ISYE Homework - Week 3 Markdown

2024-06-05

Question 7.2

We'll start by reading in our data and taking a look at it. Notice the class of temps is just a dataframe, so we'll want to convert it into a time series object in order to do Holt Winters on it.

```
temps <- read.table("~/Documents/ISYE 6501/week_3_Homework-summer/week 3 data-summer/temps.txt", string
head(temps)
```

```
##     DAY X1996 X1997 X1998 X1999 X2000 X2001 X2002 X2003 X2004 X2005 X2006 X2007
## 1 1-Jul    98    86    91    84    89    84    90    73    82    91    93    95
## 2 2-Jul    97    90    88    82    91    87    90    81    81    89    93    85
## 3 3-Jul    97    93    91    87    93    87    87    87    86    86    93    82
## 4 4-Jul    90    91    91    88    95    84    89    86    88    86    91    86
## 5 5-Jul    89    84    91    90    96    86    93    80    90    89    90    88
## 6 6-Jul    93    84    89    91    96    87    93    84    90    82    81    87
##   X2008 X2009 X2010 X2011 X2012 X2013 X2014 X2015
## 1    85    95    87    92   105    82    90    85
## 2    87    90    84    94    93    85    93    87
## 3    91    89    83    95    99    76    87    79
## 4    90    91    85    92    98    77    84    85
## 5    88    80    88    90   100    83    86    84
## 6    82    87    89    90    98    83    87    84
```

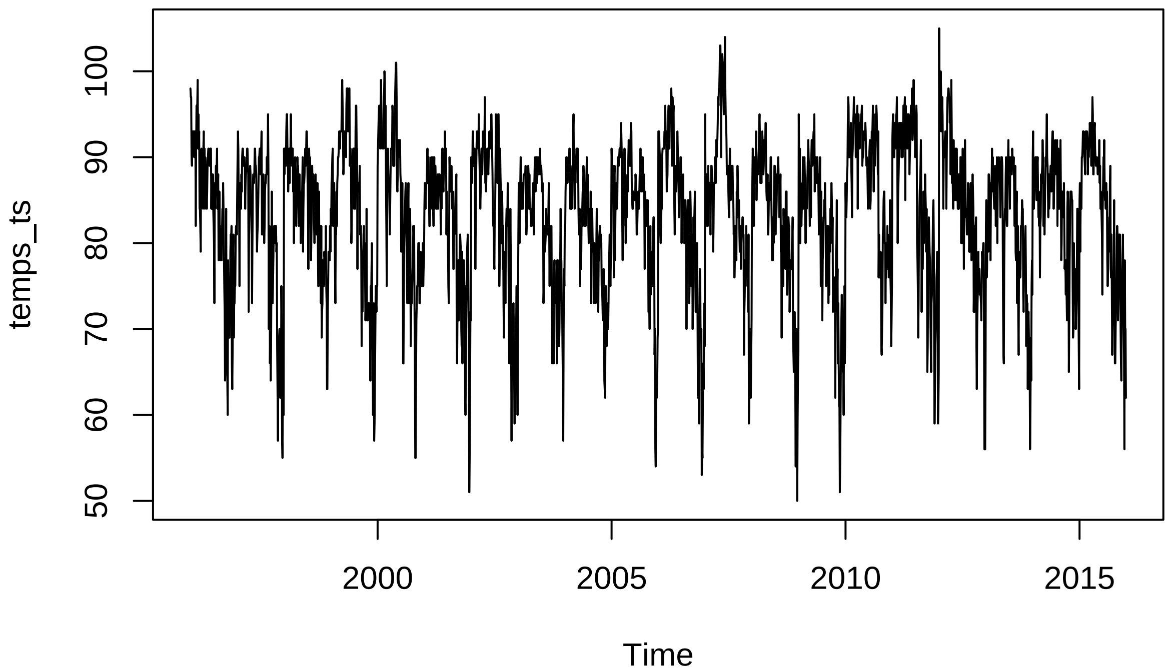
```
class(temps)
```

```
## [1] "data.frame"
```

Now, we'll convert temps into a time series object. We'll take all the temperature observations and put them in a vector, then use that vector to create the ts object.

