Homework 3 - Week 3

Question 7.1

Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

I am an Analyst at Mindtree Limited. A year back I worked on a project on a Proof of Concept to one of the clients, that involved the time series data. I use Python for analysis and used the Scikit Learn package in Python. In that project, I ask was to estimate the ATM cash demand forecasts for New York, Illinois, California. I created a model to predict the demand for the number of bills for each denomination. I tried different modeling techniques like ARIMA, SARIMA, and Exponential Smoothing (Holt-Winters). In the end, I used Holt-Winters, which gave a great result, to predict the demand forecast for a few weeks. The error rate of the prediction models was less than 15%. I used α of 0.7 giving importance/weightage to my recent value but at the same time not leaving the previous outputs.

```
Few of the KPI's(predictors) are as follows,

 1. Zipcode
 2. Time of loading the money
 3. Count of bills that was loaded (for each denomination)
 4. Time at which each type of bill was empty (for each denomination)
 5. Distance of ATMs from one and another in the same city
```

Question 7.2

Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file temps.txt), build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years. (Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

Note: in R, you can use either HoltWinters (simpler to use) or the smooth package's es function (harder to use, but more general). If you use es, the Holt-Winters model uses model="AAM" in the function call (the first and second constants are used "A"dditively, and the third (seasonality) is used "M"ultiplicatively; the documentation doesn't make that clear).

For this Question I had a reference from : [https://fukamilab.github.io/BIO202/09-A-time-series.html (https://fukamilab.github.io/BIO202/09-A-time-series.html)](https://fukamilab.github.io/BIO202/09-A-time-series.html)

In [270]:
```
# if (!require("fma")) install.packages("fma")
# if (!require("expsmooth")) install.packages("expsmooth")
# install.packages("forecast", repos='http://cran.us.r-project.org')
library(fma)
library(expsmooth)
suppressWarnings(suppressMessages(library("TTR")))
suppressWarnings(suppressMessages(library("forecast")))
```

In [271]:
```
temperature <- read.table("temps.txt", sep = '\t', header = TRUE, check.names = FALSE)
```

In [272]:
```
head(temperature)
```

DAY	1996	1997	1998	1999	2000	2001	2002	2003	2004	...	2006	2007	2008	2009	2010	2011	2012	2013	2014	2
1-Jul	98	86	91	84	89	84	90	73	82	...	93	95	85	95	87	92	105	82	90	
2-Jul	97	90	88	82	91	87	90	81	81	...	93	85	87	90	84	94	93	85	93	
3-Jul	97	93	91	87	93	87	87	87	86	...	93	82	91	89	83	95	99	76	87	
4-Jul	90	91	91	88	95	84	89	86	88	...	91	86	90	91	85	92	98	77	84	
5-Jul	89	84	91	90	96	86	93	80	90	...	90	88	88	80	88	90	100	83	86	
6-Jul	93	84	89	91	96	87	93	84	90	...	81	87	82	87	89	90	98	83	87	

In [273]:
```
dim(temperature)
```

123 21

In [274]:
```
# Flatten the data frame to a single vector (time series data of frequency = 1 day).

temp_vector <- as.vector(unlist(temperature[,2:21], recursive = TRUE, use.names = TRUE))
```

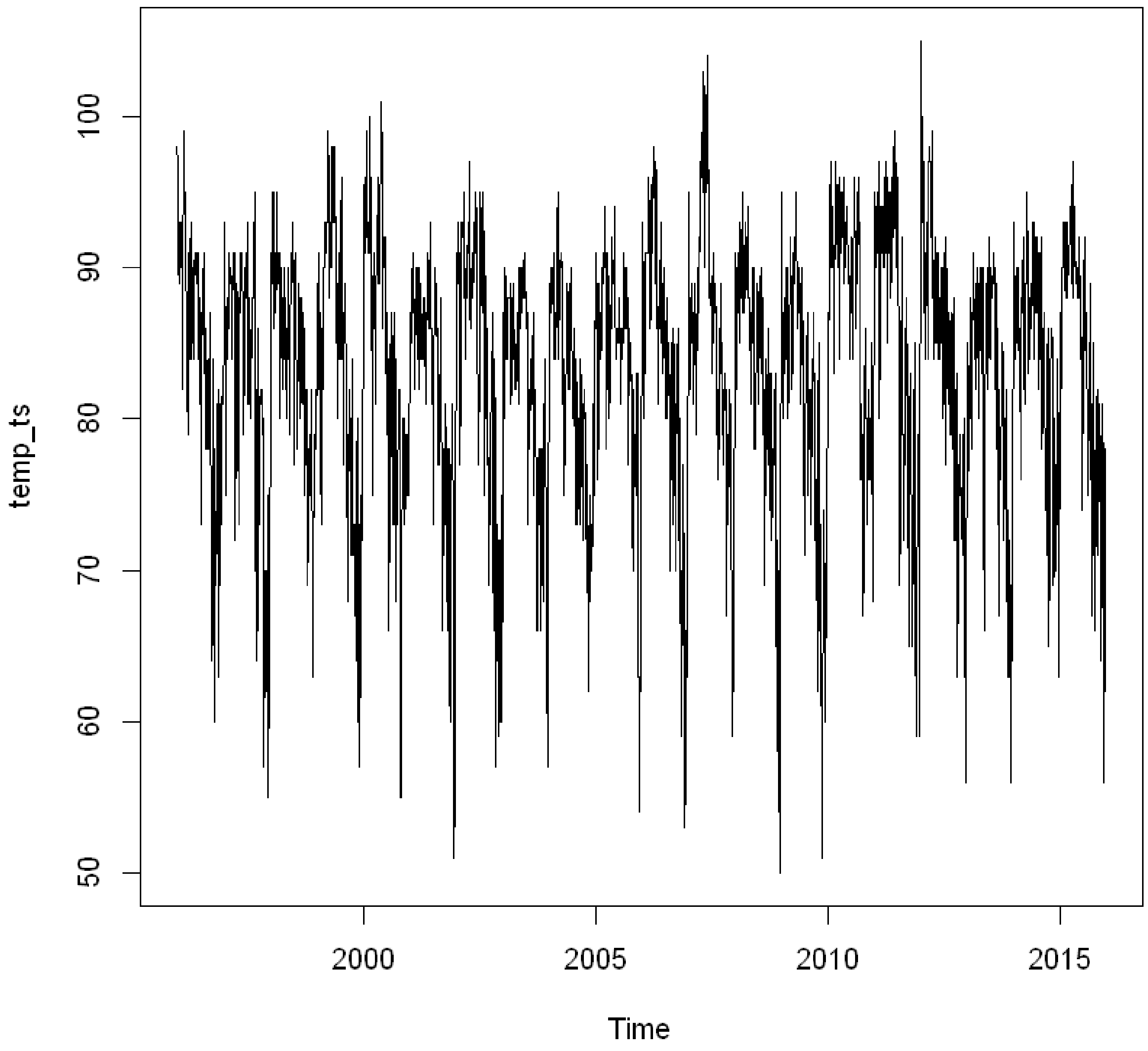
In [275]:
```
# There are 123 days of data. We want frequency of 1 day. So, deltat = 1/123

temp_ts <- ts(temp_vector, start=c(1996,1), end = c(2015,123), deltat = 1/123)
```

In [276]:
```
str(temp_ts)
```

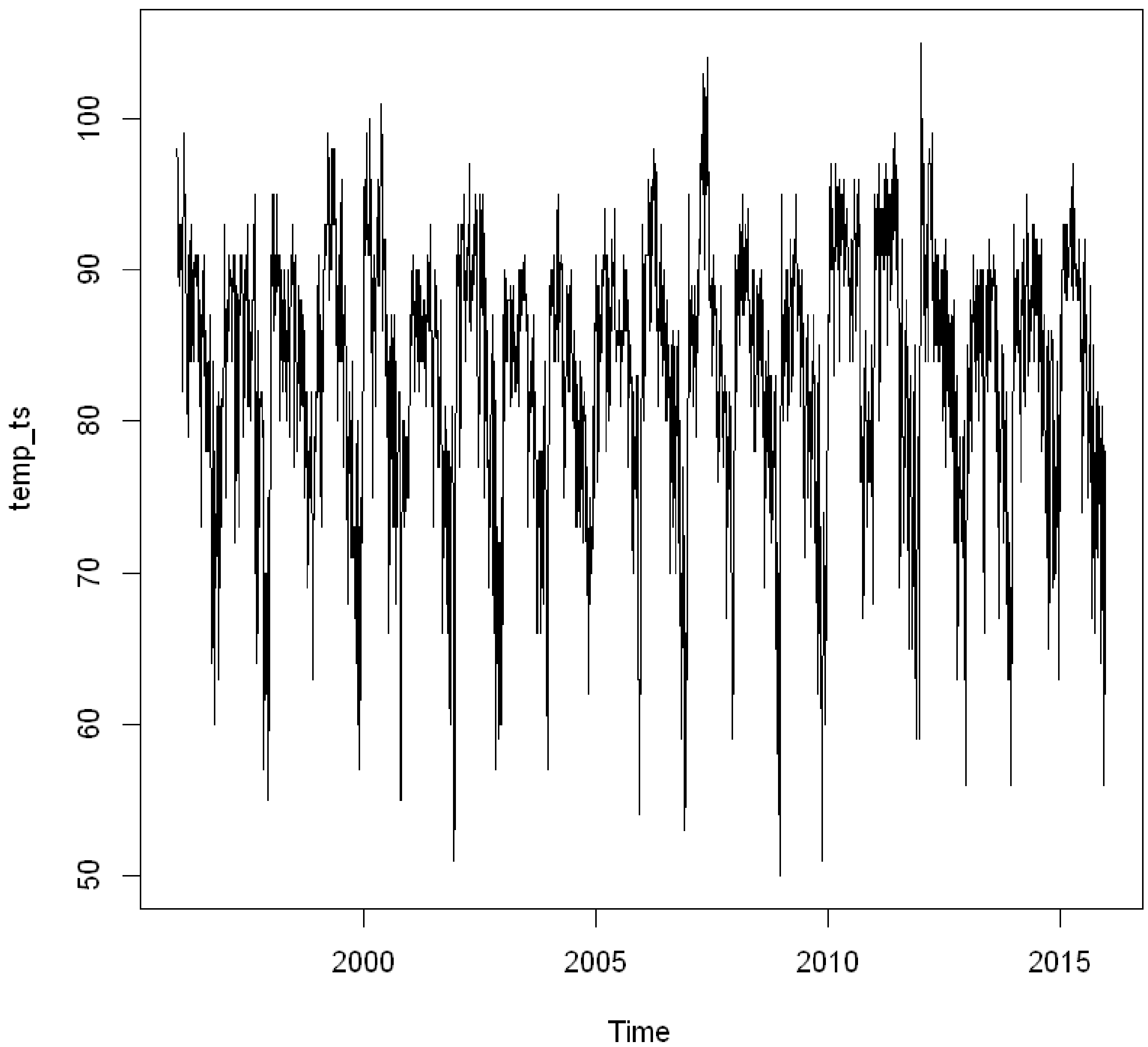
```
Time-Series [1:2460] from 1996 to 2016: 98 97 97 90 89 93 93 91 93 93 ...
```

Ploting the Time Series data across Years

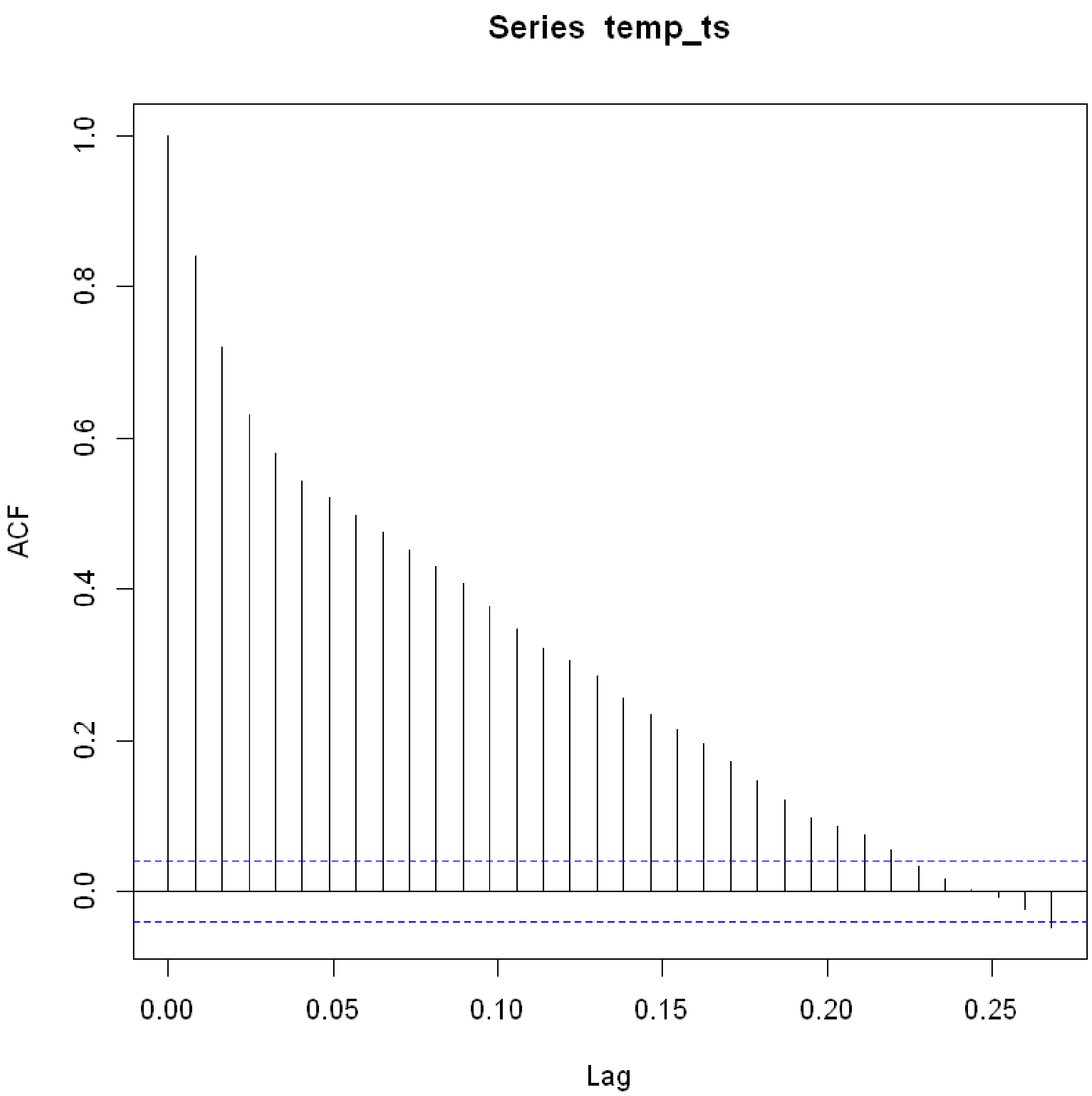
In [277]:
```
plot(temp_ts)
```



In [278]:
```
plot.ts(temp_ts)
```



In [145]:
```
# Auto Correlation plot showing that only one value lying outside the 95% limits and

acf(temp_ts, lag.max = NULL, type = "correlation",plot = TRUE)
```



