ASSIGNMENT- 1

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

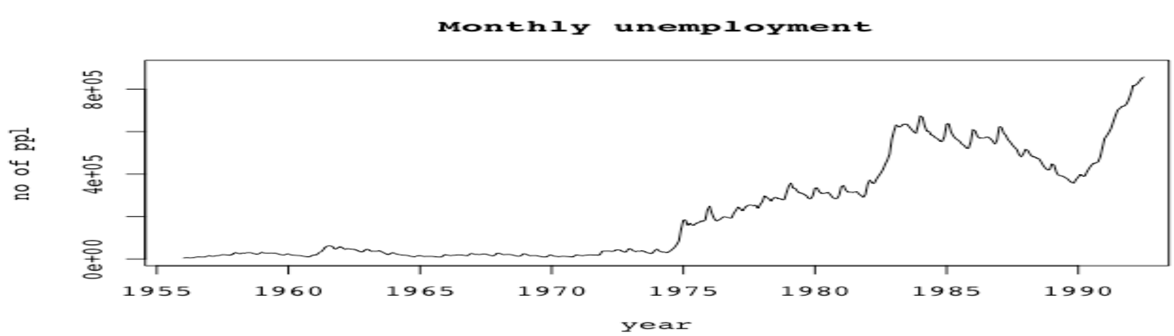
 a. Monthly total of people on unemployed benefits in Australia (January 1956–July 1992).

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

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

a. Below is the sample data for the Monthly total of people on unemployed benefits in Australia (January 1956–July 1992).

```
> dole
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1956  4742  6128  6494  5379  6011  7003  9164 10333  9614  9545 10096 13277
1957 15711 13135 13077 15453 15995 18071 20291 20175 18975 17928 19782
26055
1958 29856 26879 24485 27745 27282 29418 29908 29278 26002 23826 22302
27565
1959 31486 28207 27669 27559 27924 27528 27410 24887 21904 19598 19037
22469
1960 23781 20020 18177 17732 16765 16310 14897 12940 11465 10364 11738
17633
```



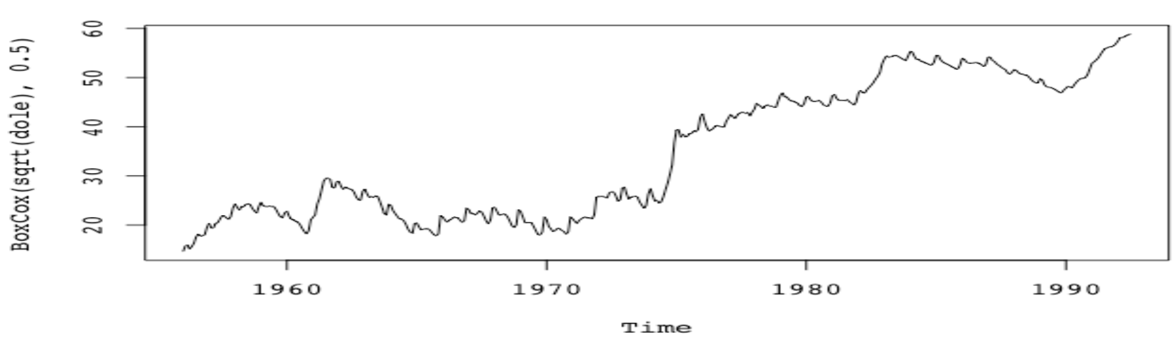

```
> plot(dole,main="Monthly unemployment",
ylab="close",ylim=c(4000,900000),xlab="year",xlim=c(1956,1992))
>
```

I plotted the data as is without any transformation and I can see that the no of people on unemployed benefits increases as the year passes by, as we can see that the from the year 1975 the increase is very steep.
Used few transformation to see if I can read the data in a better way, used log and sqrt transformation and I could see that with sqrt transformation helped me to clear plot the

graph and I could see that people of unemployed benefits increases as the year passes by.

```
plot(log(dole),main="Monthly unemployment",  ylab="log-no of ppl",ylim=c(5,20),xlab="year",xlim=c(1956,1992))

> plot(sqrt(dole),main="Monthly unemployment",  ylab="sqrt-no of ppl",xlab="year",xlim=c(1956,1992))
>
```



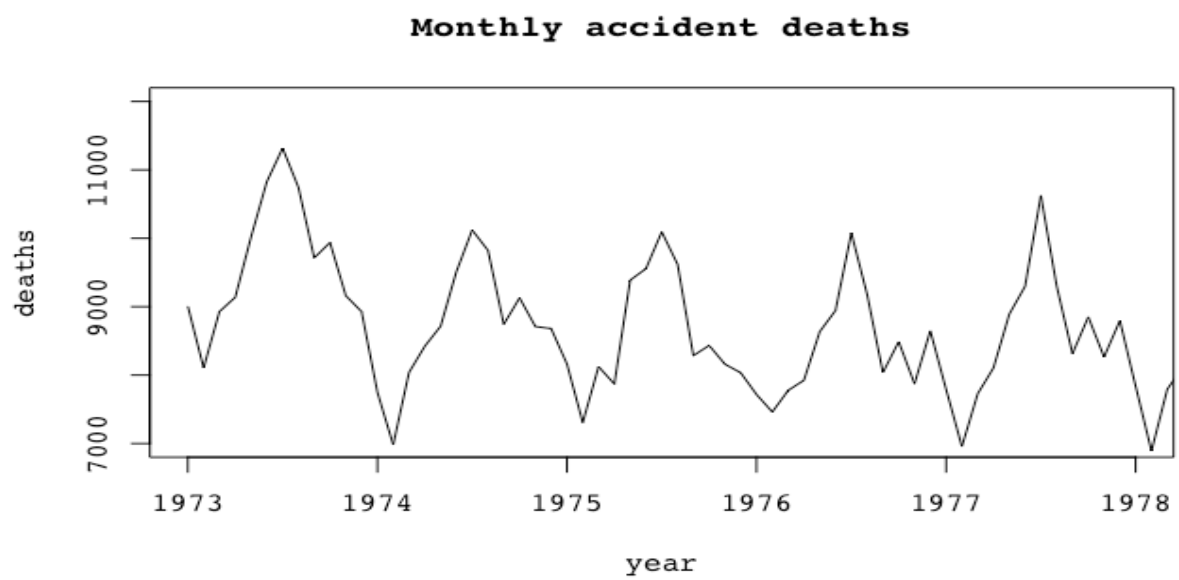
b. Monthly total of accidental deaths in the United States (January 1973–December 1978).