When we create our time series object, we set the frequency to 123 since there's 123 observations per year. We'll start at 1996 because that is the first year of our data!

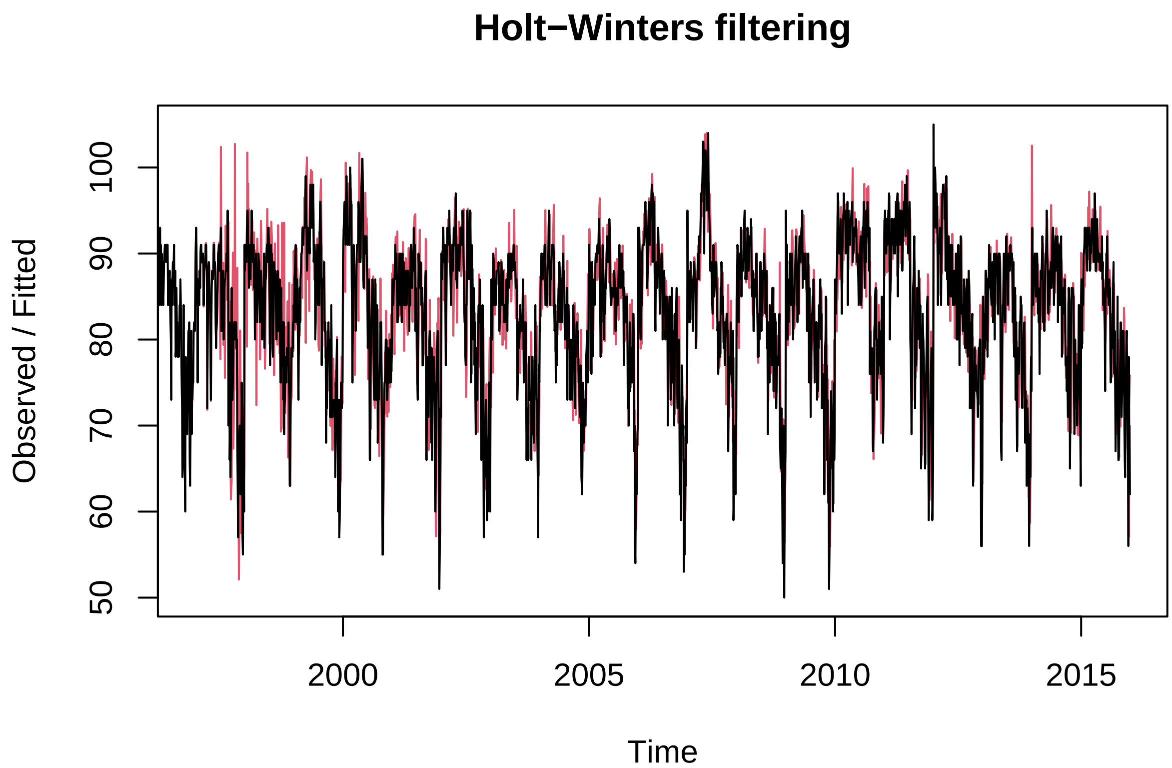
```
#turn our temp values into a vector
temps_vector <- as.vector(unlist(temps[,2:21]))

#then we take our temp values vector and us the ts function to create the time series object
temps_ts <- ts(temps_vector, start=1996, frequency=123)
plot(temps_ts) #we can see the data with all the random variance here
```



Next, we'll perform exponential smoothing on the data using the Holt Winters function. We will try setting the parameter "seasonal" to both "multiplicative" and "additive" in order to see whether one model performs better than the other.