Reference: https://www.statisticshowto.com/ljung-box-test/ (https://www.statisticshowto.com/ljung-box-test/)

In [146]:
```
# the L-jung box test has a p-value < 2.2e-16

Box.test(temp_ts, lag = 10, type = "Ljung-Box", fitdf = 0)
```

```
	Box-Ljung test

data:  temp_ts
X-squared = 8350.9, df = 10, p-value < 2.2e-16
```

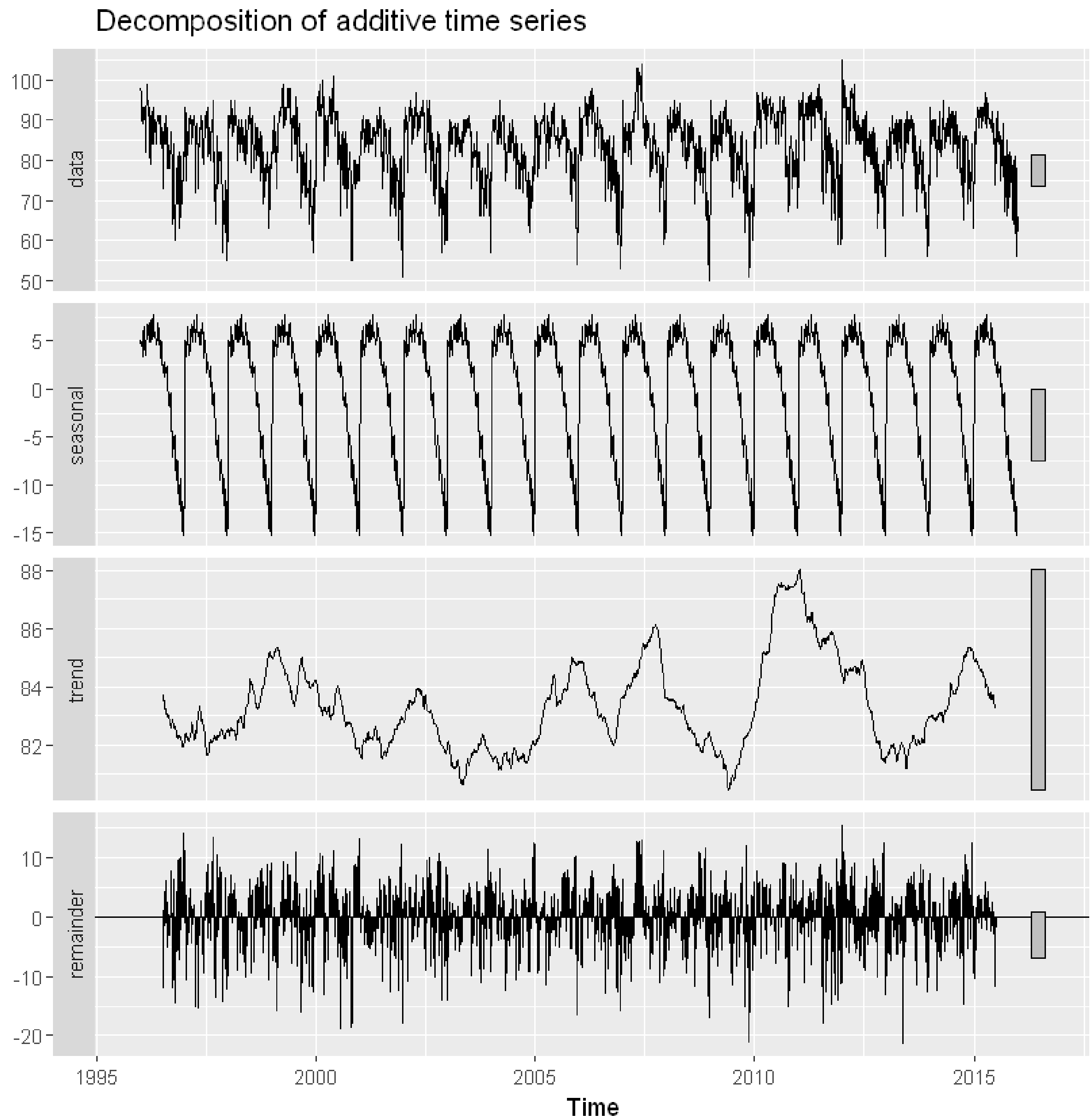
The null hypothesis of the Box Ljung Test, H0, is that our model does not show lack of fit (or in simple terms—the model is just fine). The alternate hypothesis, Ha, is just that the model does show a lack of fit.

Our p-value is less than 0.05 in this test which rejects the null hypothesis that the time series isn't autocorrelated.

Decomposing Time Series

Decomposing time series dismantles each sequence into its constituents–trend, irregular, and (if applicable) seaonsal components

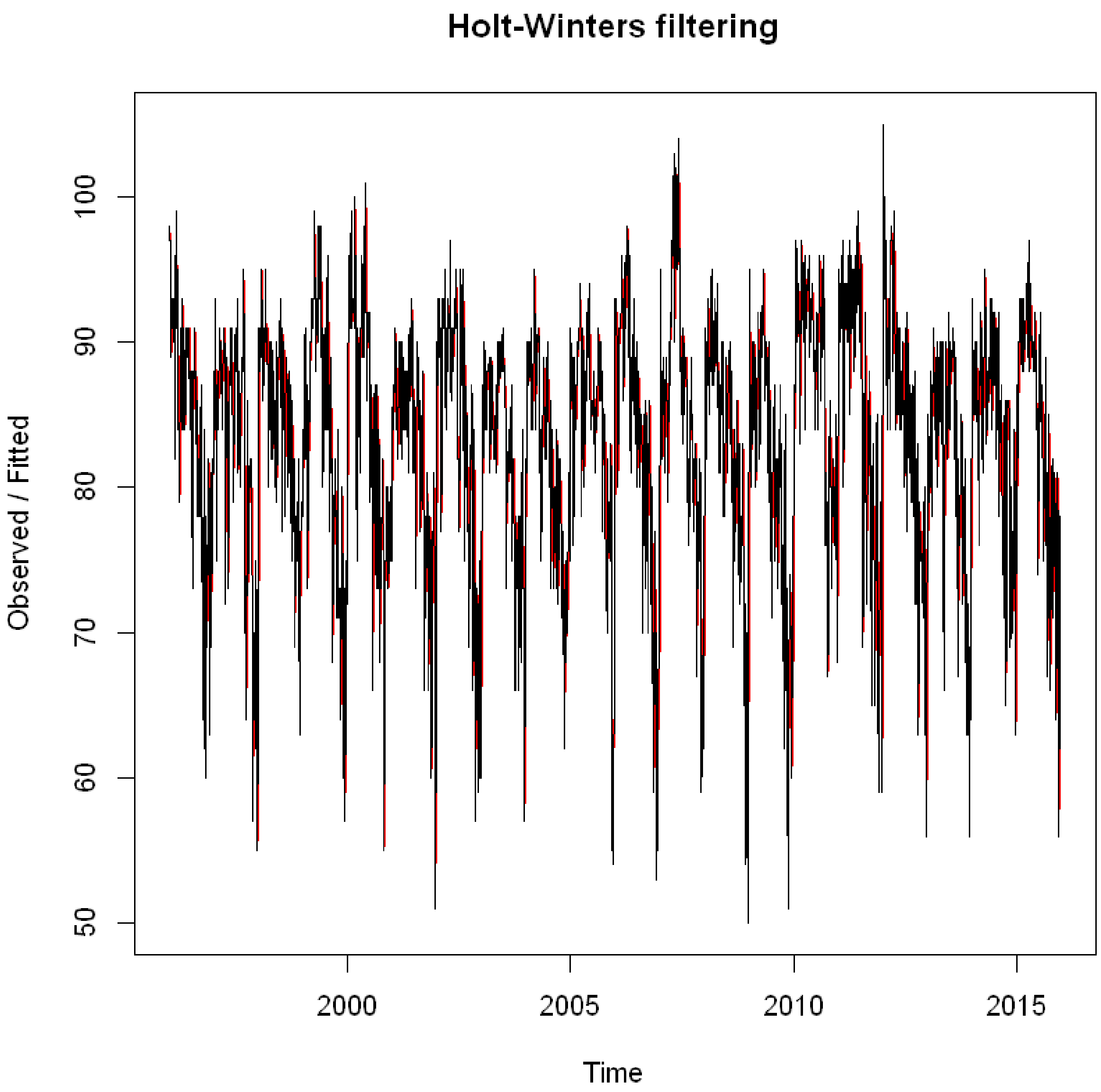
In [265]:
```
temp_components <- decompose(temp_ts)
# print(temp_components$seasonal)
autoplot(temp_components)
```



Now, I am going to perform single exponential smoothing. Here, I'm using a model with no trend and seasonality. I am going to let R determine the value of alpha.

In [191]:
```
temp_single_es <- HoltWinters(temp_ts, beta = FALSE, gamma = FALSE)
```

In [193]:
```
plot(temp_single_es)
```



In [194]:
```
temp_single_es
temp_single_es$SSE
```

```
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = temp_ts, beta = FALSE, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8388021
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 63.30952
```

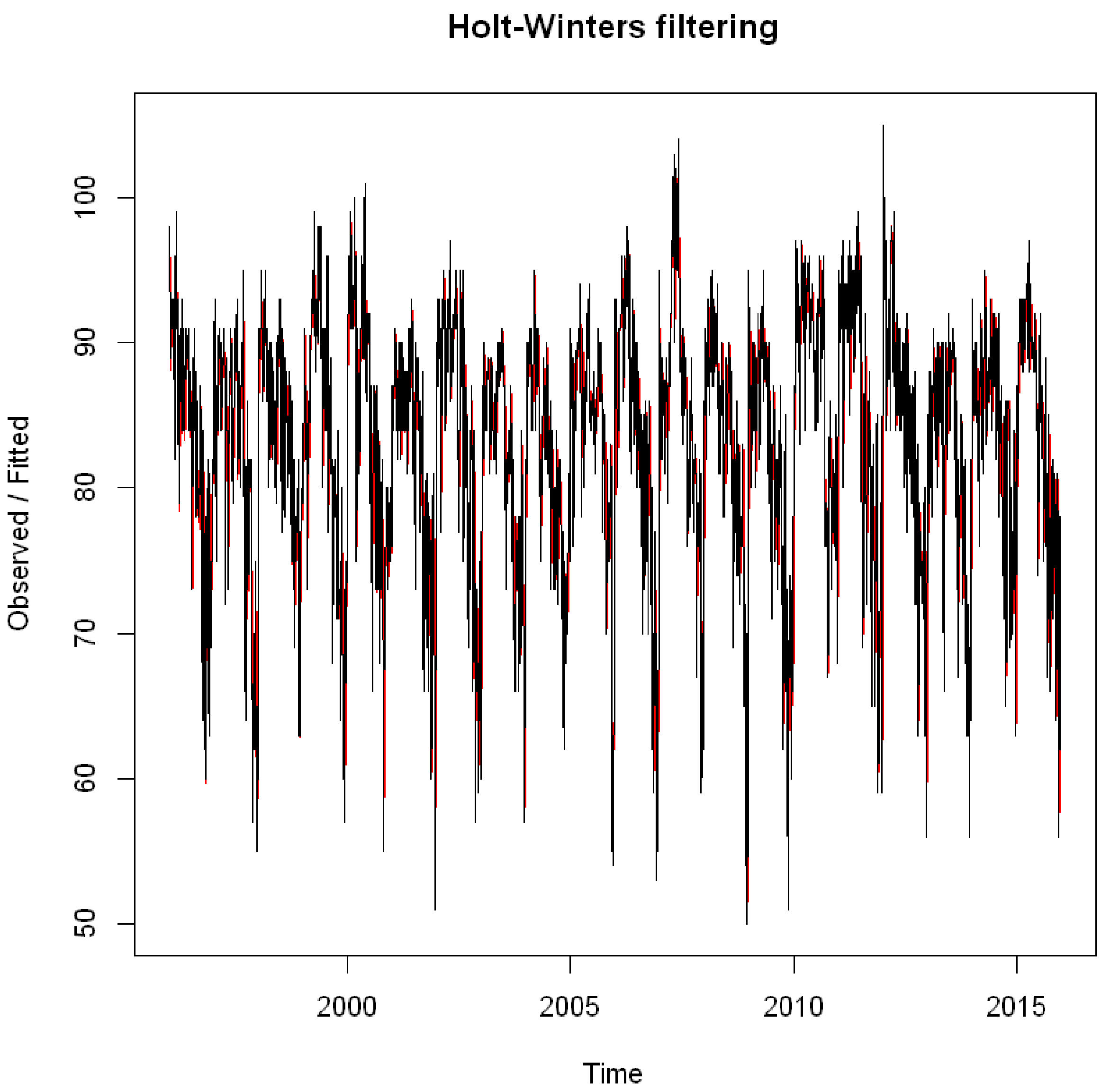
56198.0955314733

The estimated value of alpha is 0.8388021. This is high value indicating that the estimate of the current value of the

level is based mostly upon very recent observations in the time series. The value of the SSE for the in-sample forecast errors is 56198.0955314733.

