Assignment 1: Forecasting / Financial Data – Fundamental Concepts

Andrew G. Dunn

January 11, 2016

Andrew G. Dunn, Northwestern University Predictive Analytics Program

Prepared for PREDICT-413: Time Series Analysis and Forecasting.

Formatted using the LaTeX, via pandoc and R Markdown. References managed using pandoc-citeproc.

Setup

```r
require(fBasics)    # for calculations
require(quantmod)   # for returns calculations
require(fpp)        # for data
require(knitr)      # for table output
require(ggplot2)    # for graphing
require(ggthemes)   # for graphing beautifully
require(gridExtra)  # for laying out graphs
```

Part 1

Consider the daily simple returns of Netflix (NFLX) stock, Center for Research In Security Prices (CRSP) value-weighted index (VW), CRSP equal-weighted index (EW), and the S&P composite index (SP) from January 2, 2009 to December 31, 2013. Returns of the three indices include dividends. The data are within the file `d-nflx3dx0913.txt` and the columns show permno, date, nflx, vw, ew, and sp, respectively, with the last for columns showing the simple returns.

```r
d1 = read.table("data/d-nflx3dx0913.txt", header=T)
head(d1)
```

```
##   PERMNO     date      nflx    vwretd    ewretd    sprtrn
## 1  89393 20090102 -0.000669  0.030501  0.038274  0.031608
## 2  89393 20090105  0.069300 -0.000580  0.016764 -0.004668
## 3  89393 20090106  0.031309  0.011297  0.033647  0.007817
## 4  89393 20090107 -0.006982 -0.030489 -0.022271 -0.030010
## 5  89393 20090108  0.013452  0.006283  0.011896  0.003397
## 6  89393 20090109 -0.026848 -0.022410 -0.018748 -0.021303
```

Part A

Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of each simple return series.

```r
d1b = basicStats(d1)
kable(d1b[c('Mean', 'Stdev', 'Skewness', 'Kurtosis', 'Minimum', 'Maximum'), -(1:2)],
      caption='Basic Statistics of the Simple Return Series')
```

Table 1: Basic Statistics of the Simple Return Series

	nflx	vwretd	ewretd	sprtrn
Mean	0.002733	0.000743	0.001049	0.000645
Stdev	0.038541	0.012554	0.012138	0.012263
Skewness	0.930459	-0.194593	-0.191914	-0.155751
Kurtosis	21.884230	3.879047	4.354323	4.091064
Minimum	-0.348957	-0.068664	-0.072385	-0.066634
Maximum	0.422235	0.069054	0.064792	0.070758

Part B

Transform the simple return to log returns. Compute the sample mean, standard deviation, skewness, excess kurtosis, minimum, and maximum of each log return series.

```
d1l = log(d1[,-(1:2)]+1)  # Log Transform, +1 as an offset so that we don't compute log(0)
d1bl = basicStats(d1l)
kable(d1bl[c('Mean', 'Stdev', 'Skewness', 'Kurtosis', 'Minimum', 'Maximum'),],
      caption='Basic Statistics of the Log Transformed Simple Return Series')
```

Table 2: Basic Statistics of the Log Transformed Simple Return Series

	nflx	vwretd	ewretd	sprtrn
Mean	0.001996	0.000664	0.000975	0.000569
Stdev	0.038373	0.012566	0.012145	0.012272
Skewness	-0.434692	-0.304465	-0.307441	-0.266958
Kurtosis	23.549867	3.915422	4.455771	4.096303
Minimum	-0.429180	-0.071135	-0.075139	-0.068958
Maximum	0.352230	0.066774	0.062779	0.068367

Part C

Test the null hypothesis that the mean of the log returns of NFLX stock is zero.

```
t.test(d1l$nflx)
```

```
##
##  One Sample t-test
##
## data:  d1l$nflx
## t = 1.8449, df = 1257, p-value = 0.06528
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0001264844  0.0041185915
## sample estimates:
##   mean of x
## 0.001996054
```

Fail to reject the null hypothesis at a 0.05 level.