My hunch is that temperature fluctuates in a relatively constant manner; it does not strike me as true that the temperature gets hotter and colder proportional to the extent of the initial warmth or cold (i.e., 90 degrees plus 1/10 of 90 = 99; 80 degrees plus 1/10 of 80 = 88… etc.) So I'm expecting an additive model will be more accurate, but we will verify this by checking the SSE for both models.

```
##              Length Class  Mode
## fitted       9348   mts    numeric
## x            2460   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients  125   -none- numeric
## seasonal        1   -none- character
## SSE             1   -none- numeric
## call            3   -none- call
```




```
## [1] 68904.57
```

```
##    alpha
## 0.615003
```

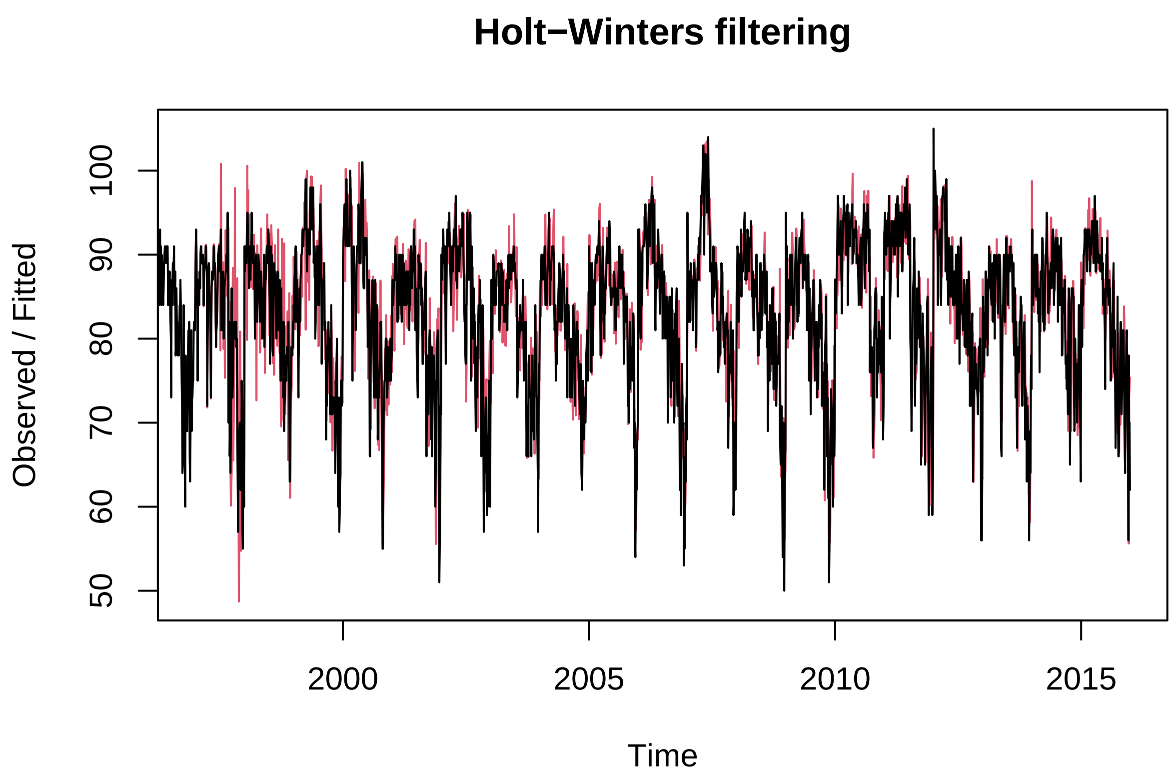
```
## beta
##    0
```

```
##     gamma
## 0.5495256
```

```
temps_hw2 <- HoltWinters(temps_ts, seasonal = "additive")
print(summary(temps_hw2))
```

```
##              Length Class  Mode
## fitted       9348   mts    numeric
## x            2460   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients  125   -none- numeric
## seasonal        1   -none- character
## SSE             1   -none- numeric
## call            3   -none- call
```

```r
plot(temps_hw2)
```




```r
print(temps_hw2$SSE) #SSE of 66244.25
```

```
## [1] 66244.25
```

```r
print(temps_hw2$alpha) #alpha of 0.661
```

```
##     alpha
## 0.6610618
```

```r
print(temps_hw2$beta) #beta of 0
```

```
## beta
##    0
```

```r
print(temps_hw2$gamma) #gamma of 0.625
```

```
##     gamma
## 0.6248076
```

It looks like the additive model is indeed a slightly better model, with a slightly smaller SSE value. Now we can discuss the values for alpha, beta, and gamma, as well as the coefficients and fitted values.