I am going to perform double exponential smoothing (gamma = FALSE). I am going to let R determine the value of alpha.

In [176]:
```
temp_double_es <- HoltWinters(temp_ts, gamma = FALSE)
plot(temp_double_es)
```



In [177]:
```
temp_double_es
temp_double_es$SSE
```

```
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = temp_ts, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8445729
 beta : 0.003720884
 gamma: FALSE

Coefficients:
        [,1]
a 63.2530022
b -0.0729933
```

56572.5375681139

The estimated value of alpha is 0.8445729. Beta is 0.0037. This means that the trend value from the recent observations has relatively very little weight when forecasting for future values. The value of the sum-of-squared-errors for the in-sample forecast errors is 56572.5375681139.

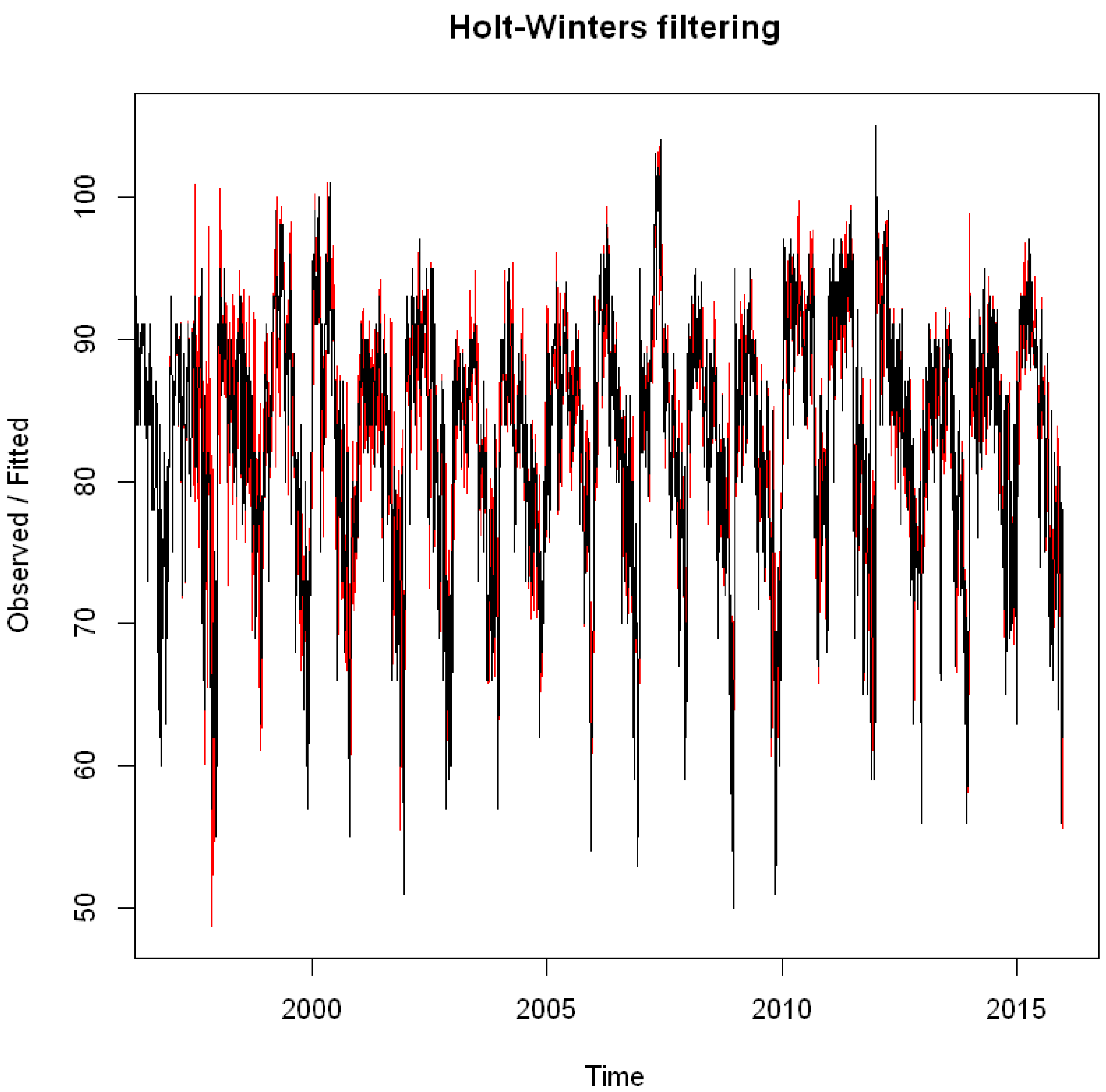
Next, I am going to check if the data can be described using an additive model. I am going to use Holt-Winters triple exponential smoothing to estimate the level (alpha), slope (beta) and seasonal (gamma) components.

In [266]:
```r
# additive model

temp_add_hw <- HoltWinters(temp_ts)
temp_add_hw
temp_add_hw$SSE
plot(temp_add_hw)
```

```
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = temp_ts)

Smoothing parameters:
 alpha: 0.6610618
 beta : 0
 gamma: 0.6248076

Coefficients:
              [,1]
a      71.477236414
b      -0.004362918
s1     18.590169842
s2     17.803098732
s3     12.204442890
s4     13.233948865
s5     12.957258705
s6     11.525341233
s7     10.854441534
s8     10.199632666
s9      8.694767348
s10     5.983076192
s11     3.123493477
s12     4.698228193
s13     2.730023168
s14     2.995935818
s15     1.714600919
s16     2.486701224
s17     6.382595268
s18     5.081837636
s19     7.571432660
s20     6.165047647
s21     9.560458487
s22     9.700133847
s23     8.808383245
s24     8.505505527
s25     7.406809208
s26     6.839204571
s27     6.368261304
s28     6.382080380
s29     4.552058253
s30     6.877476437
s31     4.823330209
s32     4.931885957
s33     7.109879628
s34     6.178469084
s35     4.886891317
s36     3.890547248
s37     2.148316257
s38     2.524866001
s39     3.008098232
s40     3.041663870
s41     2.251741386
s42     0.101091985
s43    -0.123337548
s44    -1.445675315
s45    -1.802768181
s46    -2.192036338
s47    -0.180954242
s48     1.538987281
s49     5.075394760
s50     6.740978049
s51     7.737089782
s52     8.579515859
s53     8.408834158
s54     4.704976718
s55     1.827215229
```

```
s56   -1.275747384
s57    1.389899699
s58    1.376842871
s59    0.509553410
s60    1.886439429
s61   -0.806454923
s62    5.221873550
s63    5.383073482
s64    4.265584552
s65    3.841481452
s66   -0.231239928
s67    0.542761270
s68    0.780131779
s69    1.096690727
s70    0.690525998
s71    2.301303414
s72    2.965913580
s73    4.393732595
s74    2.744547070
s75    1.035278911
s76    1.170709479
s77    2.796838283
s78    2.000312540
s79    0.007337449
s80   -1.203916069
s81    0.352397232
s82    0.675108103
s83   -3.169643942
s84   -1.913321175
s85   -1.647780450
s86   -5.281261301
s87   -5.126493027
s88   -2.637666754
s89   -2.342133004
s90   -3.281910970
s91   -4.242033198
s92   -2.596010530
s93   -7.821281290
s94   -8.814741200
s95   -8.996689798
s96   -7.835655534
s97   -5.749139155
s98   -5.196182693
s99   -8.623793296
s100 -11.809355220
s101 -13.129428554
s102 -16.095143067
s103 -15.125436350
s104 -13.963606549
s105 -12.953304848
s106 -16.097179844
s107 -15.489223470
s108 -13.680122300
s109 -11.921434142
s110 -12.035411347
s111 -12.837047727
s112  -9.095808127
s113  -5.433029341
s114  -6.800835107
s115  -8.413639598
s116 -10.912409484
s117 -13.553826535
s118 -10.652543677
s119 -12.627298331
s120  -9.906981556
s121 -12.668519900
s122  -9.805502547
s123  -7.775306633
```

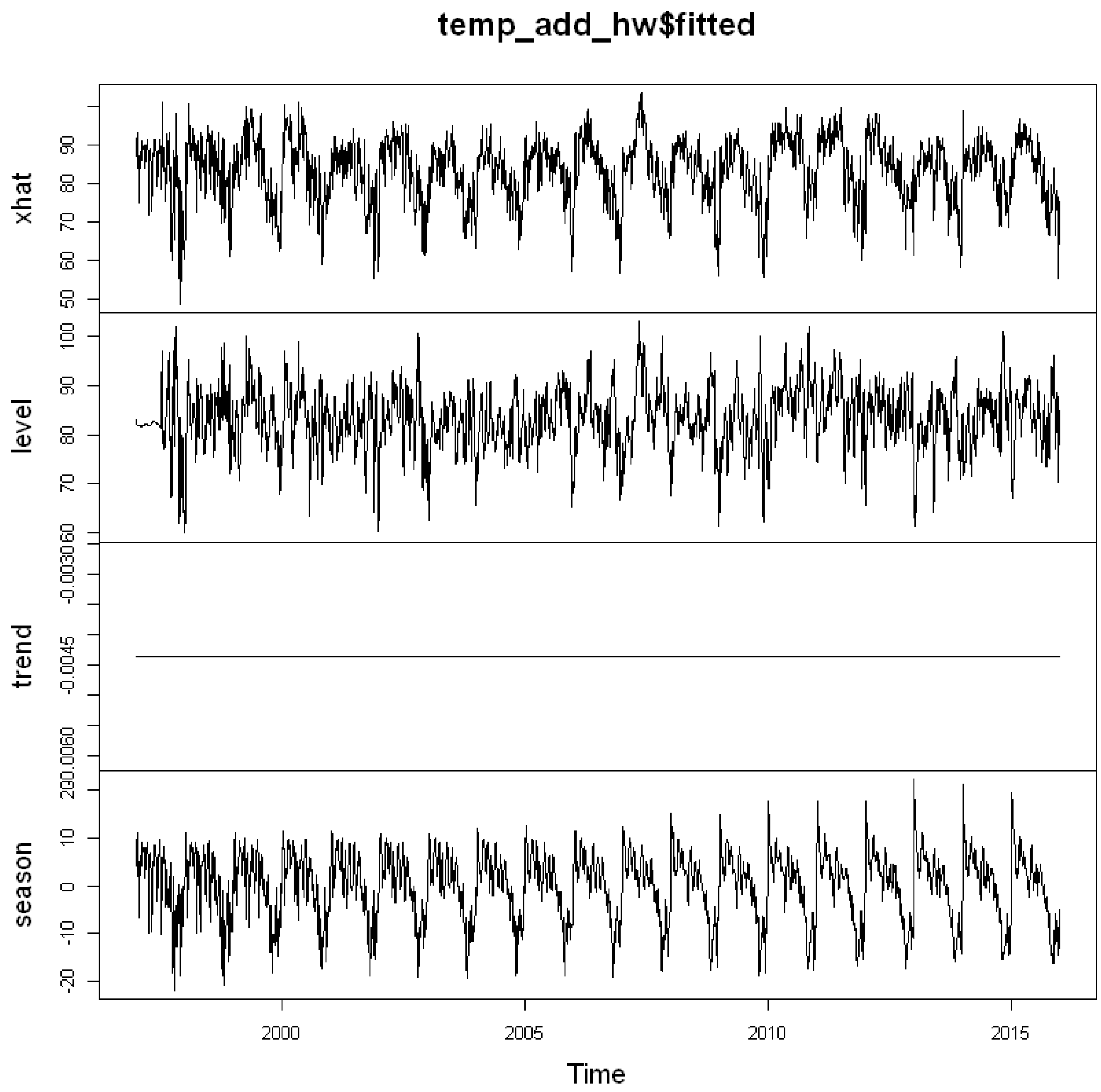
66244.2504058467



The value of beta is zero, suggesting no trend from recent observations on forecasting future values. The level parameter is 0.6610618, and the seasonal smoothing parameter, gamma is 0.6248076. SSE is 66244.2504058466.

The forecasts made by HoltWinters function are stored in a named element of this list variable called fitted.

In [267]:
```
# print(head(temp_add_hw$fitted,100))
plot(temp_add_hw$fitted)
```



I am going to check if the data can be described using an multiplicative model. I am going to use Holt-Winters triple

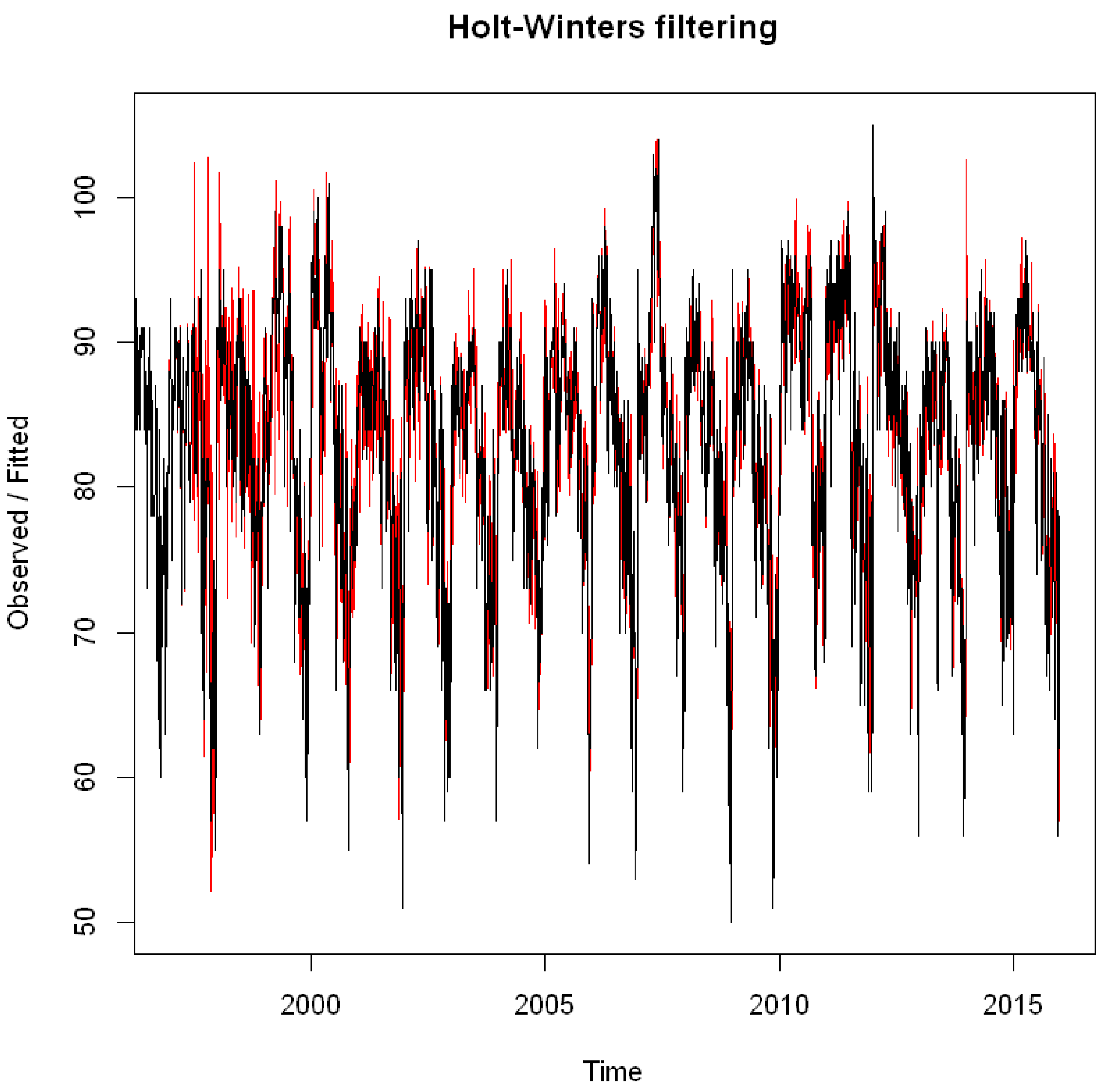
exponential smoothing to estimate the level (alpha), slope (beta) and seasonal (gamma) components.

