 3714.74 4349.61  
## 1988 2499.81 5198.24 7225.14 4806.03 5900.88 4951.34 6179.12  
## 1989 4717.02 5702.63 9957.58 5304.78 6492.43 6630.80 7349.62  
## 1990 5921.10 5814.58 12421.25 6369.77 7609.12 7224.75 8121.22  
## 1991 4826.64 6470.23 9638.77 8821.17 8722.37 10209.48 11276.55  
## 1992 7615.03 9849.69 14558.40 11587.33 9332.56 13082.09 16732.78  
## 1993 10243.24 11266.88 21826.84 17357.33 15997.79 18601.53 26155.15  
## Aug Sep Oct Nov Dec  
## 1987 3566.34 5021.82 6423.48 7600.60 19756.21  
## 1988 4752.15 5496.43 5835.10 12600.08 28541.72  
## 1989 8176.62 8573.17 9690.50 15151.84 34061.01  
## 1990 7979.25 8093.06 8476.70 17914.66 30114.41  
## 1991 12552.22 11637.39 13606.89 21822.11 45060.69  
## 1992 19888.61 23933.38 25391.35 36024.80 80721.71  
## 1993 28586.52 30505.41 30821.33 46634.38 104660.67

plot(fancy/1000,ylab="Sales in Thousands",main="Sales of the Shop 1987-1994")



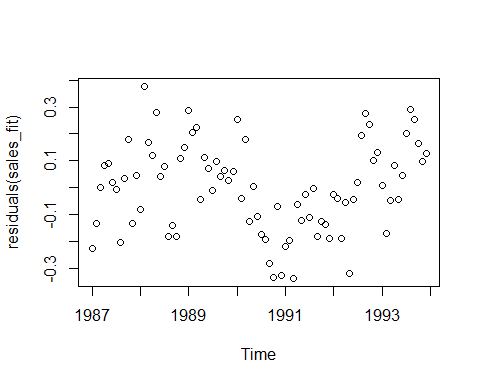
##There is a spike in sales in December of every month.  
##After 1988, there is a spike in sales in March every year  
  
##Fluctuations- In 1993, there is a very high increase in sales

##We need to take log of the time series before modelling, because that will make the   
##time series stable. That removes the fluctiations in seasonality.

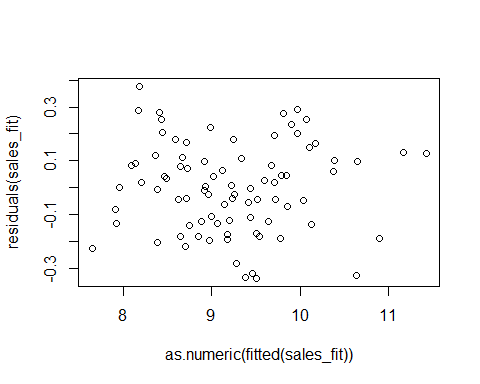
fancy\_log <- log(fancy)  
dummy\_data = rep(0, length(fancy))  
dummy\_data[seq\_along(dummy\_data)%%12 == 3] <- 1  
dummy\_data[3] <- 0  
dummy\_data <- ts(dummy\_data, freq = 12, start=c(1987,1))  
fancy\_newdata <- data.frame(fancy\_log,dummy\_data)  
  
sales\_fit <- tslm(fancy\_log ~ trend + season + dummy\_data, data=fancy\_newdata)  
  
future\_data <- data.frame(dummy\_data = rep(0, 12))  
future\_data[3,] <- 1  
  
forecast(sales\_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

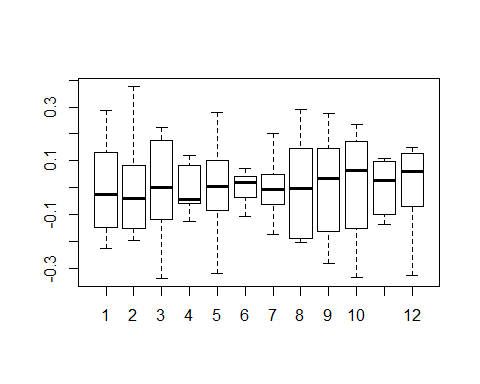
plot(residuals(sales\_fit), type='p')



plot(as.numeric(fitted(sales\_fit)), residuals(sales\_fit), type='p')



boxplot(resid(sales\_fit) ~ cycle(resid(sales\_fit)))



library('lmtest')