Sample data:

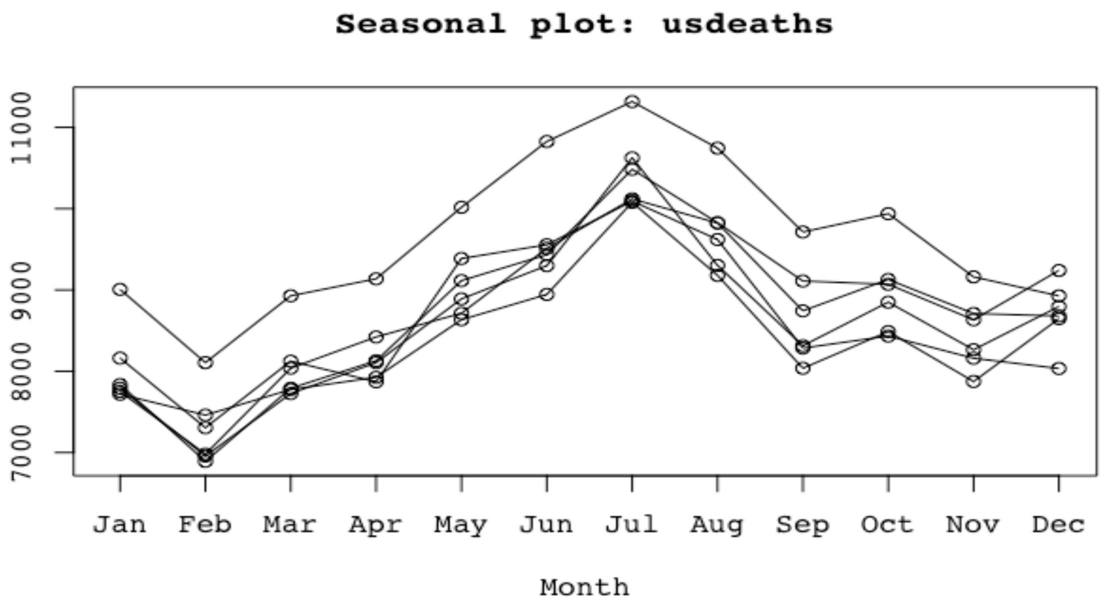
```
> usdeaths
      Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
1973  9007  8106  8928  9137 10017 10826 11317 10744  9713  9938  9161  8927
1974  7750  6981  8038  8422  8714  9512 10120  9823  8743  9129  8710  8680
1975  8162  7306  8124  7870  9387  9556 10093  9620  8285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