In [268]:
```r
temp_mul_hw <- HoltWinters(temp_ts,alpha = NULL,beta = NULL,gamma = NULL, seasonal = "multiplicative"
temp_mul_hw
temp_mul_hw$SSE
plot(temp_mul_hw)
```

```
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = temp_ts, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = "multiplicative")

Smoothing parameters:
 alpha: 0.615003
 beta : 0
 gamma: 0.5495256

Coefficients:
            [,1]
a    73.679517064
b    -0.004362918
s1    1.239022317
s2    1.234344062
s3    1.159509551
s4    1.175247483
s5    1.171344196
s6    1.151038408
s7    1.139383104
s8    1.130484528
s9    1.110487514
s10   1.076242879
s11   1.041044609
s12   1.058139281
s13   1.032496529
s14   1.036257448
s15   1.019348815
s16   1.026754142
s17   1.071170378
s18   1.054819556
s19   1.084397734
s20   1.064605879
s21   1.109827336
s22   1.112670130
s23   1.103970506
s24   1.102771209
s25   1.091264692
s26   1.084518342
s27   1.077914660
s28   1.077696145
s29   1.053788854
s30   1.079454300
s31   1.053481186
s32   1.054023885
s33   1.078221405
s34   1.070145761
s35   1.054891375
s36   1.044587771
s37   1.023285461
s38   1.025836722
s39   1.031075732
s40   1.031419152
s41   1.021827552
s42   0.998177248
s43   0.996049257
s44   0.981570825
s45   0.976510542
s46   0.967977608
s47   0.985788411
s48   1.004748195
s49   1.050965934
s50   1.072515008
s51   1.086532279
s52   1.098357400
s53   1.097158461
s54   1.054827180
s55   1.022866587
s56   0.987259326
s57   1.016923524
```

s58	1.016604903
s59	1.004320951
s60	1.019102781
s61	0.983848662
s62	1.055888360
s63	1.056122844
s64	1.043478958
s65	1.039475693
s66	0.991019224
s67	1.001437488
s68	1.002221759
s69	1.003949213
s70	0.999566344
s71	1.018636837
s72	1.026490773
s73	1.042507768
s74	1.022500795
s75	1.002503740
s76	1.004560984
s77	1.025536556
s78	1.015357769
s79	0.992176558
s80	0.979377825
s81	0.998058079
s82	1.002553395
s83	0.955429116
s84	0.970970220
s85	0.975543504
s86	0.931515830
s87	0.926764603
s88	0.958565273
s89	0.963250387
s90	0.951644060
s91	0.937362688
s92	0.954257999
s93	0.892485444
s94	0.879537700
s95	0.879946892
s96	0.890633648
s97	0.917134959
s98	0.925991769
s99	0.884247686
s100	0.846648167
s101	0.833696369
s102	0.800001437
s103	0.807934782
s104	0.819343668
s105	0.828571029
s106	0.795608740
s107	0.796609993
s108	0.815503509
s109	0.830111282
s110	0.829086181
s111	0.818367239
s112	0.863958784
s113	0.912057203
s114	0.898308248
s115	0.878723779
s116	0.848971946
s117	0.813891909
s118	0.846821392
s119	0.819121827
s120	0.851036184
s121	0.820416491
s122	0.851581233
s123	0.874038407

68904.569331748



In [ ]:

Again, the value of beta is zero, suggesting no trend from recent observations on forecasting future values. The level parameter is 0.615003, and the seasonal smoothing parameter, gamma is 0.5495256. SSE is 68904.569331748.

I am going to write the fitted values to a csv file to perform CUSUM approach to detect unofficial end of summer.

In [211]:
```
df_temp1 <- matrix(temp_mul_hw$fitted[,4], nrow = 123) # taking season and performing CUSUM
```

In [214]:
```
# install.packages("xlsx", repos='http://cran.us.r-project.org')
suppressWarnings(suppressMessages(require(xlsx)))
```

In [245]:
```
write.csv(df_temp1, file = 'smoothed_temperature.csv')
```

Next, I am going to try to predict the temperatures for July 1 through Oct 31 for 2016 and 2017 using the Holt-Winters mulultiplicative model. To do this, I am using the predict() function that inputs the HW object, prediction interval, number of predictions, and confidence level.

In [244]:
```
predicted_temp <- predict(temp_mul_hw, n.ahead = 123*2, prediction.interval = FALSE, level = 0.95)
print(predicted_temp)
```

```
Time Series:
Start = c(2016, 1)
End = c(2017, 123)
Frequency = 123
             fit
  [1,] 91.28516
  [2,] 90.93510
  [3,] 85.41693
  [4,] 86.57116
  [5,] 86.27852
  [6,] 84.77782
  [7,] 83.91440
  [8,] 83.25410
  [9,] 81.77658
 [10,] 79.25010
 [11,] 76.65370
 [12,] 77.90779
 [13,] 76.01528
 [14,] 76.28765
```

In [236]:
```
new_df <- t(as.data.frame(matrix(round(predicted_temp),ncol =123,byrow = T)))
```

In [237]:
```
head(new_df)
```

V1	91	91
V2	91	90
V3	85	85
V4	87	86
V5	86	86
V6	85	84

In [238]:
```
# names(new_df) <- c('X2016', 'X2017')
temperature_data <- cbind(temperature, new_df)
```

In [240]:
```
head(temperature_data)
```

	DAY	1996	1997	1998	1999	2000	2001	2002	2003	2004	...	2008	2009	2010	2011	2012	2013	2014	2015	1
V1	1-Jul	98	86	91	84	89	84	90	73	82	...	85	95	87	92	105	82	90	85	91
V2	2-Jul	97	90	88	82	91	87	90	81	81	...	87	90	84	94	93	85	93	87	91
V3	3-Jul	97	93	91	87	93	87	87	87	86	...	91	89	83	95	99	76	87	79	85
V4	4-Jul	90	91	91	88	95	84	89	86	88	...	90	91	85	92	98	77	84	85	87
V5	5-Jul	89	84	91	90	96	86	93	80	90	...	88	80	88	90	100	83	86	84	86
V6	6-Jul	93	84	89	91	96	87	93	84	90	...	82	87	89	90	98	83	87	84	85

In [241]:
```
names(temperature_data) <- c('DAY','X1996','X1997','X1998','X1999','X2000','X2001','X2002','X2003','X
'X2005','X2006','X2007','X2008','X2009','X2010','X2011','X2012','X2013','X2014',
'X2015','X2016','X2017')
```

In [242]:
```
head(temperature_data)
```

	DAY	X1996	X1997	X1998	X1999	X2000	X2001	X2002	X2003	X2004	...	X2008	X2009	X2010	X2011	X2012	X201
V1	1-Jul	98	86	91	84	89	84	90	73	82	...	85	95	87	92	105	ε
V2	2-Jul	97	90	88	82	91	87	90	81	81	...	87	90	84	94	93	ε
V3	3-Jul	97	93	91	87	93	87	87	87	86	...	91	89	83	95	99	7
V4	4-Jul	90	91	91	88	95	84	89	86	88	...	90	91	85	92	98	7
V5	5-Jul	89	84	91	90	96	86	93	80	90	...	88	80	88	90	100	ε
V6	6-Jul	93	84	89	91	96	87	93	84	90	...	82	87	89	90	98	ε

In this problem, I have used Holt Winters approach to exponentially smoothe data, and use the smoothed data to predict unofficial end of summer using CUSUM approach. In addition, I have also used the smoothed data to predict temperatures of each day (July1 to Oct 31) for the next two years.