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

## Loading required package: zoo

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

##   
## Attaching package: 'zoo'

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

dwtest(sales\_fit, alt="two.sided")

##   
## Durbin-Watson test  
##   
## data: sales\_fit  
## DW = 0.88889, p-value = 1.956e-07  
## alternative hypothesis: true autocorrelation is not 0

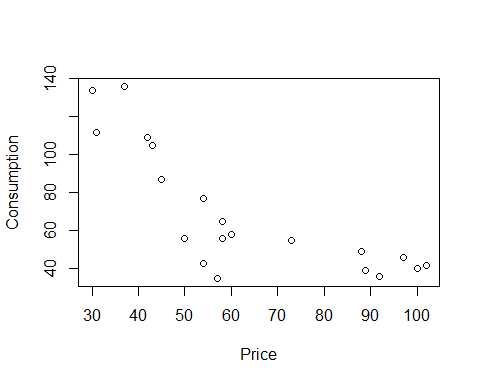
future\_data <- data.frame(dummy\_data = rep(0, 36))  
  
pred\_sales <- forecast(sales\_fit, newdata=future\_data)  
pred\_sales

## 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 9.801475 9.461879 10.141071 9.277961 10.32499  
## 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  
## Jan 1995 9.755590 9.499844 10.011336 9.361338 10.14984  
## Feb 1995 10.029027 9.773281 10.284773 9.634775 10.42328  
## Mar 1995 10.065713 9.722498 10.408928 9.536620 10.59481  
## Apr 1995 10.205703 9.949957 10.461449 9.811451 10.59996  
## May 1995 10.253157 9.997411 10.508903 9.858904 10.64741  
## Jun 1995 10.314518 10.058772 10.570264 9.920265 10.70877  
## Jul 1995 10.498164 10.242418 10.753910 10.103911 10.89242  
## Aug 1995 10.497694 10.241948 10.753440 10.103441 10.89195  
## Sep 1995 10.601079 10.345333 10.856825 10.206826 10.99533  
## Oct 1995 10.701161 10.445415 10.956907 10.306908 11.09541  
## Nov 1995 11.182537 10.926791 11.438282 10.788284 11.57679  
## Dec 1995 11.960050 11.704304 12.215796 11.565797 12.35430  
## Jan 1996 10.019828 9.760564 10.279093 9.620151 10.41951  
## Feb 1996 10.293265 10.034000 10.552530 9.893588 10.69294  
## Mar 1996 10.329951 9.982679 10.677222 9.794605 10.86530  
## Apr 1996 10.469941 10.210677 10.729206 10.070264 10.86962  
## May 1996 10.517395 10.258130 10.776659 10.117718 10.91707  
## Jun 1996 10.578756 10.319491 10.838021 10.179079 10.97843  
## Jul 1996 10.762402 10.503137 11.021667 10.362725 11.16208  
## Aug 1996 10.761932 10.502667 11.021196 10.362254 11.16161  
## Sep 1996 10.865317 10.606052 11.124582 10.465640 11.26499  
## Oct 1996 10.965399 10.706134 11.224664 10.565722 11.36508  
## Nov 1996 11.446774 11.187510 11.706039 11.047097 11.84645  
## Dec 1996 12.224288 11.965023 12.483552 11.824611 12.62396

sales\_df <- as.data.frame(pred\_sales)  
sales\_df <- exp(sales\_df)  
sales\_df