```
> plot(usdeaths,main="Monthly accident deaths",
ylab="deaths",ylim=c(7000,12000),xlab="year",xlim=c(1973,1978))
```

from the below graph we can see that the no of accidents deaths increases at the mid of the year and are less during the start and the end of the year.



From the below plot, we can see the maximum accident deaths in US are in the month of July, this may be because of the July 4th long weekend where people take lot of road trips.



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

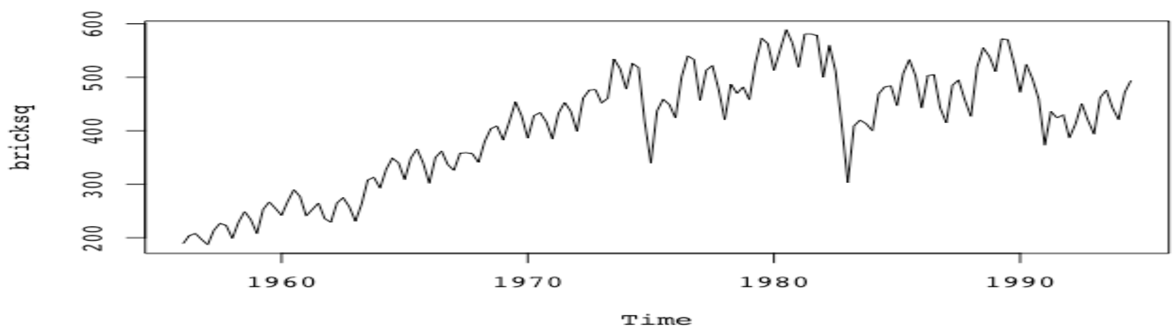
Sample data:

> bricksq

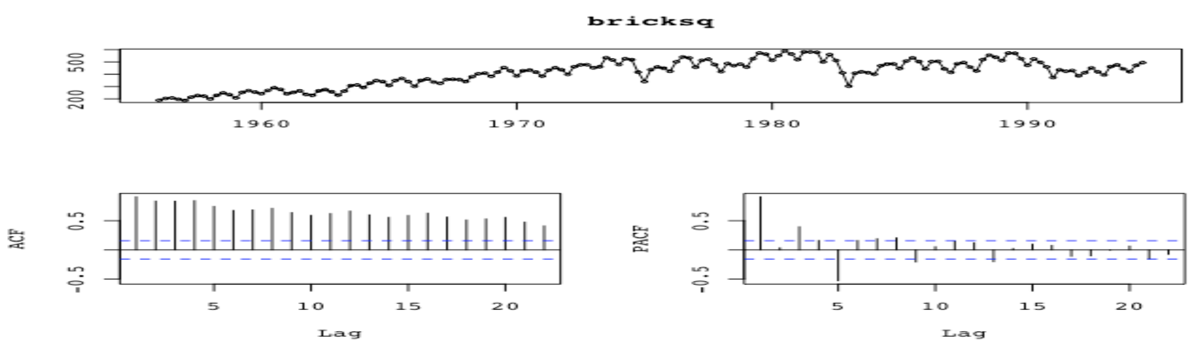
	Qtr1	Qtr2	Qtr3	Qtr4
1956	189	204	208	197
1957	187	214	227	223
1958	199	229	249	234
1959	208	253	267	255
1960	242	268	290	277
1961	241	253	265	236
1962	229	265	275	258
1963	231	263	308	313
1964	293	328	349	340
1965	309	349	366	340
1966	302	350	362	337
1967	326	358	359	357
1968	341	380	404	409

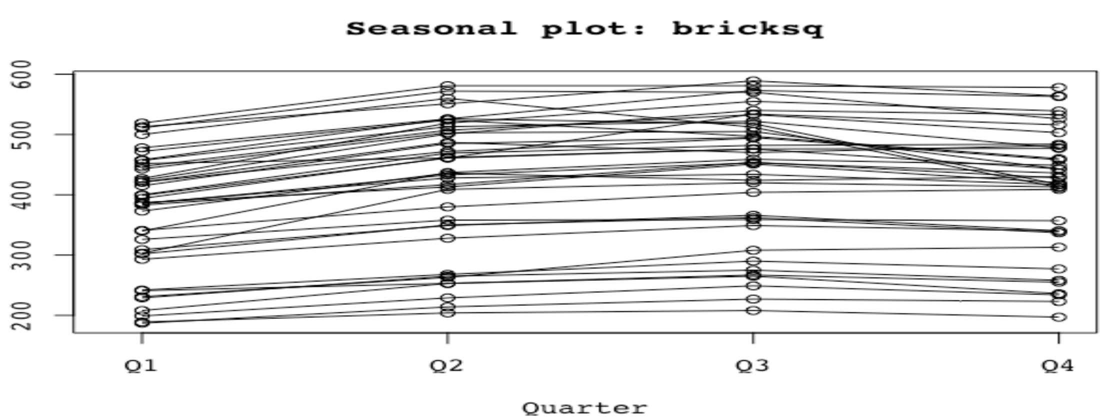
```
> plot(bricksq)
> tsdisplay(bricksq)
> seasonplot(bricksq)
```

From the plots we can see that the production of bricks is increasing over the years and there is a sudden drop in the production in the year 83 may be because of the maintenance of the plant.



Also we can see that the production is more in the Q2 and Q3 when compared to the other quarters.



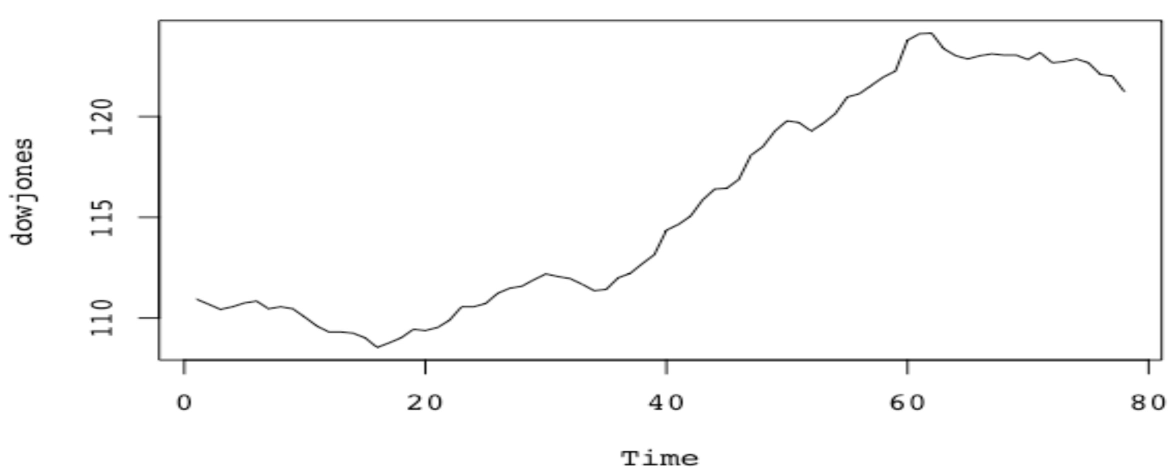


2. Use the Dow Jones index (data set dowjones) to do the following:

a. Produce a time plot of the series.
b. Produce forecasts using the drift method and plot them.
c. Show that the graphed forecasts are identical to extending the line drawn between the first and last observations.
d. Try some of the other benchmark functions to forecast the same data set. Which do you think is best? Why?

Below is the sample data:
> dowjones
Time Series:
Start = 1
End = 78
Frequency = 1
[1] 110.94 110.69 110.43 110.56 110.75 110.84 110.46 110.56 110.46 110.05 109.60 109.31 109.31 109.25
[15] 109.02 108.54 108.77 109.02 109.44 109.38 109.53 109.89 110.56 110.56 110.72 111.23 111.48 111.58
[29] 111.90 112.19 112.06 111.96 111.68 111.36 111.42 112.00 112.22 112.70 113.15 114.36 114.65 115.06

Below is the time plot of the series,

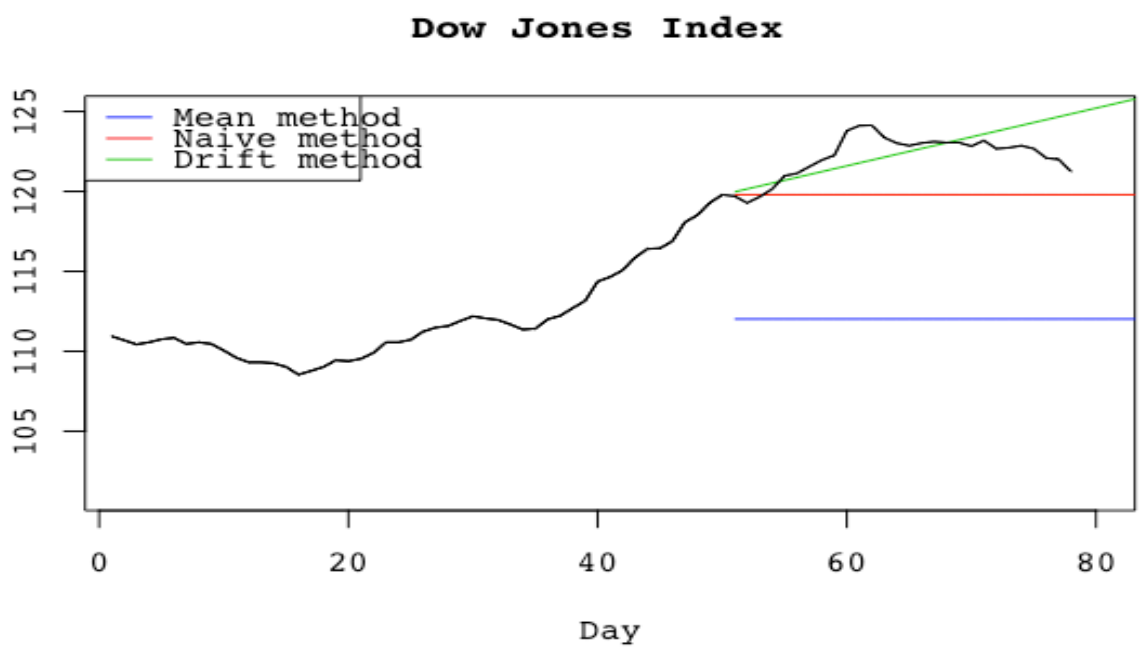


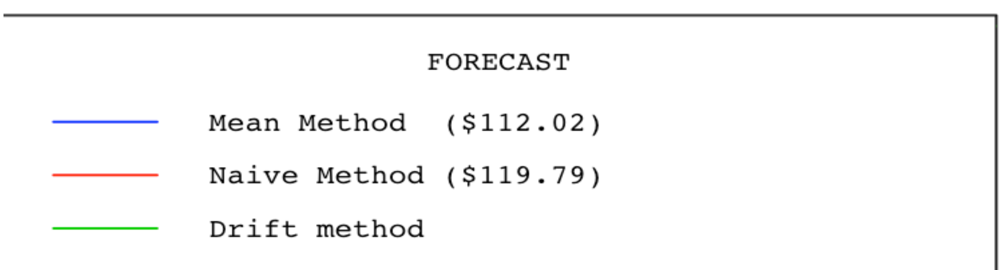
b/c/d combined:

Below I have used the drift/naïve and the mean method for forecasting:

Code for the following graph:

```
dj2 <- window(dowjones, end=50)
plot(dj2, main="Dow Jones Index ",
  ylab="", xlab="Day", xlim=c(2,290))
lines(meanf(dj2,h=42)$mean, col=4)
lines(rwf(dj2,h=42)$mean, col=2)
lines(rwf(dj2,drift=TRUE,h=42)$mean, col=3)
legend("topleft", lty=1, col=c(4,2,3),
  legend=c("Mean method","Naive method","Drift method"))
lines(dowjones)
```





After forecasting by three methods, below I am trying to find the accuracy of the three methods.

By going through the accuracy metrics I can see that the Drift method is more accurate here.

```
> dj3 <- window(dj, start=51)
> accuracy(meanf(dj2,h=42), dj3)
                       ME        RMSE         MAE         MPE     MAPE         MASE      ACF1 Theil's U
Training set 5.401062e-15    2.881174    2.213664 -0.06424737  1.95186     6.593893 0.9151357        NA
Test set     3.614697e+03 3615.026288 3614.696886 96.99369332 96.99369 10767.182213 0.8071042  215.3386
> accuracy(rwf(dj2,h=42), dj3)
                    ME      RMSE       MAE        MPE       MAPE     MASE      ACF1 Theil's U
Training set    0.1806122    0.4255393    0.3357143  0.1559374  0.2967809     1.00 0.5098258        NA
Test set     3606.9242857 3607.2543981 3606.9242857 96.7850934 96.7850934 10744.03 0.8071042  214.8758
> accuracy(rwf(dj2,drift=TRUE,h=42), dj3)
                       ME         RMSE          MAE         MPE       MAPE         MASE      ACF1 Theil's U
Training set 1.450036e-15    0.3853088    0.3055227 -0.00537228  0.2715387 9.100676e-01 0.5098258        NA
Test set     3.603041e+03 3603.3448572 3603.0411224 96.68158509 96.6815851 1.073246e+04 0.7995154  214.6405
> 
```

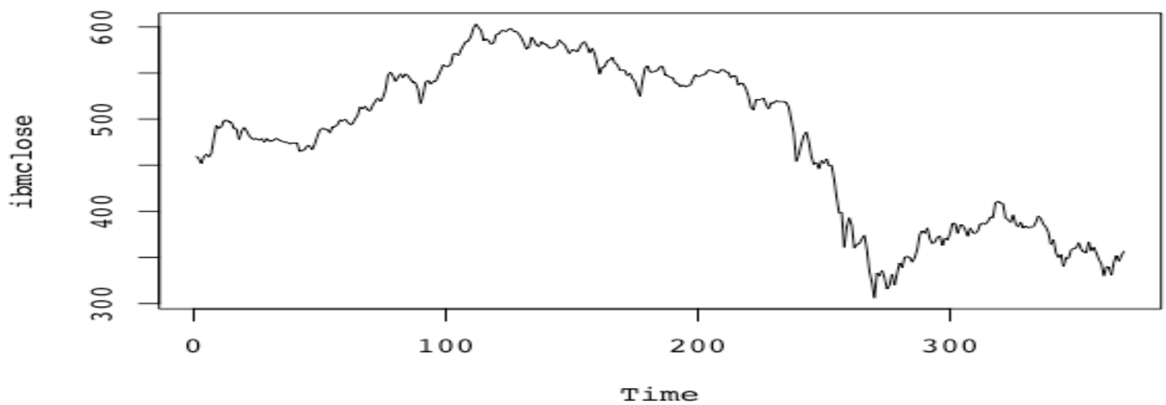
3. Consider the daily closing IBM stock prices (data set ibmclose).

a. Produce some plots of the data in order to become familiar with it.
b. Split the data into a training set of 300 observations and a test set of 69 observations.
c. Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

Below is the sample data:

Time Series:
Start = 1
End = 369
Frequency = 1
[1] 460 457 452 459 462 459 463 479 493 490 492 498 499 497 496 490 489 478 487 491 487 482 479 478 479
[26] 477 479 475 479 476 476 478 479 477 476 475 475 473 474 474 474 465 466 467 471 471 467 473 481 488
[51] 490 489 489 485 491 492 494 499 498 500 497 494 495 500 504 513 511 514 510 509 515 519 523 519 523

Just plotted the time series to get a gist of the data:



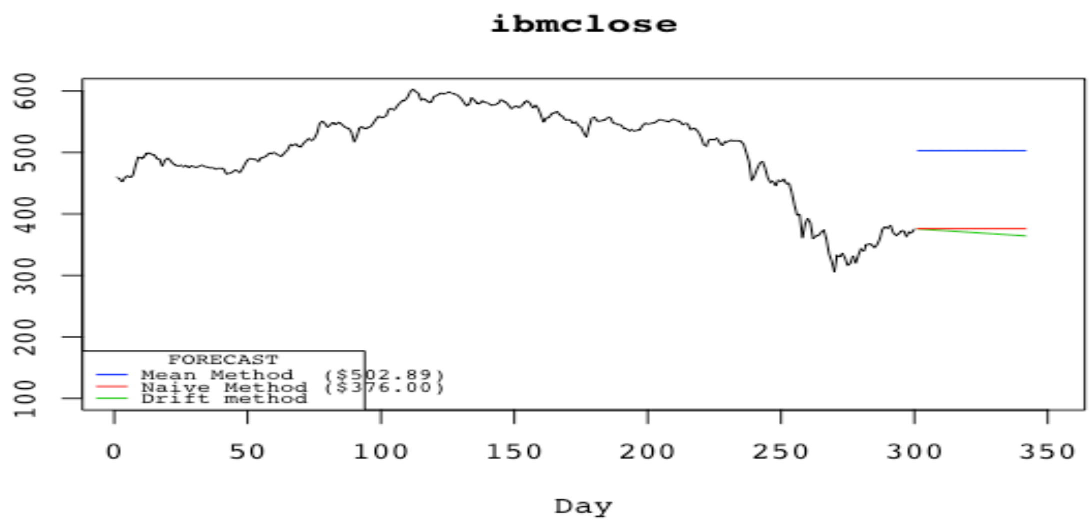
b. Training and test set:

```
> training_set <- window(ibmclose, end=300)
> training_set
Time Series:
Start = 1
End = 300
Frequency = 1
  [1] 460 457 452 459 462 459 463 479 493 490 492 498 499 497 496 490 489 478 487 491 487 482 479 478 479
 [26] 477 479 475 479 476 476 478 479 477 476 475 475 473 474 474 474 465 466 467 471 471 467 473 481 488
 [51] 490 489 489 485 491 492 494 499 498 500 497 494 495 500 504 513 511 514 510 509 515 519 523 519 523
 [76] 531 547 551 547 541 545 549 545 549 547 543 540 539 532 517 527 540 542 538 541 541 547 553 559 557
[101] 557 560 571 571 569 575 580 584 585 590 599 603 599 596 585 587 585 581 583 592 592 596 596 595 598
[126] 598 595 595 592 588 582 576 578 589 585 580 579 584 581 581 577 577 578 580 586 583 581 576 571 575
[151] 575 573 577 582 584 579 572 577 571 560 549 556 557 563 564 567 561 559 553 553 553 547 550 544 541
[176] 532 525 542 555 558 551 551 552 553 557 557 548 547 545 545 539 539 535 537 535 536 537 543 548 546
[201] 547 548 549 553 553 552 551 550 553 554 551 551 545 547 547 537 539 538 533 525 513 510 521 521 521
[226] 523 516 511 518 517 520 519 519 519 518 513 499 485 454 462 473 482 486 475 459 451 453 446 455 452
[251] 457 449 450 435 415 398 399 361 383 393 385 360 364 365 370 374 359 335 323 306 333 330 336 328 316
[276] 320 332 320 333 344 339 350 351 350 345 350 359 375 379 376 382 370 365 367 372 373 363 371 369 376
> test_set <- window(ibmclose,start=301)
> test_set
Time Series:
Start = 301
End = 369
Frequency = 1
 [1] 387 387 376 385 385 380 373 382 377 376 379 386 387 386 389 394 393 409 411 409 408 393 391 388 396 387
[27] 383 388 382 384 382 383 383 388 395 392 386 383 377 364 369 355 350 353 340 350 349 358 360 360 366 359
[53] 356 355 367 357 361 355 348 343 330 340 339 331 345 352 346 352 357
```

c.

```
> plot(training_set, main="ibmclose",
+     ylab="",ylim=c(101,600), xlab="Day", xlim=c(2,350))
> lines(rwf(training_set,drift=TRUE,h=42)$mean, col=3)
> lines(rwf(training_set,h=42)$mean, col=2)
```

```
> lines(meanf(training_set,h=42)$mean, col=4)
> legend("bottomleft",lty=1,col=c(4,2,3),cex=0.7, title="FORECAST",
+       legend=c(paste0("Mean Method  ($",
sprintf("%4.2f",max(mean(training_set))),")"),
+              paste0("Naive Method ($",
sprintf("%4.2f",max(rwf(training_set,h=60)$mean)),")"),"Drift method"))
> lines(test_set)
```



d. Testing the accuracy of the model, by using the test and training set:

```
accuracy(meanf(training_set,h=42), test_set)
accuracy(rwf(training_set,h=42), test_set)
accuracy(rwf(training_set,drift=TRUE,h=42), test_set)
```

```
> accuracy(meanf(training_set,h=42), test_set)
                        ME      RMSE       MAE        MPE     MAPE     MASE      ACF1 Theil's U
Training set  1.660438e-14  73.61532  58.72231  -2.642058 13.03019 11.52098 0.9895779        NA
Test set     -1.169886e+02 117.49988 116.98857 -30.420483 30.42048 22.95248 0.7296952   17.9457
> accuracy(rwf(training_set,h=42), test_set)
                     ME      RMSE      MAE         MPE     MAPE     MASE      ACF1 Theil's U
Training set -0.2809365  7.302815  5.09699 -0.08262872 1.115844 1.000000 0.1351052        NA
Test set      9.9047619 14.764823 11.95238  2.48806561 3.055375 2.344988 0.7296952  2.205498
> accuracy(rwf(training_set,drift=TRUE,h=42), test_set)
                      ME      RMSE       MAE         MPE     MAPE     MASE      ACF1 Theil's U
Training set 2.87048e-14  7.297409  5.127996 -0.02530123 1.121650 1.006083 0.1351052        NA
Test set     1.59449e+01 19.323121 16.468546  4.05898695 4.205572 3.231034 0.7596624  2.898499
>
```

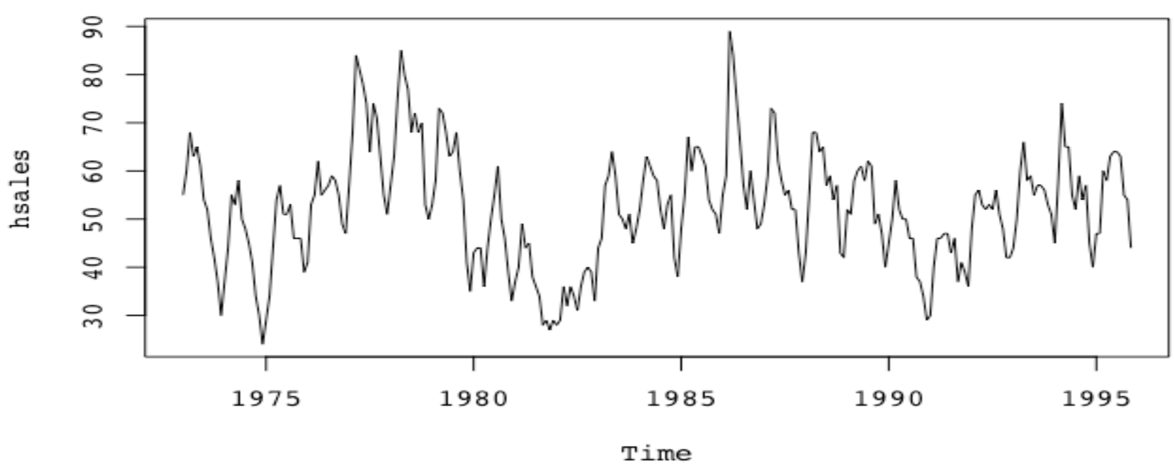
By comparing the above chart for RMSE, MAE and MASE, it looks like the naïve method is more efficient in this case.

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

a. Produce some plots of the data in order to become familiar with it.
b. Split the hsales data set into a training set and a test set, where the test set is the last two years of data.
c. Try various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?

Below is the sample data for the hsales data set:

```
> 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
```



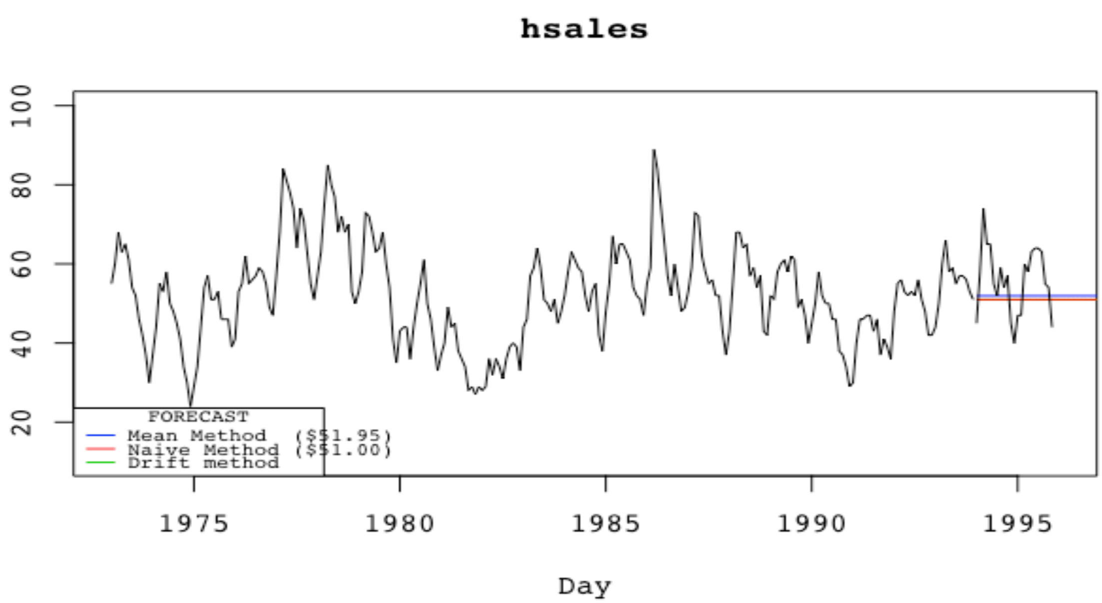
B. Converting the data into training set and test set:

```
> training_set <- window(hsales, start=1973, end=c(1993,12))
> tests_set <- window(hsales,start=1994)
> training_set
     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

> tests_set

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
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	

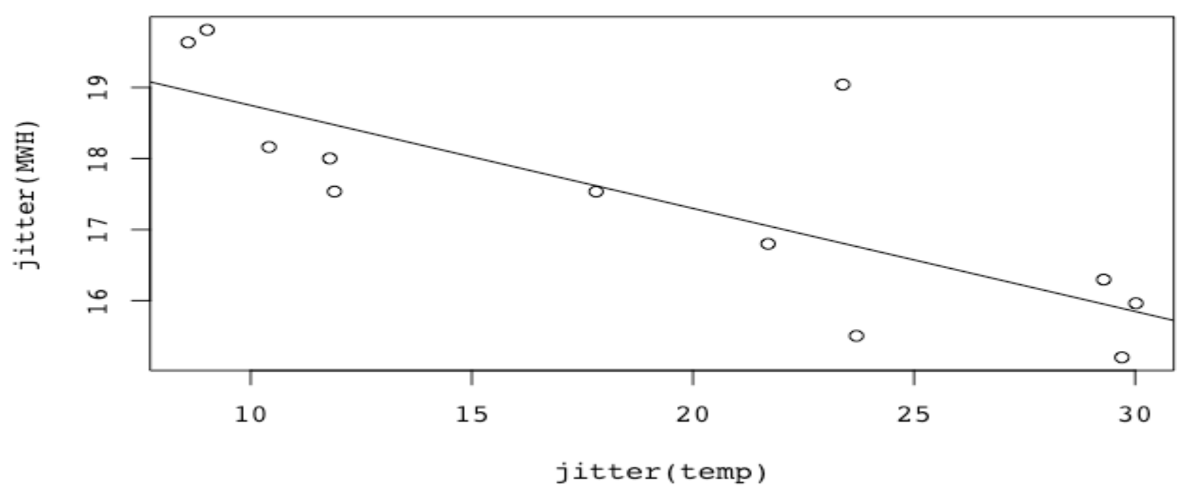


```
> accuracy(meanf(training_set,h=42), tests_set)
                       ME      RMSE      MAE       MPE     MAPE     MASE      ACF1 Theil's U
Training set 2.480763e-15 12.138802 9.498898 -6.120182 20.30851 1.119163 0.8661515        NA
Test set     4.051587e+00  9.216133 7.850759  5.074990 13.75973 0.924979 0.5095178   1.13105
> accuracy(rwf(training_set,h=42), tests_set)
                      ME     RMSE      MAE        MPE      MAPE      MASE      ACF1 Theil's U
Training set -0.01593625 6.289813 4.988048 -0.7800232  9.880157 0.5876934 0.1829708        NA
Test set      5.00000000 9.670664 8.304348  6.8080182 14.381673 0.9784210 0.5095178  1.179633
> accuracy(rwf(training_set,drift=TRUE,h=42), tests_set)
                       ME     RMSE      MAE        MPE     MAPE      MASE      ACF1 Theil's U
Training set 2.377739e-15 6.289793 4.987730 -0.7474544  9.87819 0.5876560 0.1829708        NA
Test set     5.023558e+00 9.683083 8.315434  6.8509188 14.39686 0.9797271 0.5094872  1.180966
>
```

By comparing the above chart for RMSE, MAE and MASE, it looks like the mean method is more efficient in this case.