For our additive model, we have alpha as 0.661, beta as 0, and gamma as 0.625. Our alpha value is slightly higher here than in the multiplicative seasonality model, meaning that the model prioritized more recent observations slightly more when compared to the multiplicative model.

In both models, our beta was 0, suggesting that our model did not detect any notable trends.

Finally, our gamma value is 0.625, which indicates that our model slightly prioritized more recent seasonal observations compared to older ones; this is also true when comparing the additive model to the seasonal model (which found a gamma value of 0.55).

Let's take a look now at our fitted values.

```r
print(summary(temps_hw2))
```

```
##              Length Class  Mode
## fitted       9348   mts    numeric
## x            2460   ts     numeric
## alpha           1   -none- numeric
## beta            1   -none- numeric
## gamma           1   -none- numeric
## coefficients  125   -none- numeric
## seasonal        1   -none- character
## SSE             1   -none- numeric
## call            3   -none- call
```

```r
print(head(temps_hw2$fitted))
```

```
##          xhat    level        trend    season
## [1,] 87.17619 82.87739 -0.004362918  4.303159
## [2,] 90.32925 82.09550 -0.004362918  8.238119
## [3,] 92.96089 81.87348 -0.004362918 11.091777
## [4,] 90.93360 81.89497 -0.004362918  9.042997
## [5,] 83.99752 81.93450 -0.004362918  2.067387
## [6,] 84.04358 81.93177 -0.004362918  2.116168
```

```r
#commented out below since it was incredibly long
#print(temps_hw2$fitted)
```

It was commented out for space concerns, but when we call print(temps_hw2$fitted), we can see that our fitted "start date" is 1997. This is because our model is essentially using the first year of data to "train" itself before beginning the exponential smoothing on the remaining data.

Our xhat values are our predicted values AFTER we've done the smoothing. This allows us to ignore the noise of the dataset and ultimately answer the question at hand - are summers ending LATER compared to 20 years ago?

From here, we'll move into Excel to use CUSUM on our xhat values before continuing the discussion below.

```r
#to copy to our clipboard for pasting into Excel
#commented out for saving as PDF

#library(clipr)
#write_clip(temps_hw2$fitted)
```

In Excel, I took our xhat values for the smoothed data, divided that data up and gave each year of data its own sheet. I then ran the CUSUM change detection model on each year to find each year's end of summer (using the month of July as the control value for each year), and I set C = 1 standard deviation and T = 4 standard deviations.

A quick note here: I made sure each year had its own control value from its own first 31 days. Had I used the same control value for each year, we may have gotten different results. I wanted to ensure each year's "end of summer" was calculated relative to its own temperatures.

Another note - by using Holt Winters on the entire dataset and THEN splitting up the smoothed data - rather than partitioning out the data and THEN smoothing it - we ensured that the seasonality remained intact. Had we split up the data first and THEN smoothed each year, our models would have missed out on seasonal patterns that informed the smoothing for subsequent years.