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1994 13244.70 10285.82 17054.73 8969.583 19557.43  
## Feb 1994 17409.81 13520.45 22418.00 11790.284 25707.73  
## Mar 1994 18060.36 12860.02 25363.61 10699.594 30484.96  
## Apr 1994 20774.16 16133.21 26750.16 14068.696 30675.62  
## May 1994 21783.73 16917.24 28050.15 14752.395 32166.37  
## Jun 1994 23162.27 17987.81 29825.24 15685.969 34201.95  
## Jul 1994 27831.56 21613.98 35837.72 18848.111 41096.73  
## Aug 1994 27818.48 21603.82 35820.87 18839.249 41077.41  
## Sep 1994 30848.42 23956.87 39722.43 20891.193 45551.50  
## Oct 1994 34095.57 26478.61 43903.67 23090.230 50346.32  
## Nov 1994 55176.84 42850.31 71049.28 37366.903 81475.41  
## Dec 1994 120067.79 93244.59 154607.08 81312.400 177294.90  
## Jan 1995 17250.40 13357.65 22277.59 11629.938 25587.08  
## Feb 1995 22675.20 17558.28 29283.31 15287.252 33633.55  
## Mar 1995 23522.50 16688.88 33154.31 13858.024 39926.92  
## Apr 1995 27057.06 20951.33 34942.16 18241.435 40133.06  
## May 1995 28371.96 21969.51 36640.25 19127.918 42083.42  
## Jun 1995 30167.42 23359.80 38958.95 20338.387 44746.58  
## Jul 1995 36248.88 28068.91 46812.70 24438.412 53767.06  
## Aug 1995 36231.84 28055.72 46790.69 24426.922 53741.78  
## Sep 1995 40178.16 31111.50 51887.06 27087.467 59595.26  
## Oct 1995 44407.37 34386.35 57348.77 29938.733 65868.34  
## Nov 1995 71864.42 55647.40 92807.48 48449.831 106594.69  
## Dec 1995 156380.86 121091.75 201954.08 105429.448 231955.81  
## Jan 1996 22467.57 17336.40 29117.46 15065.329 33506.86  
## Feb 1996 29533.04 22788.25 38274.14 19802.984 44043.89  
## Mar 1996 30636.60 21648.24 43356.94 17936.708 52328.52  
## Apr 1996 35240.15 27191.96 45670.42 23629.808 52555.15  
## May 1996 36952.72 28513.41 47889.88 24778.151 55109.18  
## Jun 1996 39291.20 30317.82 50920.48 26346.183 58596.65  
## Jul 1996 47211.93 36429.60 61185.57 31657.322 70409.18  
## Aug 1996 47189.73 36412.48 61156.80 31642.439 70376.07  
## Sep 1996 52329.57 40378.47 67817.91 35088.887 78041.33  
## Oct 1996 57837.85 44628.77 74956.52 38782.394 86256.08  
## Nov 1996 93598.96 72222.70 121302.09 62761.521 139588.15  
## Dec 1996 203676.38 157160.50 263959.89 136572.460 303751.35

texasgas\_df <- (texasgas)  
plot(texasgas\_df$price, texasgas\_df$consumption , xlab = "Price", ylab = "Consumption")



##The data is not linear. Slope should be changed.

texasgas\_fit <- lm(consumption ~ exp(price), texasgas\_df)  
  
summary(texasgas\_fit)

##   
## Call:  
## lm(formula = consumption ~ exp(price), data = texasgas\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -35.86 -25.09 -13.86 20.64 65.14   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.086e+01 7.670e+00 9.238 2.98e-08 \*\*\*  
## exp(price) -1.642e-43 1.711e-43 -0.959 0.35   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 33.19 on 18 degrees of freedom  
## Multiple R-squared: 0.04864, Adjusted R-squared: -0.004214   
## F-statistic: 0.9203 on 1 and 18 DF, p-value: 0.3501

(summary(texasgas\_fit)$sigma)\*\*2

## [1] 1101.359

##Piecewise Linear Regression  
texasgas\_lin <- lm(consumption ~ price, texasgas\_df)  
library(segmented)  
segmented.mod <- segmented(texasgas\_lin, 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   
(summary(segmented.mod)$sigma)\*\*2

## [1] 167.8511

##Polynomial Regression  
texasgas\_poly <- lm(consumption ~ poly(price, 2), texasgas\_df)  
  
##Residual variance  
texasgas\_poly <- lm(consumption ~ poly(price, 2), texasgas\_df)

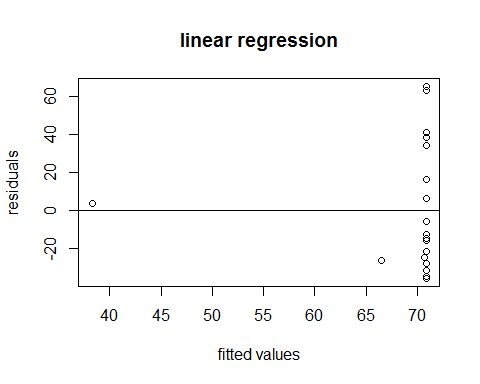
summary(texasgas\_fit)