4.1.

Electricity consumption was recorded for a small town on 12 randomly chosen days. The following maximum temperatures (degrees Celsius) and consumption (megawatt-hours) were recorded for each day.



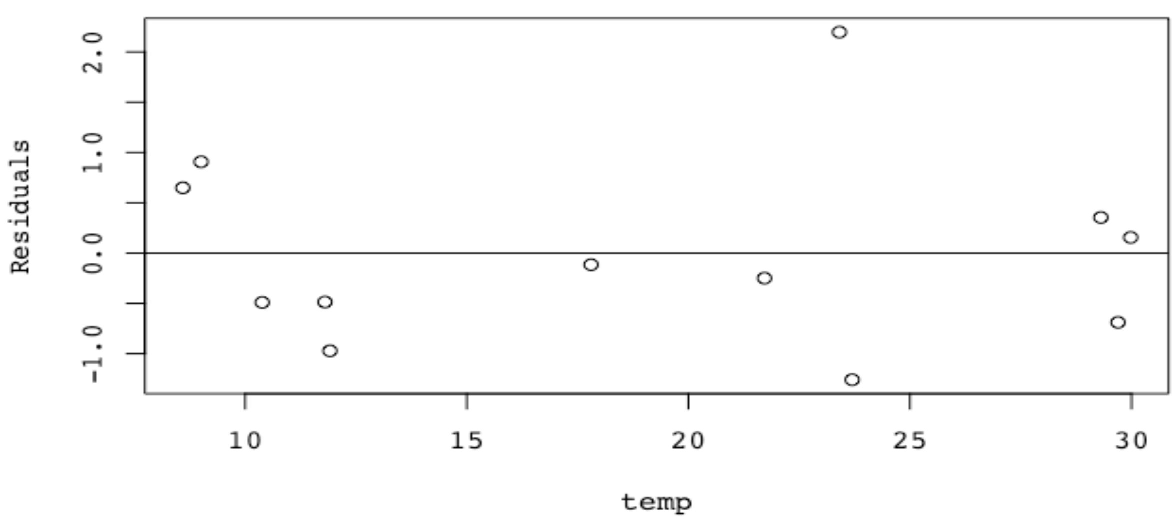
```
Residuals:
    Min      1Q  Median      3Q     Max 
-1.2593 -0.5395 -0.1827  0.4274  2.1972 

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
```

There is a negative relationship because as the temperature increases the consumption comes down.
Here the interpretation is 1-degree increase of temperature the consumption goes down by 0.14 mega watt

```
res<- residuals(fit)
> plot(jitter(res)~jitter(temp), ylab="Residuals", xlab="temp")
> abline(0,0)
```



we could see from the residual plot that the residual corresponding to the temp 10 to 25 are negative and positive after 25 and also we could see that apart from couple of observations there are no outliers. This plot looks ok and the model is fitted ok.

for 10 degrees:

The model is y = 20.19952 – 0.14516 * 10
=18.7

For 35 degrees:

The model is y = 20.19952- 0.14516 * 35
= 15.11892

Yes, the predictions looks reasonable.

D. Give prediction intervals for your forecasts.

The forecast is 18.74 and 15.11 respectively for 10 degrees and 35 degrees.

Below is the prediction interval:

	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**

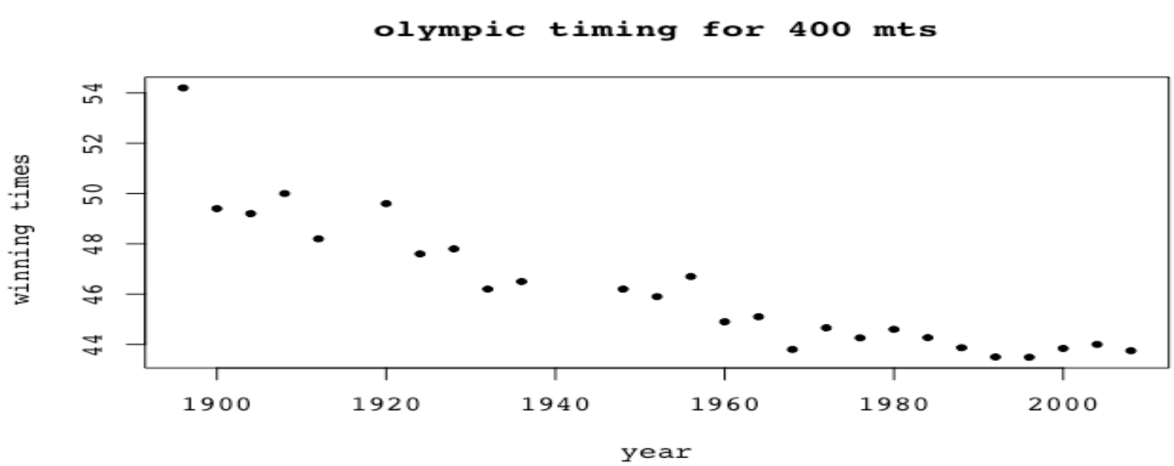
2. The following table gives the winning times (in seconds) for the men's 400 meters final in each Olympic Games from 1896 to 2012 (data set `olympic`).

a. The latest data is added to the time series, now we have observation till 2008, I used rbind to append the data.