CUSUM APPROCH

C	Threshold
0	0.01

mu ----->	1.000	0.998	0.998	0.998	0.997	0.996	0.996	0.996	0.996	0.995	0.995	0.995	0.994	0.994	0.994	0.993	0.993	0.994	0.993
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19
1	1.0527	1.0495	1.1206	1.1033	1.1184	1.1082	1.1409	1.1406	1.1254	1.1221	1.1614	1.1981	1.1989	1.2430	1.2438	1.2384	1.3002	1.2906	1.2545
2	1.1007	1.0997	1.1080	1.0983	1.1102	1.1162	1.1268	1.1541	1.1422	1.1319	1.1445	1.1347	1.1534	1.1654	1.1729	1.1907	1.1920	1.2192	1.2288
3	1.1354	1.1354	1.1391	1.1428	1.1432	1.1385	1.1297	1.1561	1.1657	1.1480	1.1495	1.1358	1.1533	1.1552	1.1573	1.1698	1.1899	1.1723	1.1690
4	1.1103	1.1105	1.1171	1.1258	1.1345	1.1261	1.1308	1.1377	1.1506	1.1470	1.1425	1.1502	1.1512	1.1578	1.1638	1.1593	1.1666	1.1680	1.1590
5	1.0252	1.0252	1.0447	1.0673	1.0847	1.0972	1.1151	1.1039	1.1208	1.1337	1.1322	1.1427	1.1392	1.1129	1.1324	1.1320	1.1452	1.1682	1.1704
6	1.0258	1.0257	1.0282	1.0423	1.0540	1.0675	1.0802	1.0943	1.1027	1.0922	1.0758	1.0885	1.0822	1.1031	1.1151	1.1186	1.1216	1.1350	1.1455
7	0.9166	0.9164	0.9470	0.9476	0.9704	0.9918	1.0025	1.0303	1.0439	1.0353	1.0383	1.0378	1.0611	1.0706	1.0939	1.1096	1.1003	1.0983	1.1150
8	1.0635	1.0634	1.0479	1.0362	1.0140	1.0179	1.0165	1.0313	1.0313	1.0621	1.0631	1.0576	1.0673	1.0589	1.0769	1.0863	1.0926	1.1180	1.1229
9	1.0270	1.0269	1.0286	1.0307	1.0349	1.0368	1.0347	1.0341	1.0360	1.0490	1.0575	1.0747	1.0723	1.0734	1.0749	1.0716	1.0817	1.0934	1.1004
10	1.0644	1.0642	1.0470	1.0439	1.0495	1.0371	1.0387	1.0248	1.0299	1.0072	1.0189	1.0258	1.0301	1.0392	1.0327	1.0426	1.0478	1.0566	1.0593
11	1.0281	1.0279	1.0290	1.0187	1.0194	1.0283	1.0132	1.0133	1.0200	1.0267	1.0419	1.0371	1.0414	1.0477	1.0506	1.0577	1.0443	1.0298	1.0313
12	1.0775	1.0773	1.0539	1.0302	1.0211	1.0241	1.0000	1.0078	1.0092	1.0215	1.0307	1.0363	1.0476	1.0588	1.0502	1.0530	1.0333	1.0482	1.0575
13	1.0532	1.0531	1.0543	1.0328	1.0239	1.0146	1.0207	1.0218	1.0242	1.0206	1.0243	1.0253	1.0123	1.0090	1.0184	1.0305	1.0426	1.0221	1.0286
14	1.1012	1.1014	1.0887	1.0965	1.0840	1.0558	1.0651	1.0479	1.0432	1.0455	1.0417	1.0331	1.0340	1.0407	1.0376	1.0216	1.0280	1.0418	1.0426
15	1.1125	1.1128	1.1160	1.1132	1.1047	1.0888	1.0931	1.0856	1.0569	1.0470	1.0452	1.0330	1.0358	1.0380	1.0445	1.0102	1.0150	1.0254	1.0193
16	1.1123	1.1125	1.1032	1.1187	1.1142	1.1116	1.1135	1.1117	1.0944	1.0850	1.0761	1.0782	1.0696	1.0657	1.0499	1.0485	1.0471	1.0468	1.0319
17	1.0872	1.0874	1.0969	1.0930	1.0917	1.1035	1.1059	1.1068	1.1011	1.1067	1.1037	1.1066	1.0944	1.0823	1.0672	1.0682	1.0659	1.0719	1.0686
18	1.0864	1.0866	1.0930	1.1059	1.1175	1.1193	1.1169	1.1149	1.1193	1.1202	1.1162	1.1147	1.1135	1.0877	1.0697	1.0747	1.0724	1.0631	1.0530
19	1.0853	1.0856	1.0981	1.1046	1.1164	1.1226	1.1210	1.1194	1.1118	1.1111	1.1189	1.1151	1.1172	1.1003	1.1093	1.1155	1.1006	1.0988	1.0954
20	1.0962	1.0966	1.0891	1.0986	1.1020	1.0995	1.0987	1.1027	1.1141	1.1134	1.1084	1.1038	1.1145	1.1143	1.1174	1.1104	1.0867	1.0784	1.0556
21	1.0823	1.0828	1.0829	1.0878	1.0769	1.0732	1.0881	1.0944	1.1031	1.1077	1.1070	1.1055	1.1117	1.1234	1.1304	1.1275	1.1308	1.1207	1.1203
22	1.0204	1.0208	1.0288	1.0440	1.0560	1.0667	1.0640	1.0637	1.0733	1.0837	1.0835	1.0683	1.0711	1.0785	1.0927	1.1043	1.1229	1.1151	1.1190
23	1.0580	1.0579	1.0581	1.0626	1.0409	1.0555	1.0519	1.0409	1.0588	1.0658	1.0567	1.0648	1.0540	1.0681	1.0803	1.0824	1.0985	1.0979	1.1164
24	1.0706	1.0705	1.0679	1.0675	1.0443	1.0339	1.0321	1.0347	1.0474	1.0500	1.0524	1.0694	1.0668	1.0721	1.0688	1.0720	1.0802	1.0892	1.0954
25	1.0829	1.0829	1.0661	1.0615	1.0333	1.0263	1.0196	1.0284	1.0203	1.0316	1.0388	1.0480	1.0540	1.0637	1.0666	1.0633	1.0706	1.0837	1.0898
26	1.0852	1.0847	1.0831	1.0826	1.0872	1.0889	1.0779	1.0768	1.0577	1.0589	1.0683	1.0656	1.0590	1.0666	1.0671	1.0755	1.0757	1.0719	1.0792
27	1.1111	1.1106	1.0817	1.0818	1.0997	1.1030	1.1036	1.0977	1.0773	1.0686	1.0750	1.0802	1.0823	1.0752	1.0687	1.0716	1.0702	1.0591	1.0719
28	1.1128	1.1122	1.1228	1.1140	1.1220	1.1019	1.1076	1.1098	1.1127	1.0963	1.0953	1.0904	1.0966	1.0882	1.0753	1.0651	1.0659	1.0732	1.0702
29	1.0891	1.0887	1.1001	1.1015	1.1020	1.1119	1.1120	1.1093	1.1052	1.0812	1.0718	1.0697	1.0739	1.0650	1.0747	1.0783	1.0686	1.0709	1.0489
30	1.0763	1.0763	1.0857	1.0989	1.0943	1.0997	1.0945	1.0864	1.0948	1.0731	1.0712	1.0831	1.0768	1.0645	1.0707	1.0801	1.0808	1.0894	1.0822
31	0.8787	0.8791	0.9105	0.9514	0.9762	0.9967	1.0200	1.0420	1.0567	1.0728	1.0795	1.0773	1.0710	1.0756	1.0756	1.0782	1.0588	1.0401	1.0502
32	0.9754	0.9758	0.9673	0.9647	0.9713	0.9728	0.9929	0.9969	1.0137	1.0233	1.0381	1.0507	1.0563	1.0713	1.0746	1.0636	1.0740	1.0824	1.0661
33	1.0241	1.0243	0.9997	0.9849	0.9706	0.9713	0.9747	0.9800	0.9875	1.0063	1.0154	1.0248	1.0394	1.0332	1.0130	1.0328	1.0542	1.0665	1.0718
34	1.0738	1.0737	1.0582	1.0266	1.0196	1.0092	1.0024	0.9986	0.9999	1.0093	1.0166	1.0245	1.0277	1.0453	1.0548	1.0559	1.0476	1.0511	1.0666
35	1.0852	1.0853	1.0822	1.0667	1.0658	1.0568	1.0437	1.0278	1.0220	1.0282	1.0292	1.0364	1.0352	1.0451	1.0558	1.0261	1.0123	1.0257	1.0415
36	1.0714	1.0718	1.0837	1.0824	1.0864	1.0838	1.0777	1.0695	1.0556	1.0433	1.0359	1.0350	1.0372	1.0406	1.0409	1.0591	1.0540	1.0453	1.0448
37	1.0209	1.0214	1.0337	1.0482	1.0568	1.0516	1.0653	1.0593	1.0405	1.0338	1.0163	1.0245	1.0322	1.0273	1.0302	1.0324	1.0261	1.0286	1.0391
38	1.0187	1.0194	1.0245	1.0319	1.0385	1.0455	1.0278	1.0385	1.0276	1.0119	1.0281	1.0287	1.0213	1.0265	1.0231	1.0254	1.0194	1.0168	1.0300
39	0.9698	0.9701	0.9751	0.9879	0.9994	1.0148	1.0145	1.0213	1.0234	1.0236	1.0251	1.0303	1.0171	1.0223	1.0266	1.0291	1.0456	1.0310	1.0213
40	0.8841	0.8844	0.8945	0.9110	0.9404	0.9556	0.9652	0.9825	0.9963	1.0095	1.0108	1.0220	1.0161	1.0207	1.0250	1.0192	1.0179	1.0331	1.0264
41	0.9679	0.9683	0.9660	0.9547	0.9547	0.9598	0.9673	0.9757	0.9588	0.9666	0.9780	0.9889	0.9936	1.0000	1.0059	1.0168	1.0144	1.0232	1.0153
42	1.0392	1.0396	1.0255	1.0219	0.9927	0.9816	0.9822	0.9750	0.9869	0.9937	0.9739	0.9728	0.9850	0.9937	0.9995	1.0027	0.9998	1.0062	0.9992
43	1.0623	1.0627	1.0415	1.0412	1.0252	1.0137	1.0118	0.9947	0.9899	0.9943	0.9764	0.9534	0.9453	0.9368	0.9504	0.9638	0.9777	0.9847	0.9987
44	1.0616	1.0619	1.0591	1.0529	1.0440	1.0220	1.0195	1.0265	1.0093	1.0020	0.9855	0.9963	0.9837	0.9814	0.9794	0.9784	0.9876	0.9859	0.9812
45	1.0507	1.0505	1.0357	1.0405	1.0397	1.0477	1.0362	1.0404	1.0445	1.0440	1.0440	1.0334	1.0282	1.0176	0.9924	0.9870	0.9817	0.9677	0.9722
46	1.0646	1.0642	1.0610	1.0456	1.0562	1.0480	1.0409	1.0440	1.0448	1.0494	1.0457	1.0478	1.0505	1.0415	1.0286	1.0040	0.9909	0.9559	0.9648
47	1.1034	1.1027	1.1042	1.0885	1.0907	1.0959	1.0879	1.0683	1.0689	1.0580	1.0624	1.0575	1.0539	1.0446	1.0389	1.0319	1.0355	0.9917	0.9899
48	1.1051	1.1045	1.0992	1.1016	1.1044	1.1061	1.1034	1.1003	1.0992	1.0909	1.0776	1.0724	1.0652	1.0565	1.0542	1.0531	1.0399	1.0178	1.0140
49	1.0810	1.0808	1.0917	1.0989	1.1052	1.1057	1.1098	1.1099	1.1075	1.1025	1.1001	1.0838	1.0812	1.0770	1.0767	1.0764	1.0661	1.0862	1.0695
50	1.0804	1.0804	1.0924	1.0950	1.0982	1.0894	1.0894	1.0937	1.1041	1.1003	1.0999	1.0886	1.0970	1.0973	1.0903	1.0921	1.0765	1.0961	1.0794
51	1.0680	1.0681	1.0455	1.0348	1.0395	1.0458	1.0596	1.0598	1.0682	1.0816	1.0979	1.0964	1.0980	1.0927	1.0934	1.0861	1.0813	1.0932	1.0953
52	0.9961	0.9960	1.0151	1.0222	1.0029	1.0176	1.0285	1.0376	1.0298	1.0472	1.0555	1.0705	1.0636	1.0687	1.0740	1.0778	1.0883	1.0914	1.1002
53	0.9615	0.9610	0.9822	0.9916	0.9882	1.0014	1.0043	1.0127	1.0201	1.0264	1.0266	1.0486	1.0426	1.0413	1.0513	1.0666	1.0624	1.0793	1.0876
54	0.9864	0.9861	0.9889	0.9888	0.9929	0.9923	1.0042	1.0122	1.0152	1.0042	1.0065	1.0062	1.0021	0.9998	1.0098	1.0105	1.0303	1.0479	1.0575
55	0.9996	0.9993	1.0000	0.9723	0.9864	0.9990	0.9976	0.9997	1.0108	1.0025	0.9937	0.9898	1.0046	1.0158	1.0092	1.0119	1.0142	1.0093	1.0091
56	1.0238	1.0237	1.0237	1.0127	1.0096	1.0025	0.9966	0.9952	0.9879	0.9871	0.9919	0.9886	0.9814	0.9928	0.9928	1.0017	1.0022	0.9978	0.9903
57	1.0608	1.0607	1.0552	1.0623	1.0452	1.0371	1.0198	1.0168	1.0195	1.0124	1.0163	1.0051	1.0119	1.0171	1.0143	1.0220	1.0302	1.0224	1.0193
58	1.0993	1.0989	1.0832	1.0774	1.0674	1.0594	1.0371	1.0346	1.0369	1.0356	1.0322	1.0145	1.0072	0.9928	0.9945	0.9929	1.0020	1.0149	1.0202
59	1.1022	1.1014	1.0973	1.0973	1.0942	1.0578	1.0439	1.0439	1.0478	1.0450	1.0466	1.0365	1.0448	1.0078	1.0004	0.9986	0.9757	0.9952	1.0083
60	1.1164	1.1157	1.1188	1.1168	1.1114	1.1071	1.0907	1.0754	1.0624	1.0555	1.0539	1.0509	1.0587	1.0714	1.0463	1.0308	1.0303	1.0310	1.0309
61	1.1182	1.1175	1.1184	1.1177	1.1048	1.0905	1.0760	1.0707	1.0692	1.0651	1.0574	1.0555	1.0575	1.0435	1.0423	1.0298	1.0140	1.0057	1.0015
62	1.0343	1.0438	1.0543	1.0494	1.0561	1.0636	1.0608	1.0645	1.0627	1.0625	1.0384	1.0341	1.0369	1.0359	1.0470	1.0460	1.0598	1.0541	1.0444
63	0.9622	0.9801	0.9876	1.0089	0.9991	1.0083	1.0368	1.0407	1.0319	1.0383	1.0474	1.0385	1.0203	1.0002	1.0140	1.0273	1.0411	1.0429	1.0440
64	0.8786	0.9073	0.9180	0.9457	0.9600	0.9583	0.9786	0.9872	0.9898	1.0079	1.0201	1.0199	1.0271	1.0381	1.0378	1.0402	1.0379	1.0240	1.0284
65	1.0475	1.0259	0.9842	0.9640	0.9681	0.9602	0.9790	0.9846	0.9887	0.9934	0.9986	1.0152	1.0138	1.0235	1.0288	1.0293	1.0207	1.0338	1.0382
66	1.0113	0.9806	1.0060	1.0149	1.0182	1.0265	1.0337	1.0240	1.0198	1.0100	1.0158	1.0257	1.0281	1.0395	1.0179	0.9829	0.9686	0.9817	0.9701
67	1.0479	1.0272	1.0296	1.0349	1.0161	1.0420	1.0368	1.0246	1.0281	1.0119	1.0012	1.0033	1.0161	1.0229	1.0207	1.0004	1.0119	1.0077	1.0150
68	1.0729	1.0620	1.0504	1.0480	0.9911	1.0010	1.0030	0.9753	0.9792	0.9833	0.9901	0.9836	0.9901	1.0030	1.0183	1.0023	1.0063	1.0035	1.0118
69	1.0748	1.0810	1.0749	1.0564	1.0311	1.0239	1.0124	1.0052	1.0130	1.0126	0.9959	0.9946	0.9996	1.0000	1.0088	0.9866	1.0076	1.0063	1.0065
70	1.0752	1.0827	1.0774	1.0696	1.0855	1.0680	1.0511	1.0569	1.0113	1.0166	1.0171	1.0171	1.0166	1.0207	1.0244	1.0244	1.0149	1.0200	1.0021
71	1.1004	1.0870	1.0562	1.0198	1.0493	1.0502	1.0574	1.0611	1.0733	1.0646	1.0544	1.0488	1.0368	1.0263	1.0242	1.0465	1.0248	1.0223	1.0148
72	1.0163	1.0181	1.0181	1.0381	1.0508	1.0584	1.0691	1.0595	1.0645	1.0577	1.0605	1.0571	1.0480	1.0306	1.0335	1.0479	1.0417	1.0347	1.0343
73	1.0411	1.0420	1.0452	1.0472	1.0554	1.0523	1.0576	1.0592	1.0612	1.0593	1.0574	1.0567	1.0427	1.0226	1.0300	1.0412	1.0376	1.0358	1.0453
74	1.0656	1.0715	1.0746	1.0691	1.0730	1.0600	1.0482	1.0536	1.0474	1.0548	1.0496	1.0434	1.0511	1.0585	1.0365	1.0420	1.0350	1.0373	1.0454
75	0.9452	0.9730	1.0052	1.0149	1.0298	1.0339	0.9943	1.0148	1.0152	1.0295	0.9979	1.0038	1.0124	1.0205	1.0126	1.0245	1.0255	1.0210	1.0129
76	0.9576	0.9655	0.9743	0.9867	0.9956	1.0091	1.0073	1.0072	1.0090	1.0157	1.0347	1.0185	1.0236	1.0297	1.0369	1.0436	1.0453	1.0261	1.0003
77	1.0415	1.0287	1.0185	1.0187	0.9996	0.9804	1.0168	1.0148	0.9967	1.0088	1.0226	1.0215	0.9993	0.9988	1.0079	1.0021	1.0211	1.0240	1.0198
78	0.9931	0.9983	1.0032	0.9864	0.9723	0.9707	0.9792	0.9837	0.9890	0.9882	1.0001	0.9883	0.9894	0.9990	0.9978	0.9595	0.9728	0.9954	1.0148
79	0.9938	1.0003	0.9994	0.9936	0.9808	0.9889	0.9835	0.9843	0.9995	0.9921	1.0008	1.0016	0.9673	0.9516	0.9738	0.9829	0.9757	0.9589	0.9734
80	0.9458	0.9536	0.9510	0.9613	0.9632	0.9699	0.9595	0.9648	0.9720	0.9796	0.9855	0.9917	1.0161	1.0188	1.0084	0.9991	0.9822	0.9715	0.9720
81	0.9586	0.9657	0.9570	0.9521	0.9825	0.9835	0.9739	0.9804	0.9772	0.9837	0.9697	0.9829	0.9913	0.9811	0.9828	0.9846	0.9864	0.9967	0.9890
82	0.9595	0.9707	0.9713	0.9349	0.9563	0.9670	0.9815	0.9896	0.9724	0.9759	0.9556	0.9610	0.9637	0.9838	0.9927	1.0044	0.9955	1.0025	0.9957
83	0.9475	0.9411	0.9475	0.9681	0.9441	0.9553	0.9584	0.9543	0.9606	0.9644	0.9645	0.9570	0.9483	0.9540	0.9613	0.9816	0.9981	0.9651	0.9798
84	0.9850	0.9227	0.9425	0.9327	0.9204	0.9317	0.9206	0.9094	0.9330	0.9312	0.9554	0.9739	0.9901	0.9962	0.9832	0.9712	0.9843	0.9984	0.9844
85	1.0219	1.0164	0.9936	0.9871	0.9918	0.9831	0.9805	0.9788	0.9832	0.9740	0.9840	0.9777	0.9781	0.9804	0.9765	0.9809	0.9689	0.9774	0.9634
86	1.0224	1.0255	1.0065	1.0095	1.0144	0.9523	0.9341	0.9406	0.9454	0.9520	0.9556	0.9731	0.9639	0.9726	0.9658	0.9672	0.9578	0.9338	0.9401
87	1.0584	1.0020	1.0022	1.0122	1.0084	0.9744	0.9505	0.9545	0.9509	0.9453	0.9252	0.9335	0.9480	0.9592	0.9706	0.9845	0.9875	0.9589	0.9534
88	1.0214	1.0198	1.0213	1.0238	0.9783	0.9895	0.9942	0.9863	0.9727	0.9451	0.9451	0.9371	0.9370	0.9389	0.9068	0.9202	0.9434	0.9715	0.9584
89	0.9598	0.9536	0.9745	0.9748	0.9764	0.9894	0.9912	0.9888	0.9633	0.9769	0.9727	0.9660	0.9518	0.9371	0.9374	0.9314	0.9423	0.9555	0.9645
90	0.9101	0.9295	0.9257	0.9407	0.9591	0.9754	0.9897	0.9644	0.9759	0.9761	0.9798	0.9757	0.9907	0.9840	0.9604	0.9405	0.9382	0.9390	0.9353
91	0.8738	0.9103	0.9005	0.9040	0.9131	0.9103	0.9135	0.9002	0.9172	0.9292	0.9114	0.9146	0.9323	0.9136	0.9267	0.9381	0.9331	0.9319	0.9238
92	0.7769	0.8245	0.8381	0.8377	0.8589	0.8690	0.8928	0.9058	0.9159	0.9138	0.9234	0.9172	0.9233	0.9233	0.9187	0.9103	0.8841	0.8961	0.9315
93	0.8024	0.7915	0.8253	0.8327	0.8462	0.8623	0.8792	0.8894	0.9012	0.9058	0.9302	0.9288	0.8987	0.9089	0.9176	0.8784	0.8826	0.8991	0.9160
94	0.8758	0.8496	0.8326	0.8387	0.8471	0.8643	0.8747	0.8689	0.8775	0.8885	0.8965	0.9076	0.9004	0.9014	0.9038	0.8999	0.8891	0.8940	0.8992