##   
## Call:  
## lm(formula = consumption ~ exp(price), data = texasgas\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -35.86 -25.09 -13.86 20.64 65.14   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 7.086e+01 7.670e+00 9.238 2.98e-08 \*\*\*  
## exp(price) -1.642e-43 1.711e-43 -0.959 0.35   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 33.19 on 18 degrees of freedom  
## Multiple R-squared: 0.04864, Adjusted R-squared: -0.004214   
## F-statistic: 0.9203 on 1 and 18 DF, p-value: 0.3501

AIC(texasgas\_fit)

## [1] 200.7363

resiplot\_1 <- residuals(texasgas\_fit)  
plot(texasgas\_fit$fitted.values, resiplot\_1, ylab='residuals', xlab='fitted values',main='linear regression')  
abline(0,0)



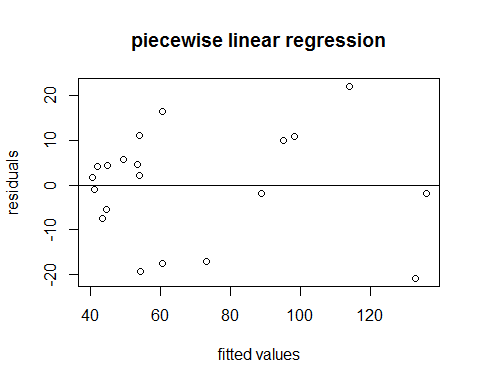
#piecewise linear regression  
  
summary(texasgas\_lin)

##   
## Call:  
## lm(formula = consumption ~ price, data = texasgas\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -40.625 -10.719 -1.136 14.073 38.292   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 138.561 13.552 10.225 6.34e-09 \*\*\*  
## price -1.104 0.202 -5.467 3.42e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 20.86 on 18 degrees of freedom  
## Multiple R-squared: 0.6241, Adjusted R-squared: 0.6033   
## F-statistic: 29.89 on 1 and 18 DF, p-value: 3.417e-05

AIC(texasgas\_lin)

## [1] 182.1631

resiplot\_2 <- residuals(segmented.mod)  
plot(segmented.mod$fitted.values, resiplot\_2, ylab='residuals', xlab='fitted values', main='piecewise linear regression')  
abline(0,0)



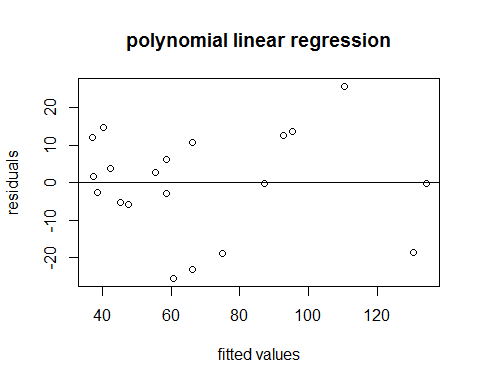
#polynomial regression.  
  
summary(texasgas\_poly)

##   
## Call:  
## lm(formula = consumption ~ poly(price, 2), data = texasgas\_df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -25.5601 -5.4693 0.7502 11.0252 25.6619   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 69.000 3.213 21.472 9.35e-14 \*\*\*  
## poly(price, 2)1 -114.043 14.371 -7.936 4.07e-07 \*\*\*  
## poly(price, 2)2 65.737 14.371 4.574 0.000269 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.37 on 17 degrees of freedom  
## Multiple R-squared: 0.8315, Adjusted R-squared: 0.8117   
## F-statistic: 41.95 on 2 and 17 DF, p-value: 2.666e-07

AIC(texasgas\_poly)

## [1] 168.1158

resiplot\_3 <- residuals(texasgas\_poly)  
plot(texasgas\_poly$fitted.values, resiplot\_3, ylab='residuals', xlab='fitted values',main='polynomial linear regression')  
abline(0,0)



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

texasgas\_new <- seq(min(new\_data), max(new\_data), length.out=5)  
intervals <- predict(segmented.mod, new\_data, interval="predict")  
intervals

## fit lwr upr  
## 1 104.53618 75.4174929 133.65487  
## 2 53.34514 23.2328452 83.45743  
## 3 47.19593 18.5085949 75.88327  
## 4 41.04673 10.7743347 71.31912  
## 5 34.89752 0.4430102 69.35203

plot(consumption ~ price, data = texasgas\_df, type = 'n')  
  
polygon(c(rev(texasgas\_new), texasgas\_new), c(rev(intervals[ ,3]), intervals[ ,2]), col = 'pink', border = NA)