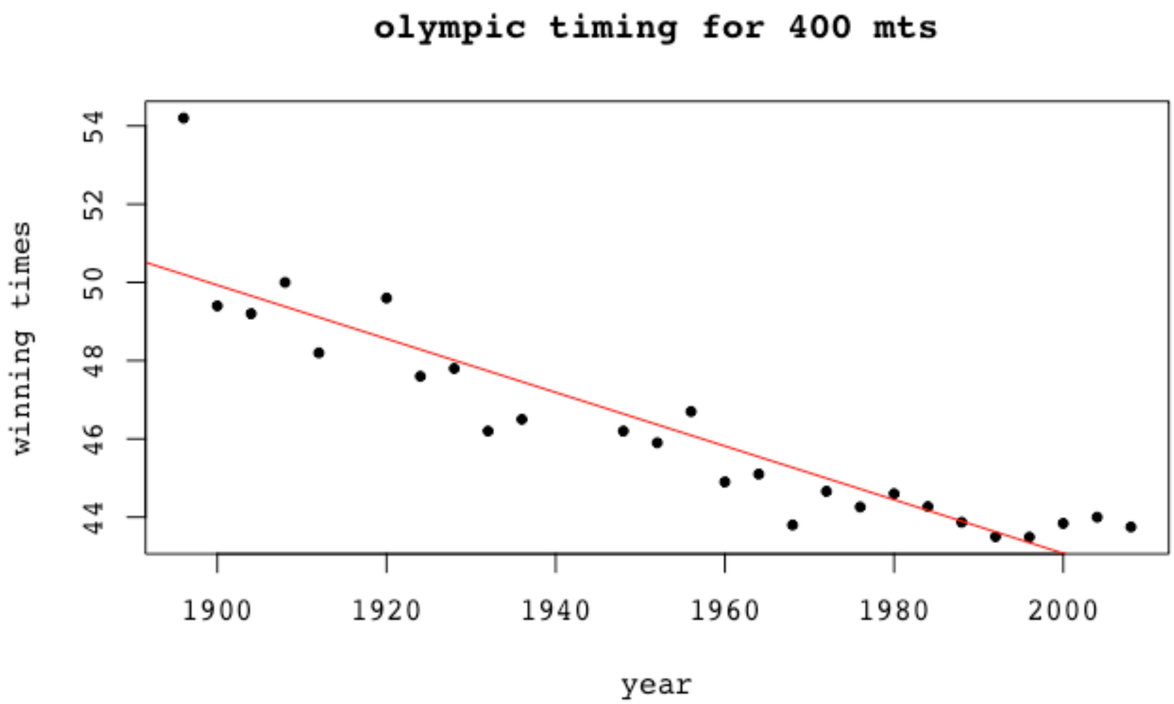
Below is the sample data:

```
> tail(olympic_final)
  Year  time
21 1988 43.87
22 1992 43.50
23 1996 43.49
24 2000 43.84
25 2004 44.00
26 2008 43.75
```

b. scatter plot:
```
> plot(Year, time, main="olympic timing for 400 mts",
     xlab="year", ylab="winning times ", pch=20)
```
From the plot we can see that as the year passed by the winning timings kept decreasing, this could be because of the advance training and the nutrition diet which the athletes go through nowadays.

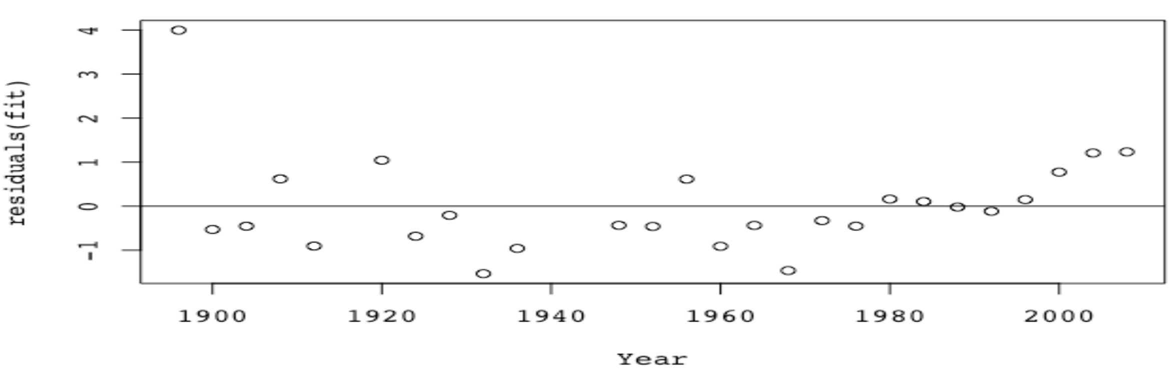


C. The average winning times are decreasing by 0.09 sec every year.



D.

> res<- residuals(fit)
> plot(residuals(fit)~Year)
> abline(time~Year)
> abline(0,0)
As we can see from the plot that the observations are pretty close to the line except for one of the outliers, the model has fitted ok.



e.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?

The model equation is below:

```
lm(formula = time ~ Year)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.5334 -0.5123 -0.2681  0.5004  3.9961 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 180.31698   12.52880   14.39 2.66e-13 ***
Year         -0.06863    0.00641  -10.71 1.28e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.126 on 24 degrees of freedom
Multiple R-squared:  0.8268,	Adjusted R-squared:  0.8196 
F-statistic: 114.6 on 1 and 24 DF,  p-value: 1.278e-10
```

y= 180.31698 -0.06863 * 2000
43.05698

y= 180.31698 -0.06863 * 2004
42.78246

y=180.31698 -0.06863 * 2008
42.50794

y=180.31698 -0.06863 * 2012
42.23342

above are the forecasting for the year 2000,2004,2008 and 2012 and below are the predictions intervals:

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
1	**43.06688**	**41.50589**	**44.62788**	**40.62216**	**45.51160**
2	**42.79238**	**41.22266**	**44.36210**	**40.33400**	**45.25077**
3	**42.51788**	**40.93876**	**44.09700**	**40.04478**	**44.99099**
4	**42.24338**	**40.65420**	**43.83257**	**39.75452**	**44.73225**

F.

By comparing the forecast and the actual values I can see that the forecast for years were not so accurate but the prediction intervals were good.

3

An elasticity coefficient is the ratio of the percentage change in the forecast variable (y) to the percentage change in the predictor variable (x). Mathematically, the elasticity is defined as (dy/dx)×(x/y)(dy/dx)×(x/y). Consider the log-log model,

$\log y=\beta_0+\beta_1 \log x+\varepsilon$.

Express y as a function of x and show that the coefficient β_1 is the elasticity coefficient.

In this model, the slope β_1 can be interpreted as an elasticity: β_1 is the average percentage change in y resulting from a 1% change in x

Y=B0 + B1*ln(X) + u --- 1% change in X is associated with a change in Y of 0.01*B1

The coefficients in a log-log model represent the elasticity of your Y variable with respect to our X variable. In other words, the coefficient is the estimated percent change in your dependent variable for a percent change in our independent variable.

Y= e ^β0+β1 log(x)+ e

After differentiating:
dy /dx1 = β1 x1 e β0+β1 log(x1)+ϱ = β1 y /x1

β1 = dy dx1/ x1 y