95	1.0224	0.9819	0.9407	0.9416	0.9295	0.9208	0.9126	0.8962	0.8890	0.8863	0.8921	0.8854	0.9017	0.9044	0.8781	0.8998	0.9009	0.9003	0.8845
96	0.8517	0.8777	0.8953	0.8729	0.8802	0.8848	0.8847	0.9126	0.9087	0.9080	0.9103	0.9111	0.9190	0.9111	0.8962	0.9128	0.9301	0.9307	0.8807
97	0.8032	0.8283	0.8445	0.8515	0.8609	0.8644	0.8793	0.8931	0.8934	0.8872	0.8968	0.9023	0.9071	0.8720	0.8826	0.8983	0.9078	0.9104	0.9118
98	0.7792	0.7952	0.7883	0.7976	0.7924	0.7872	0.8058	0.8187	0.8227	0.8143	0.8111	0.8320	0.8457	0.8691	0.8813	0.8830	0.8920	0.8963	0.9173
99	0.7308	0.7462	0.7716	0.7830	0.7672	0.7658	0.7654	0.7739	0.7845	0.7893	0.7843	0.7993	0.8110	0.8441	0.8718	0.8699	0.8388	0.8287	0.8579
100	0.9507	0.9043	0.8444	0.8330	0.7820	0.8028	0.7739	0.7754	0.7747	0.7808	0.7806	0.7871	0.7722	0.7936	0.8154	0.8240	0.8083	0.8062	0.8292
101	0.8535	0.8518	0.8471	0.8395	0.8169	0.7859	0.7658	0.7680	0.7648	0.7624	0.7809	0.7793	0.7819	0.7978	0.8111	0.8015	0.8179	0.8243	0.8293
102	0.8794	0.8777	0.8728	0.8642	0.8774	0.8695	0.8528	0.8342	0.8181	0.8219	0.8185	0.8097	0.8087	0.7851	0.7946	0.7847	0.8011	0.8156	0.8193
103	0.8435	0.8503	0.8659	0.8666	0.8898	0.8891	0.8985	0.8610	0.8635	0.8667	0.8555	0.8085	0.8106	0.8076	0.8065	0.7923	0.8014	0.8047	0.8084
104	0.8437	0.8396	0.8476	0.8442	0.8621	0.8747	0.8947	0.9120	0.8914	0.8670	0.8403	0.8375	0.8298	0.7969	0.7865	0.8043	0.8140	0.8169	0.8137
105	0.8924	0.8824	0.8764	0.8519	0.8619	0.8667	0.8646	0.8771	0.8518	0.8579	0.8242	0.8312	0.8296	0.8342	0.8218	0.8235	0.8141	0.8173	0.8160
106	0.9653	0.9106	0.8944	0.9105	0.9017	0.8875	0.8534	0.8605	0.8512	0.8560	0.8623	0.8665	0.8652	0.8406	0.8177	0.8261	0.8265	0.8140	0.7924
107	0.9897	0.9448	0.9412	0.9321	0.9292	0.9175	0.8613	0.8397	0.8377	0.8446	0.8519	0.8557	0.8597	0.8366	0.8402	0.8454	0.8414	0.8106	0.7883
108	0.9776	0.9410	0.9379	0.9456	0.9492	0.9010	0.9098	0.8984	0.9104	0.8896	0.8498	0.8470	0.8547	0.8466	0.8412	0.8516	0.8416	0.8443	0.8303
109	1.0019	1.0128	0.9928	0.9920	0.9834	0.9484	0.9354	0.9138	0.9257	0.9114	0.9259	0.9228	0.8720	0.8313	0.8414	0.8474	0.8468	0.8418	0.8605
110	0.8064	0.8406	0.8689	0.8712	0.8919	0.9097	0.9244	0.9240	0.9211	0.9315	0.9486	0.9298	0.9013	0.8924	0.8783	0.8613	0.8556	0.8543	0.8495
111	0.7699	0.8010	0.8272	0.8239	0.8251	0.8620	0.8846	0.9053	0.8945	0.9084	0.9170	0.9139	0.9002	0.9165	0.9154	0.8649	0.8594	0.8309	0.8269
112	0.8311	0.8327	0.8202	0.8037	0.8084	0.8341	0.8528	0.8737	0.8842	0.8915	0.8800	0.8840	0.8879	0.9165	0.9038	0.8608	0.8545	0.8633	0.8546
113	0.9656	0.9352	0.9009	0.8881	0.8763	0.8814	0.8734	0.8906	0.8962	0.8895	0.8855	0.8932	0.9090	0.9236	0.9246	0.9152	0.9139	0.9100	0.9066
114	0.9909	0.9468	0.8999	0.9200	0.9185	0.9138	0.8831	0.8737	0.8793	0.8468	0.8566	0.8510	0.8551	0.8785	0.8835	0.9025	0.9120	0.9027	0.8845
115	0.8451	0.8611	0.8612	0.8309	0.8391	0.8555	0.8412	0.8463	0.8466	0.8488	0.8010	0.8297	0.8194	0.8304	0.8375	0.8586	0.8687	0.8574	0.8651
116	0.8949	0.8875	0.9067	0.8906	0.8808	0.8736	0.8901	0.8688	0.8718	0.8250	0.8207	0.7712	0.7567	0.7664	0.7918	0.8153	0.8299	0.8418	0.8528
117	0.8957	0.9280	0.9348	0.9384	0.9318	0.9017	0.8779	0.8722	0.8848	0.8555	0.8439	0.8239	0.8436	0.8173	0.8052	0.8181	0.8252	0.8017	0.8204
118	0.9203	0.9167	0.9316	0.9483	0.9357	0.8961	0.9115	0.8880	0.8919	0.9016	0.9097	0.9077	0.9057	0.8806	0.8824	0.8731	0.8639	0.8614	0.8717
119	0.9204	0.8752	0.8891	0.9190	0.9279	0.8676	0.8818	0.8676	0.8778	0.8903	0.9074	0.9042	0.8709	0.8537	0.8397	0.8495	0.8267	0.8545	0.8598
120	0.9934	0.9493	0.9387	0.9343	0.9395	0.9270	0.9360	0.8946	0.8856	0.8858	0.8844	0.8962	0.8524	0.8795	0.8632	0.8500	0.7995	0.8024	0.8003
121	1.0053	1.0170	0.9948	0.9777	0.9569	0.9755	0.9686	0.9863	0.9534	0.9450	0.9581	0.9215	0.9301	0.9348	0.8986	0.8410	0.8223	0.8458	0.8257
122	1.0050	1.0160	1.0183	1.0152	1.0042	1.0294	0.9929	1.0143	1.0032	1.0050	1.0001	0.9922	1.0005	0.9592	0.9478	0.9185	0.8930	0.8944	0.8615
123	0.9920	0.9788	0.9797	0.9865	0.9911	1.0002	0.9651	0.9734	0.9879	0.9948	0.9800	0.9931	1.0025	0.9949	0.9941	0.9853	0.9909	0.9582	0.9138

	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
01-Jul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
03-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
04-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
05-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
06-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
07-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
08-Jul	0.0834	0.0818	0.0511	0.0499	0.0267	0.0044	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
09-Jul	0.0199	0.0166	0.0012	0.0113	0.0098	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
18-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
19-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
26-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
27-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
28-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
29-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
30-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
31-Jul	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
01-Aug	0.1213	0.1191	0.0875	0.0461	0.0210	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
02-Aug	0.1459	0.1415	0.1183	0.0790	0.0468	0.0234	0.0026	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
03-Aug	0.1218	0.1154	0.1167	0.0916	0.0734	0.0483	0.0234	0.0157	0.0081	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
04-Aug	0.0479	0.0398	0.0565	0.0626	0.0509	0.0352	0.0165	0.0129	0.0039	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
05-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
06-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
07-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
08-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
09-Aug	0.0302	0.0280	0.0229	0.0096	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10-Aug	0.1461	0.1418	0.1265	0.0961	0.0567	0.0406	0.0303	0.0132	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11-Aug	0.1782	0.1718	0.1585	0.1389	0.0992	0.0770	0.0585	0.0332	0.0368	0.0284	0.0168	0.0057	0.0007	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12-Aug	0.1390	0.1303	0.1311	0.1145	0.1037	0.0916	0.0718	0.0540	0.0456	0.0297	0.0377	0.0276	0.0101	0.0005	0.0000	0.0000	0.0000	0.0000	0.0000
13-Aug	0.0767	0.0658	0.0877	0.0708	0.0756	0.0741	0.0555	0.0550	0.0513	0.0304	0.0561	0.0688	0.0591	0.0578	0.0436	0.0296	0.0155	0.0088	0.0000
14-Aug	0.0151	0.0021	0.0267	0.0155	0.0287	0.0482	0.0315	0.0243	0.0376	0.0234	0.0655	0.0671	0.0697	0.0705	0.0584	0.0446	0.0212	0.0164	0.0117
15-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0163	0.0284	0.0358	0.0470	0.0600	0.0510	0.0327	0.0423	0.0324
16-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0255	0.0403	0.0351	0.0799	0.0605
17-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0018	0.0000	0.0817	0.0635
18-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0573	0.0424
19-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
20-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
21-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
22-Aug	0.0039	0.0022	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
23-Aug	0.0424	0.0394	0.0159	0.0059	0.0089	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
24-Aug	0.0561	0.0515	0.0250	0.0146	0.0131	0.0039	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
25-Aug	0.0564	0.0504	0.0231	0.0398	0.0239	0.0012	0.0000	0.0000	0.0000	0.0000	0.0011	0.0049	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
26-Aug	0.0326	0.0248	0.0000	0.0246	0.0114	0.0000	0.0000	0.0005	0.0077	0.0079	0.0039	0.0109	0.0129	0.0013	0.0013	0.0000	0.0000	0.0000	0.0026
27-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
28-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0013	0.0000	0.0005	0.0000	0.0000	0.0000
29-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0175	0.0000	0.0000
30-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
31-Aug	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
01-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
02-Sep	0.0378	0.0181	0.0105	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
03-Sep	0.1593	0.1090	0.0906	0.0518	0.0372	0.0379	0.0169	0.0086	0.0058	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