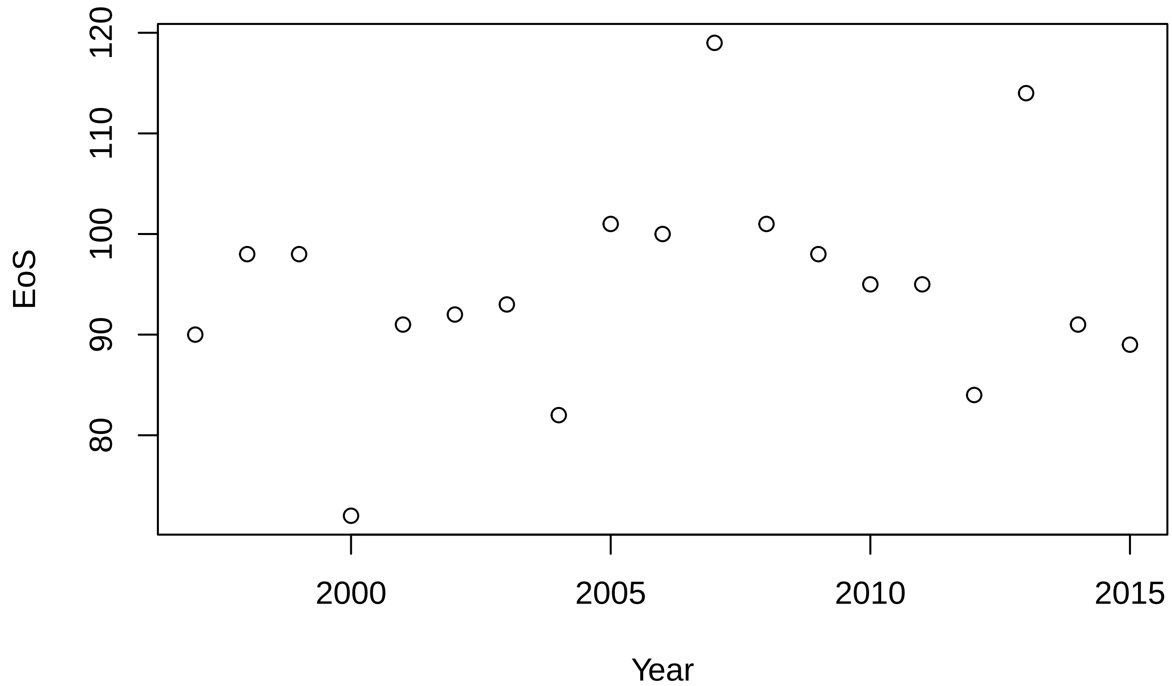
Now back to our main task. Below, I created a table with the years and their respective End of Summer dates (found numerically by their index-1 from my Excel file). I plotted those below and ran a regression model to see if we could fit a line to show an upward trajectory over time.

As shown in the model and the plots, we do NOT see evidence that summers are ending later on average than they did 20 years ago.

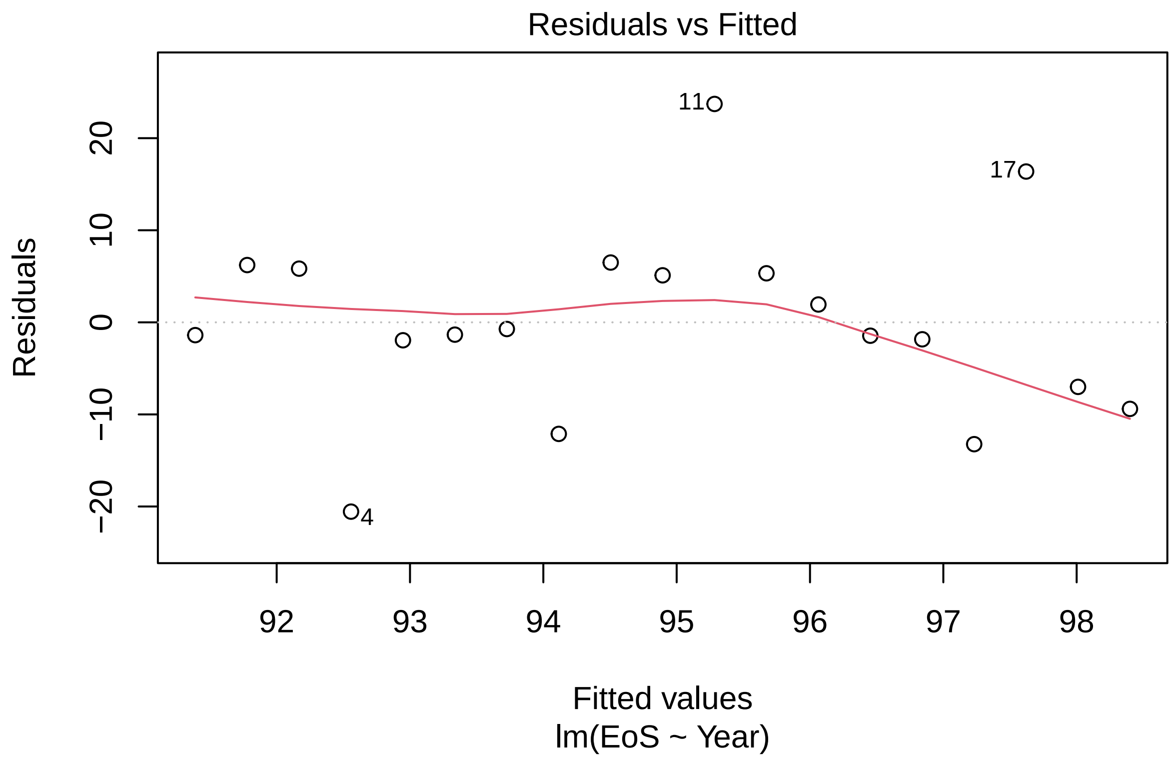
```
my_table <- data.frame(
  Year = c(1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 20
  EoS = c(90, 98, 98, 72, 91, 92, 93, 82, 101, 100, 119, 101, 98, 95, 95, 84, 114, 91, 89))

print(my_table)
```