04-Sep	0.1118	0.0813	0.1044	0.0854	0.0662	0.0739	0.0335	0.0198	0.0127	0.0015	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
05-Sep	0.1005	0.0988	0.0965	0.0680	0.0451	0.0436	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0105	0.0246	0.0118	0.0228
06-Sep	0.0526	0.0698	0.0650	0.0307	0.0262	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0035	0.0060	0.0000	0.0007
07-Sep	0.0000	0.0059	0.0126	0.0000	0.0322	0.0000	0.0000	0.0204	0.0164	0.0117	0.0047	0.0110	0.0042	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
08-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0110	0.0000	0.0000	0.0036	0.0111	0.0000	0.0000	0.0000	0.0068	0.0000	0.0000	0.0000
09-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
11-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
12-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
13-Sep	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
14-Sep	0.0548	0.0252	0.0000	0.0000	0.0000	0.0000	0.0012	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
15-Sep	0.0972	0.0579	0.0237	0.0109	0.0015	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
16-Sep	0.0557	0.0274	0.0033	0.0000	0.0000	0.0158	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
17-Sep	0.0626	0.0272	0.0000	0.0111	0.0248	0.0413	0.0163	0.0120	0.0066	0.0068	0.0000	0.0064	0.0049	0.0000	0.0000	0.0339	0.0205	0.0000	0.0000
18-Sep	0.0689	0.0251	0.0000	0.0150	0.0411	0.0487	0.0283	0.0234	0.0027	0.0096	0.0000	0.0000	0.0320	0.0425	0.0203	0.0443	0.0381	0.0346	0.0195
19-Sep	0.1230	0.0697	0.0471	0.0512	0.0751	0.0750	0.0643	0.0543	0.0264	0.0250	0.0093	0.0030	0.0102	0.0179	0.0060	0.0387	0.0491	0.0567	0.0403
20-Sep	0.1644	0.1022	0.0881	0.0966	0.0897	0.0877	0.0859	0.0697	0.0448	0.0363	0.0344	0.0147	0.0132	0.0309	0.0173	0.0474	0.0560	0.0534	0.0442
21-Sep	0.2049	0.1297	0.1149	0.1592	0.1305	0.1169	0.1000	0.0758	0.0681	0.0554	0.0736	0.0483	0.0439	0.0412	0.0187	0.0364	0.0537	0.0444	0.0414
22-Sep	0.2574	0.1867	0.1654	0.1886	0.1836	0.1578	0.1371	0.1173	0.1032	0.0860	0.1039	0.0860	0.0899	0.0814	0.0515	0.0481	0.0488	0.0729	0.0544
23-Sep	0.2724	0.2622	0.2210	0.2534	0.2604	0.2223	0.2120	0.2036	0.1657	0.1497	0.1434	0.1067	0.0942	0.0793	0.0624	0.0703	0.0578	0.0680	0.0629
24-Sep	0.2505	0.2440	0.2255	0.2638	0.2657	0.2354	0.2270	0.2206	0.1782	0.1707	0.1542	0.1237	0.1104	0.0930	0.0801	0.0828	0.0821	0.0841	0.0924
25-Sep	0.2281	0.2167	0.2171	0.2518	0.2485	0.2793	0.2884	0.2757	0.2284	0.2137	0.1934	0.1453	0.1409	0.1146	0.1084	0.1090	0.1176	0.1438	0.1451
26-Sep	0.1697	0.2129	0.2129	0.2371	0.2372	0.3010	0.3334	0.3169	0.2731	0.2633	0.2630	0.2064	0.1872	0.1495	0.1319	0.1179	0.1234	0.1784	0.1846
27-Sep	0.1483	0.1914	0.1897	0.2108	0.2560	0.3077	0.3348	0.3264	0.2961	0.3131	0.3127	0.2640	0.2446	0.2047	0.2192	0.1910	0.1732	0.2004	0.2191
28-Sep	0.1885	0.2359	0.2132	0.2336	0.2767	0.3145	0.3390	0.3333	0.3284	0.3312	0.3348	0.2927	0.2871	0.2618	0.2760	0.2530	0.2241	0.2384	0.2475
29-Sep	0.2784	0.3046	0.2856	0.2904	0.3147	0.3353	0.3449	0.3646	0.3481	0.3501	0.3498	0.3117	0.2907	0.2719	0.3096	0.3059	0.2792	0.2930	0.3050
30-Sep	0.4047	0.3925	0.3832	0.3839	0.3987	0.4212	0.4269	0.4602	0.4265	0.4158	0.4332	0.3917	0.3528	0.3524	0.3770	0.3612	0.3394	0.3546	0.3741
01-Oct	0.6278	0.5662	0.5431	0.5437	0.5369	0.5484	0.5296	0.5501	0.5063	0.4970	0.5046	0.4692	0.4238	0.4232	0.4523	0.4443	0.4485	0.4520	0.4355
02-Oct	0.8254	0.7729	0.7158	0.7085	0.6879	0.6823	0.6460	0.6564	0.6007	0.5862	0.5692	0.5350	0.5194	0.5084	0.5289	0.5593	0.5591	0.5464	0.5124
03-Oct	0.9496	0.9215	0.8812	0.8673	0.8379	0.8142	0.7668	0.7833	0.7188	0.6926	0.6675	0.6221	0.6133	0.6012	0.6191	0.6528	0.6633	0.6459	0.6061
04-Oct	0.9271	0.9377	0.9386	0.9233	0.9056	0.8896	0.8497	0.8828	0.8254	0.8012	0.7702	0.7313	0.7059	0.6909	0.7352	0.7464	0.7556	0.7392	0.7145
05-Oct	1.0755	1.0582	1.0414	1.0479	1.0225	1.0010	0.9605	0.9659	0.9124	0.8882	0.8547	0.8149	0.7813	0.7739	0.8330	0.8269	0.8188	0.8021	0.8267
06-Oct	1.2723	1.2281	1.1949	1.1939	1.1588	1.1328	1.0767	1.0685	1.0146	0.9960	0.9528	0.9073	0.8685	0.8960	0.9445	0.9220	0.9043	0.8852	0.9078
07-Oct	1.4931	1.4312	1.4046	1.3938	1.3635	1.3418	1.2664	1.2456	1.1876	1.1767	1.1365	1.0699	1.0171	1.0210	1.0573	1.0324	1.0055	0.9824	0.9833
08-Oct	1.7623	1.6832	1.6311	1.6083	1.5934	1.5721	1.4966	1.4674	1.3988	1.3824	1.3471	1.2653	1.2004	1.1711	1.1796	1.1559	1.1600	1.1473	1.1183
09-Oct	1.8116	1.7770	1.7848	1.7728	1.8085	1.7655	1.7182	1.6878	1.6197	1.5965	1.5612	1.4729	1.4225	1.3716	1.3582	1.3253	1.3449	1.3346	1.2820
10-Oct	1.9581	1.9234	1.9358	1.9308	1.9888	1.9758	1.9479	1.9155	1.8506	1.8291	1.7751	1.6882	1.6350	1.5679	1.5412	1.5172	1.5203	1.5038	1.4456
11-Oct	2.0787	2.0439	2.0610	2.0641	2.1085	2.1025	2.0907	2.0771	2.0281	2.0022	1.9514	1.8732	1.8206	1.7769	1.7407	1.7258	1.7125	1.6818	1.6192
12-Oct	2.2353	2.1918	2.1932	2.1950	2.2159	2.2096	2.1876	2.2118	2.1602	2.1304	2.0907	2.0594	2.0044	1.9634	1.9283	1.9270	1.9043	1.8706	1.8037
13-Oct	2.3916	2.3504	2.3436	2.3483	2.3509	2.3311	2.2884	2.2956	2.2645	2.2584	2.2452	2.2165	2.1689	2.1607	2.1359	2.1161	2.0836	2.0472	1.9828
14-Oct	2.4992	2.4662	2.4653	2.4939	2.4862	2.4606	2.4193	2.4142	2.4083	2.3955	2.4159	2.3799	2.3337	2.3206	2.3082	2.2859	2.2627	2.2234	2.1597
15-Oct	2.5339	2.5538	2.5689	2.5809	2.5816	2.5693	2.5614	2.5495	2.5527	2.5344	2.5484	2.5081	2.4628	2.4742	2.4846	2.4532	2.4295	2.4029	2.3602
16-Oct	2.5442	2.6072	2.6257	2.6464	2.6495	2.6481	2.6956	2.7055	2.7107	2.6848	2.6913	2.6470	2.5974	2.6317	2.6385	2.6012	2.5813	2.5858	2.5649
17-Oct	2.5666	2.6644	2.6859	2.6983	2.6974	2.7433	2.7813	2.8028	2.7959	2.7902	2.8363	2.7947	2.7370	2.7792	2.7914	2.7429	2.7330	2.7350	2.7274
18-Oct	2.5647	2.6499	2.6912	2.7038	2.7112	2.7910	2.8414	2.8847	2.8658	2.8738	2.9052	2.8665	2.8593	2.9421	2.9441	2.8889	2.8794	2.8867	2.8599
19-Oct	2.7583	2.8074	2.8203	2.8301	2.8164	2.8775	2.9125	2.9564	2.9403	2.9373	2.9514	2.9314	2.9523	3.0437	3.0599	3.0210	3.0170	3.0259	3.0032
20-Oct	2.9884	3.0046	2.9911	3.0037	2.9884	3.0117	3.0234	3.0469	3.0414	3.0238	3.0292	3.0121	3.0464	3.1213	3.1386	3.1495	3.1508	3.1885	3.1692
21-Oct	3.1573	3.1701	3.1690	3.1974	3.1772	3.1738	3.1661	3.1689	3.1529	3.1273	3.1440	3.1227	3.1528	3.1990	3.2289	3.2821	3.2896	3.3188	3.3075
22-Oct	3.1916	3.2331	3.2662	3.3068	3.2980	3.2886	3.2882	3.2740	3.2523	3.2327	3.2534	3.2242	3.2381	3.2695	3.2984	3.3603	3.3689	3.4024	3.3938
23-Oct	3.2008	3.2845	3.3643	3.3844	3.3766	3.3709	3.4006	3.3961	3.3686	3.3809	3.3916	3.3679	3.3773	3.3851	3.4091	3.4512	3.4501	3.4932	3.5022
24-Oct	3.3557	3.4216	3.5011	3.5510	3.5347	3.5116	3.5550	3.5455	3.5176	3.5271	3.5854	3.5329	3.5523	3.5488	3.5657	3.5859	3.5746	3.6293	3.6300
25-Oct	3.4608	3.5323	3.5925	3.6579	3.6510	3.6342	3.6604	3.6724	3.6414	3.6970	3.7595	3.7563	3.7899	3.7765	3.7680	3.7640	3.7379	3.7810	3.7701
26-Oct	3.5650	3.6024	3.6557	3.7171	3.7163	3.7286	3.7779	3.7959	3.7522	3.8365	3.9104	3.9271	3.9406	3.9533	3.9569	3.9394	3.9060	3.9728	3.9425
27-Oct	3.6447	3.6839	3.7221	3.7662	3.7778	3.8288	3.8619	3.9036	3.8559	3.9298	3.9955	4.0141	4.0292	4.0669	4.0686	4.0597	4.0353	4.1049	4.0637
28-Oct	3.7244	3.8068	3.8311	3.8447	3.8470	3.9573	3.9757	4.0318	3.9737	4.0344	4.0830	4.1045	4.1526	4.2073	4.2230	4.2036	4.2019	4.2440	4.1968
29-Oct	3.7309	3.8557	3.8905	3.9079	3.9046	4.0266	4.0351	4.1329	4.0838	4.1436	4.1933	4.2030	4.2946	4.3219	4.3539	4.3470	4.3957	4.4351	4.3894
30-Oct	3.7256	3.8368	3.8937	3.9277	3.9448	4.0473	4.0621	4.1424	4.1260	4.1936	4.2301	4.2762	4.3588	4.3813	4.4494	4.4994	4.5666	4.5828	4.5566
31-Oct	3.7207	3.8190	3.8735	3.9100	3.9377	4.0141	4.0647	4.1238	4.1184	4.1836	4.2248	4.2787	4.3526	4.4162	4.4957	4.5743	4.6668	4.6819	4.6879

Question 8.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

I have not worked on regression analysis, although I would love to. But one application where I think I might be able to apply this in real life is, football(aka soccer). I am a big fan and a suppoter of a club called Football Club Barcelona. The club owners and the manager of the team always tries to do the best for the club like either buy players from other clubs or hire scouts to scout for potential talents in the game. FC Barcelona as a club has got some of the best players using scouts such as Lionel Messi, who is Greatest Of All Time(GOAT), and currently the new talents like Ansu Fati and Junior Firpo. Upto my understanding usually the players are bought considering who have had the best goals to matches ratio (for forwards) or tackles (for defenders) to matches ratio.

But I think that based on the stats that the scout provides such as (other than the mainstream stats such as age, matches played, goals scored etc) minutes played, number of successful passes completed, number and recurrence of injuries, nation (i can assign numerical metrics to this categorical variable), build up plays leading to goal, or the number of commanding saves when it comes to a goalkeeper, the sports analyst for the club might be able to build a mathematical regression model, and even remove insignificant predictors to build a successful model, buy the successful players.

Question 8.2

Using crime data from [http://www.statsci.org/data/general/uscrime.txt (http://www.statsci.org/data/general/uscrime.txt)](http://www.statsci.org/data/general/uscrime.txt) (file uscrime.txt, description at [http://www.statsci.org/data/general/uscrime.html (http://www.statsci.org/data/general/uscrime.html)](http://www.statsci.org/data/general/uscrime.html)), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data:

1. `M = 14.0`
2. `So = 0`
3. `Ed = 10.0`
4. `Po1 = 12.0`
5. `Po2 = 15.5`
6. `LF = 0.640`
7. `M.F = 94.0`
8. `Pop = 150`
9. `NW = 1.1`
10. `U1 = 0.120`
11. `U2 = 3.6`
12. `Wealth = 3200`
13. `Ineq = 20.1`
14. `Prob = 0.04`
15. `Time = 39.0`

Show your model (factors used and their coefficients), the software output, and the quality of fit. Note that because there are only 47 data points and 15 predictors, you'll probably notice some overfitting. We'll see ways of dealing with this sort of problem later in the course.

In [2]:
```
crime_df <- read.table("uscrime.txt",header = TRUE)
```

In [3]:
```
head(crime_df)
```

M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
15.1	1	9.1	5.8	5.6	0.510	95.0	33	30.1	0.108	4.1	3940	26.1	0.084602	26.2011	791
14.3	0	11.3	10.3	9.5	0.583	101.2	13	10.2	0.096	3.6	5570	19.4	0.029599	25.2999	1635
14.2	1	8.9	4.5	4.4	0.533	96.9	18	21.9	0.094	3.3	3180	25.0	0.083401	24.3006	578
13.6	0	12.1	14.9	14.1	0.577	99.4	157	8.0	0.102	3.9	6730	16.7	0.015801	29.9012	1969
14.1	0	12.1	10.9	10.1	0.591	98.5	18	3.0	0.091	2.0	5780	17.4	0.041399	21.2998	1234
12.1	0	11.0	11.8	11.5	0.547	96.4	25	4.4	0.084	2.9	6890	12.6	0.034201	20.9995	682

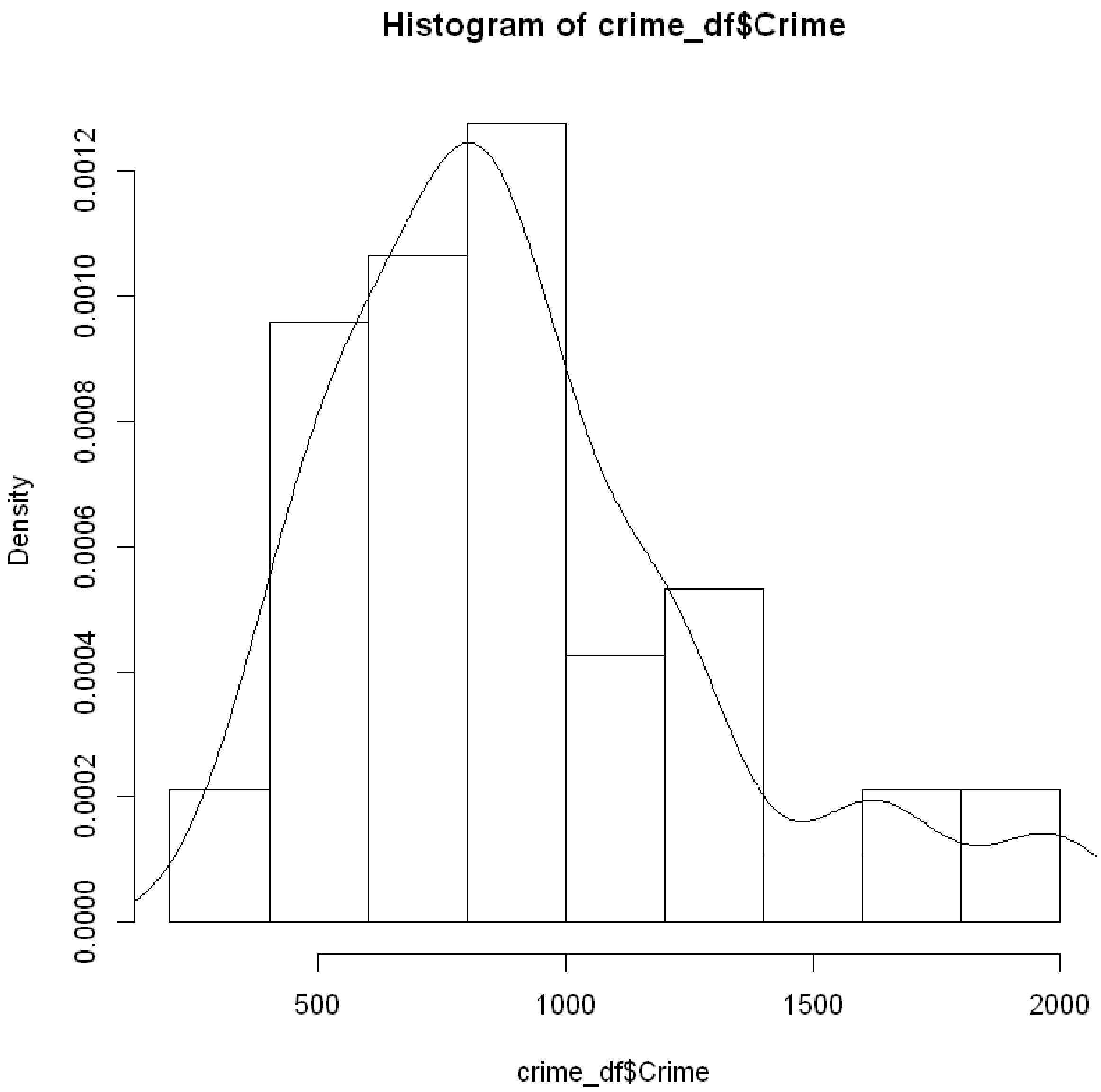
In [4]:
```
str(crime_df)
```

```
'data.frame':	47 obs. of  16 variables:
 $ M     : num  15.1 14.3 14.2 13.6 14.1 12.1 12.7 13.1 15.7 14 ...
 $ So    : int  1 0 1 0 0 0 1 1 1 0 ...
 $ Ed    : num  9.1 11.3 8.9 12.1 12.1 11 11.1 10.9 9 11.8 ...
 $ Po1   : num  5.8 10.3 4.5 14.9 10.9 11.8 8.2 11.5 6.5 7.1 ...
 $ Po2   : num  5.6 9.5 4.4 14.1 10.1 11.5 7.9 10.9 6.2 6.8 ...
 $ LF    : num  0.51 0.583 0.533 0.577 0.591 0.547 0.519 0.542 0.553 0.632 ...
 $ M.F   : num  95 101.2 96.9 99.4 98.5 ...
 $ Pop   : int  33 13 18 157 18 25 4 50 39 7 ...
 $ NW    : num  30.1 10.2 21.9 8 3 4.4 13.9 17.9 28.6 1.5 ...
 $ U1    : num  0.108 0.096 0.094 0.102 0.091 0.084 0.097 0.079 0.081 0.1 ...
 $ U2    : num  4.1 3.6 3.3 3.9 2 2.9 3.8 3.5 2.8 2.4 ...
 $ Wealth: int  3940 5570 3180 6730 5780 6890 6200 4720 4210 5260 ...
 $ Ineq  : num  26.1 19.4 25 16.7 17.4 12.6 16.8 20.6 23.9 17.4 ...
 $ Prob  : num  0.0846 0.0296 0.0834 0.0158 0.0414 ...
 $ Time  : num  26.2 25.3 24.3 29.9 21.3 ...
 $ Crime : int  791 1635 578 1969 1234 682 963 1555 856 705 ...
```

In [5]:
```
# Checking if there are any Null values in the dataframe

sapply(crime_df, function(x) sum(is.na(x)))
```

M	0
So	0
Ed	0
Po1	0
Po2	0
LF	0
M.F	0
Pop	0
NW	0
U1	0
U2	0
Wealth	0
Ineq	0
Prob	0
Time	0
Crime	0

In [6]:
```
hist(crime_df$Crime, freq = FALSE)
dens <- density(crime_df$Crime)
lines(dens)
```

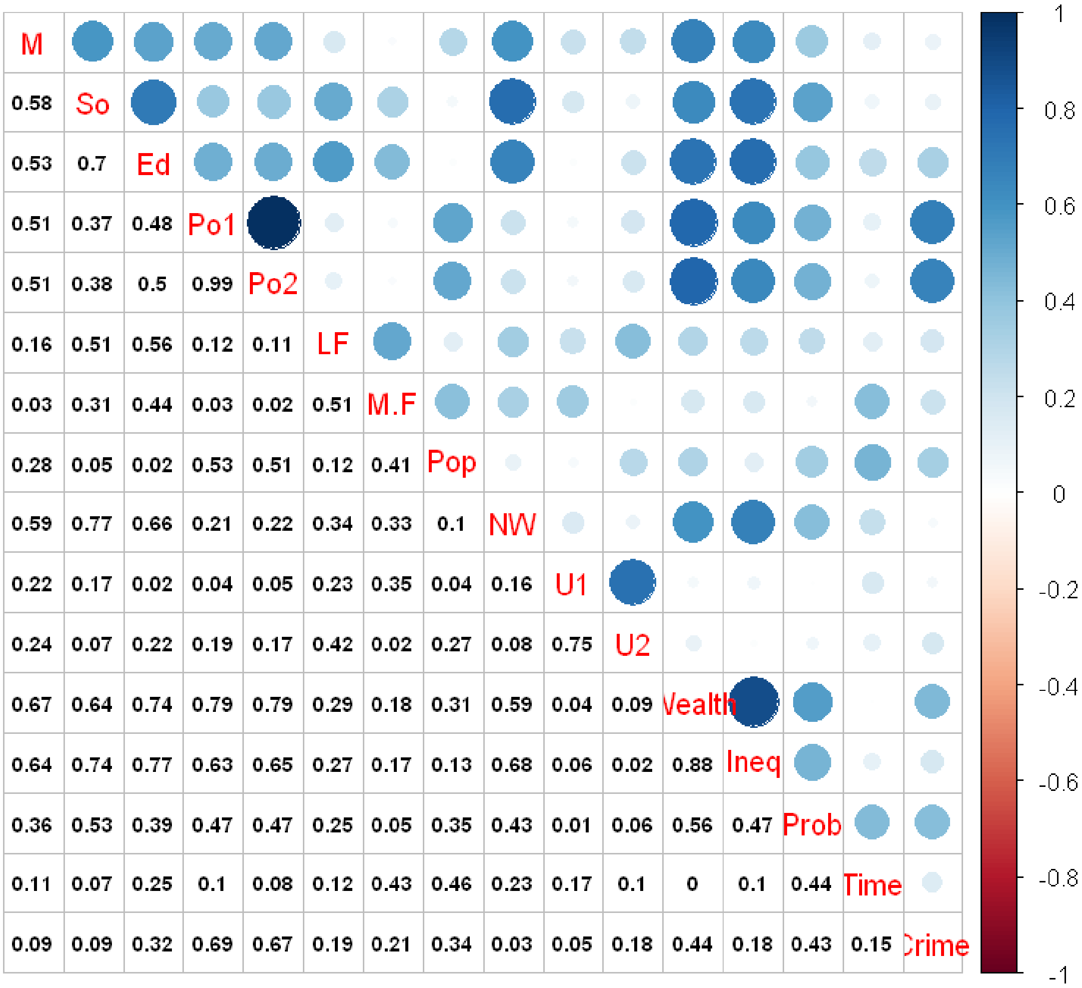


Use the following link for understanding the "corrplot" : [http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram (http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram)](http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram)

In [7]:
```
M<-cor(crime_df)
```

In [8]:
```
library(corrplot)
corrplot.mixed(abs(M), lower = "number", upper = "circle", lower.col = "black", number.cex = .7)
```

```
Warning message:
"package 'corrplot' was built under R version 3.6.3"corrplot 0.84 loaded
```



From the corrplot, it can be inferred that Crime response is more dependent on Po1, Po2, Pop, Wealth, Prob, and Ed than on the rest of input data

They are the inputs with more than 30% correlation to Crime response.

Next, I am going to create a test dataframe that I will use to predict the regression model. The test dataframe inputs the following predictors:

1. M = 14.0
2. So = 0
3. Ed = 10.0

4. Po1 = 12.0
5. Po2 = 15.5
6. LF = 0.640
7. M.F = 94.0
8. Pop = 150
9. NW = 1.1
10. U1 = 0.120
11. U2 = 3.6
12. Wealth = 3200
13. Ineq = 20.1
14. Prob = 0.04
15. Time = 39.0

I am going to fit the regression model with all the predictors. Then I will test the quality of this model by predicting the Crime rate for the baseline dataframe.

In [9]:
```
base_model <- lm(Crime ~. , data = crime_df)
```

In [10]:
```
summary(base_model)
```

```
Call:
lm(formula = Crime ~ ., data = crime_df)

Residuals:
    Min      1Q  Median      3Q     Max 
-395.74  -98.09   -6.69  112.99  512.67 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
M            8.783e+01  4.171e+01   2.106 0.043443 *  
So          -3.803e+00  1.488e+02  -0.026 0.979765    
Ed           1.883e+02  6.209e+01   3.033 0.004861 ** 
Po1          1.928e+02  1.061e+02   1.817 0.078892 .  
Po2         -1.094e+02  1.175e+02  -0.931 0.358830    
LF          -6.638e+02  1.470e+03  -0.452 0.654654    
M.F          1.741e+01  2.035e+01   0.855 0.398995    
Pop         -7.330e-01  1.290e+00  -0.568 0.573845    
NW           4.204e+00  6.481e+00   0.649 0.521279    
U1          -5.827e+03  4.210e+03  -1.384 0.176238    
U2           1.678e+02  8.234e+01   2.038 0.050161 .  
Wealth       9.617e-02  1.037e-01   0.928 0.360754    
Ineq         7.067e+01  2.272e+01   3.111 0.003983 ** 
Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *  
Time        -3.479e+00  7.165e+00  -0.486 0.630708    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 209.1 on 31 degrees of freedom
Multiple R-squared:  0.8031,	Adjusted R-squared:  0.7078 
F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

Understanding the Summary Output:

From the summary I am able to understand that Predictors such as M, Ed, Ineq, Prob are having p-value < 0.05 meaning they are statistically significant to be as a Predictor to the dependent variable(Crime).

Moreover, the R squared value is 0.8 meaning the model is a good fit for the data. But as we can see the Adjusted R squared is much less than the R squared value meaning the model is Overfitted and there are unwanted predictors which are misleading the value of R squared

In [11]:
```
# Creating a new dataframe to predict the observed crime rate in a city with the data that was provid

baseline_df <- data.frame(M = 14.0,
                          So = 0,
                          Ed = 10.0,
                          Po1 = 12.0,
                          Po2 = 15.5,
                          LF = 0.640,
                          M.F = 94.0,
                          Pop = 150,
                          NW = 1.1,
                          U1 = 0.120,
                          U2 = 3.6,
                          Wealth = 3200,
                          Ineq = 20.1,
                          Prob = 0.04,
                          Time = 39.0)
```

In [12]:
```
# Predicting the output of crime rate in the city for the given data

predict.lm(base_model,baseline_df)
```

1: 155.434896887443

The baseline regression model with all predictors has predicted a crime rate of 155.434896887446.

To reduce Overfitting probably we can split the data into train and test and fir the model on train and predict on test data.

We can also perform Cross Validation on the data to handle Overfitting.

Reference:http://www.sthda.com/english/articles/38-regression-model-validation/157-cross-validation-essentials-in-r/ (http://www.sthda.com/english/articles/38-regression-model-validation/157-cross-validation-essentials-in-r/)