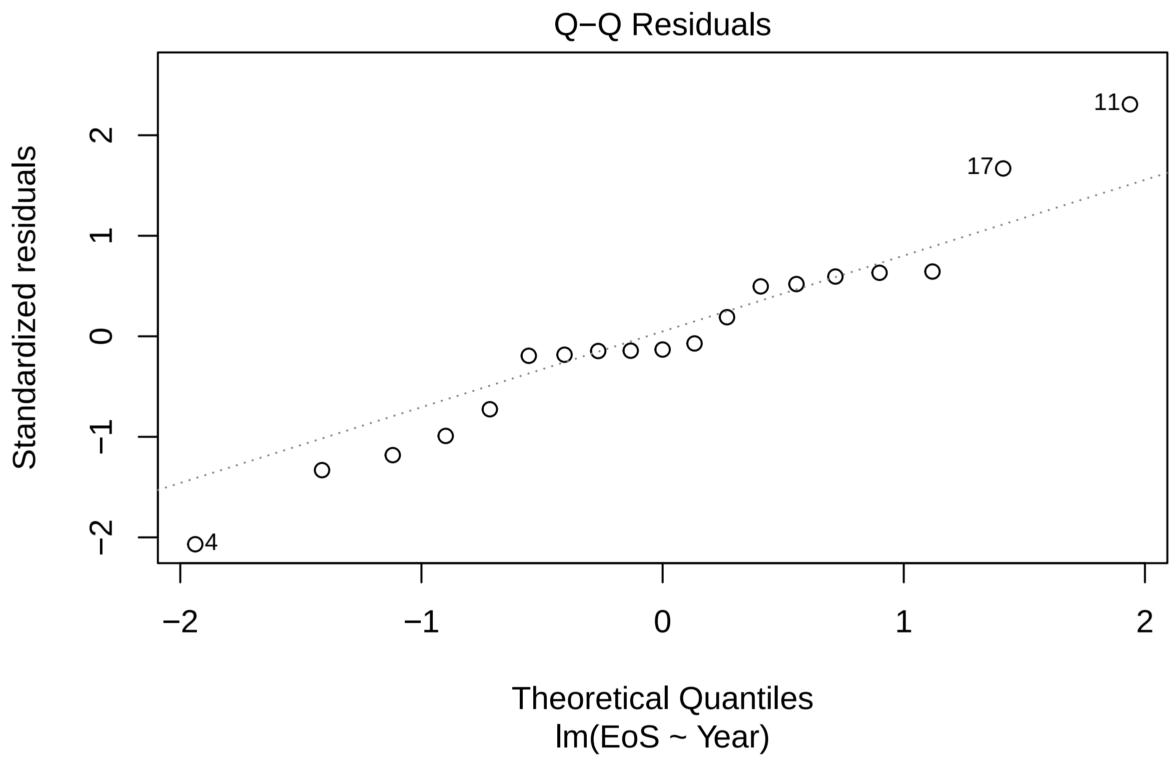
```
##    Year EoS
## 1  1997  90
## 2  1998  98
## 3  1999  98
## 4  2000  72
## 5  2001  91
## 6  2002  92
## 7  2003  93
## 8  2004  82
## 9  2005 101
## 10 2006 100
## 11 2007 119
## 12 2008 101
## 13 2009  98
## 14 2010  95
## 15 2011  95
## 16 2012  84
## 17 2013 114
## 18 2014  91
## 19 2015  89
```

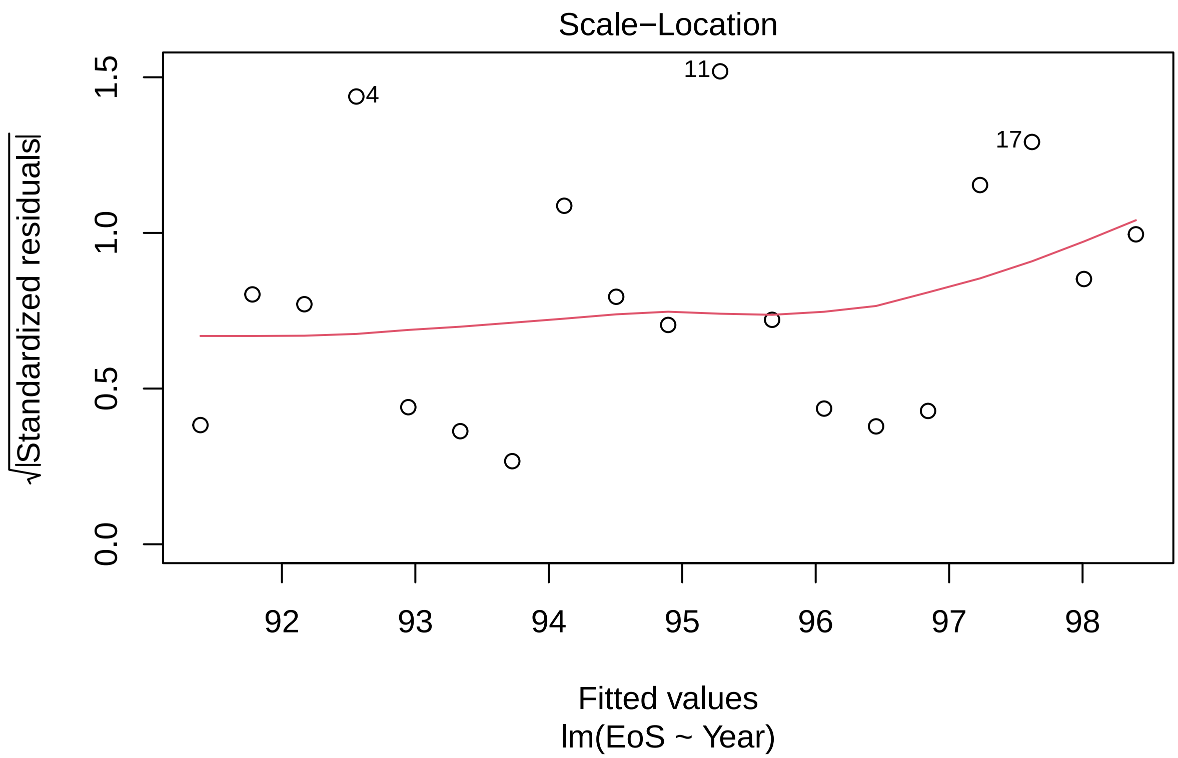
```
plot(my_table)
```

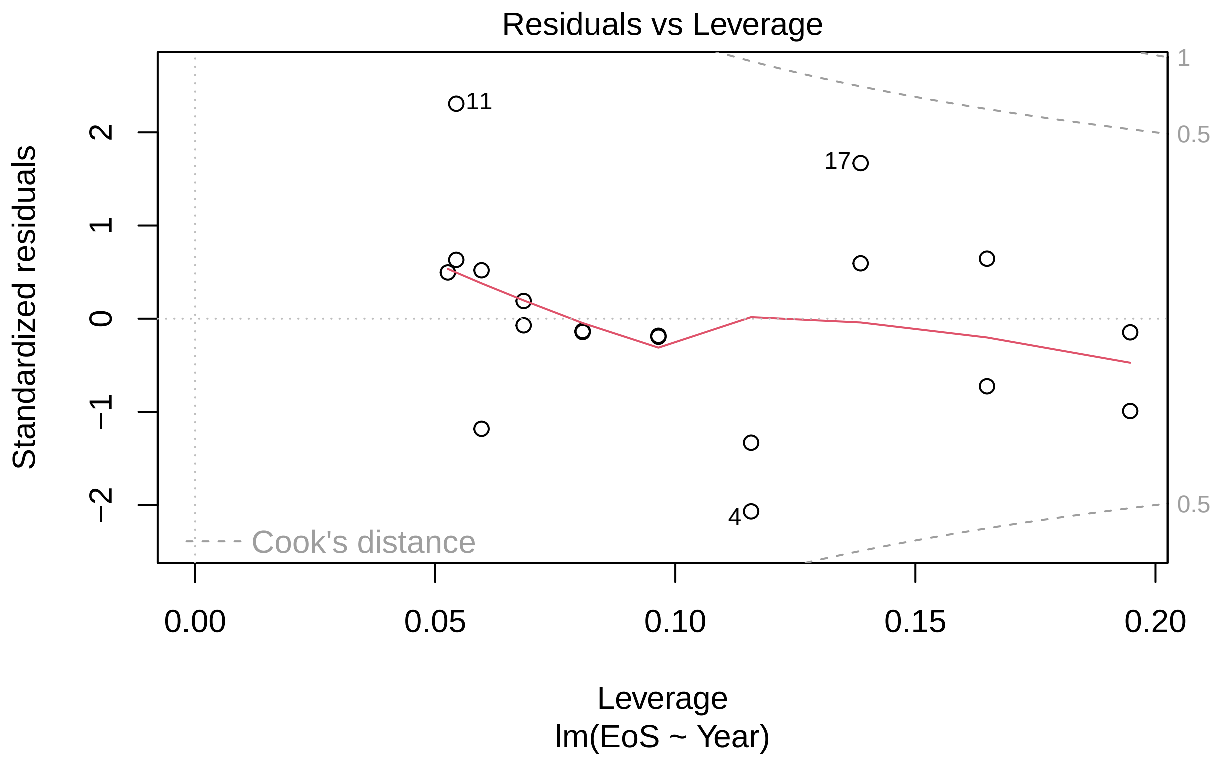



```
model <- lm(EoS ~ Year, data=my_table)
plot(model)
```


Residuals vs Fitted
Residuals
20
10
0
−10
−20
11
17
4
92
93
94
95
96
97
98
Fitted values
lm(EoS ~ Year)


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(EoS ~ Year)
11
17
4
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
4
11
17
92
93
94
95
96
97
98
Fitted values
lm(EoS ~ Year)




```r
summary(model)
```

```
##
## Call:
## lm(formula = EoS ~ Year, data = my_table)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.558  -4.479  -1.337   5.579  23.716
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -686.3895   888.1175  -0.773    0.450
## Year           0.3895     0.4427   0.880    0.391
##
## Residual standard error: 10.57 on 17 degrees of freedom
## Multiple R-squared:  0.04354,    Adjusted R-squared:  -0.01272
## F-statistic: 0.7739 on 1 and 17 DF,  p-value: 0.3913
```