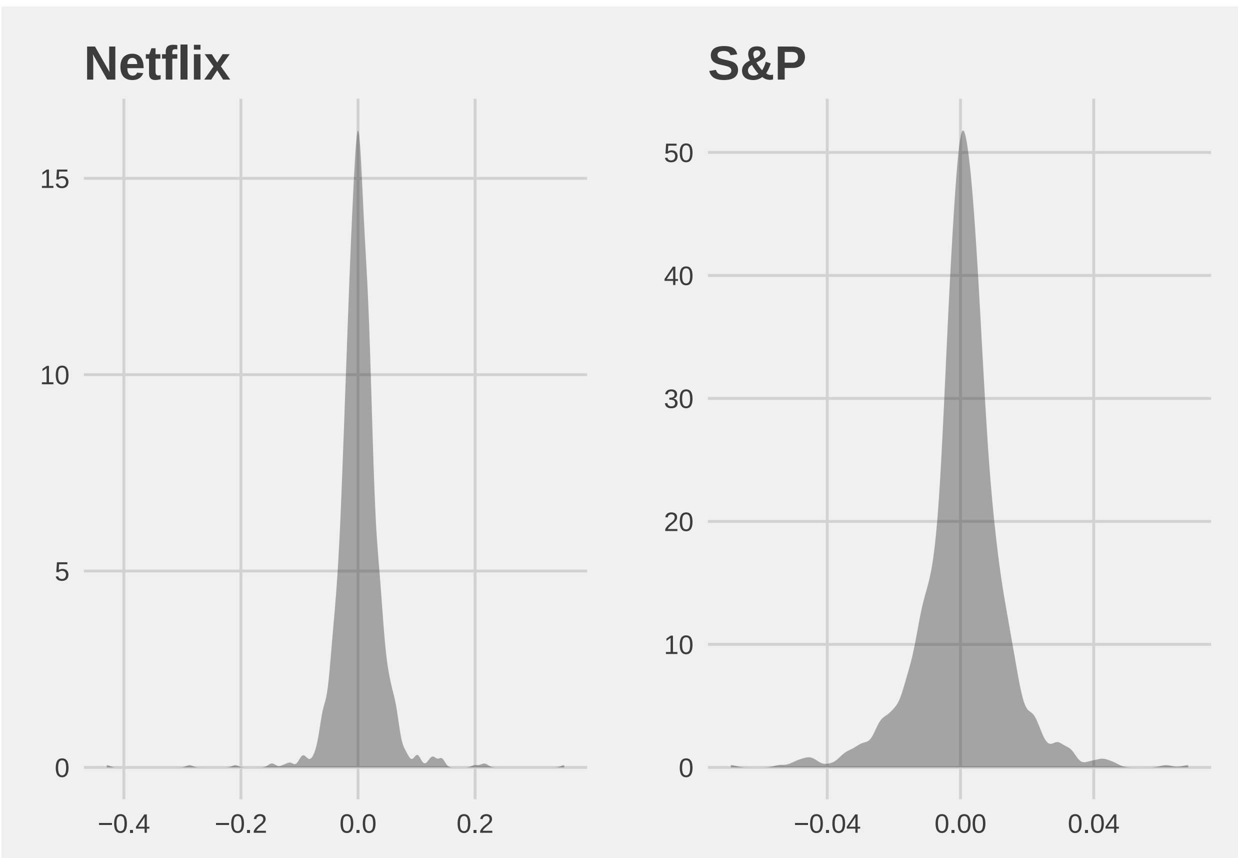
Part D

Obtain the empirical density plot of the daily log returns of Netflix stock and the S&P composite index.

```
pnflx = ggplot(d1l, aes(nflx)) +
  stat_density(alpha = 0.4) +
  labs(x="Returns", y="Density") +
  ggtitle("Netflix") + theme_fivethirtyeight()

psprtrn = ggplot(d1l, aes(sprtrn)) +
  stat_density(alpha = 0.4) +
  labs(x="Returns", y="Density") +
  ggtitle("S&P") + theme_fivethirtyeight()

grid.arrange(pnflx, psprtrn, ncol=2)
```



Part 2

Consider the monthly log returns of General Electric (GE) stock from January 1981 to December 2013. The original data are monthly returns for GE stock, CRSP value-weighted index (VW), CRSP equal-weighted index (EW), and S&P composite index (SP) from January 1981 to December 2013. The returns include dividend distributions. The data are within the file `m-ge3dx8113.txt` and the columns show permno, date, ge, vwretd, ewretd, and sprtrn, respectively. Perform tests and draw conclusions using the 5% significance level.

```r
d2 = read.table("data/m-ge3dx8113.txt", header=T)
head(d2)
```

```
##   PERMNO     date        ge    vwretd    ewretd    sprtrn
## 1  12060 19810130  0.000000 -0.040085  0.005615 -0.045742
## 2  12060 19810227  0.089796  0.015521  0.002150  0.013277
## 3  12060 19810331  0.014981  0.046184  0.072674  0.036033
## 4  12060 19810430 -0.020522 -0.011268  0.027885 -0.023456
## 5  12060 19810529  0.001905  0.013551  0.027187 -0.001657
## 6  12060 19810630 -0.046768 -0.010242 -0.013194 -0.010408
```

Part A

Construct a 95% confidence interval for the monthly log returns of GE stock.

```r
# This seems the be the way that is presented in the example code
d2l = log(d2[,-(1:2)]+1)  # Log Transform, +1 as an offset so that we don't compute log(0)
t.test(d2l$ge)
```

```
##
##  One Sample t-test
##
## data:  d2l$ge
## t = 2.8708, df = 395, p-value = 0.004315
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.003248467 0.017365132
## sample estimates:
## mean of x
## 0.0103068
```

Per the t test output above, a 95% cofidence interval is (0.003248467,0.017365132)

Part B

Test $H_0 : m_3 = 0$ versus $H_a : m_3 \neq 0$, where m_3 denotes the skewness of the return.

Test algorithm found on page 26.

```r
st = skewness(d2l$ge) / sqrt(6 / length(d2l$ge))  # compute skewness test
paste(2*(1-pnorm(abs(st))))  # computing the p-value
```

```
## [1] "5.81316366377038e-07"
```

Fail to reject the Null of Symmetry

Part C

Test $H_0 : K = 3$ versus $H_a : K! = 3$, where K denotes the kurtosis.

```
kt = kurtosis(d2l$ge) / sqrt(24 / length(d2l$ge))  # compute kurtosis test
paste(2*(1-pnorm(abs(kt))))
```

```
## [1] "0"
```

Reject null hypothesis at a 0.05 level.

Part 3

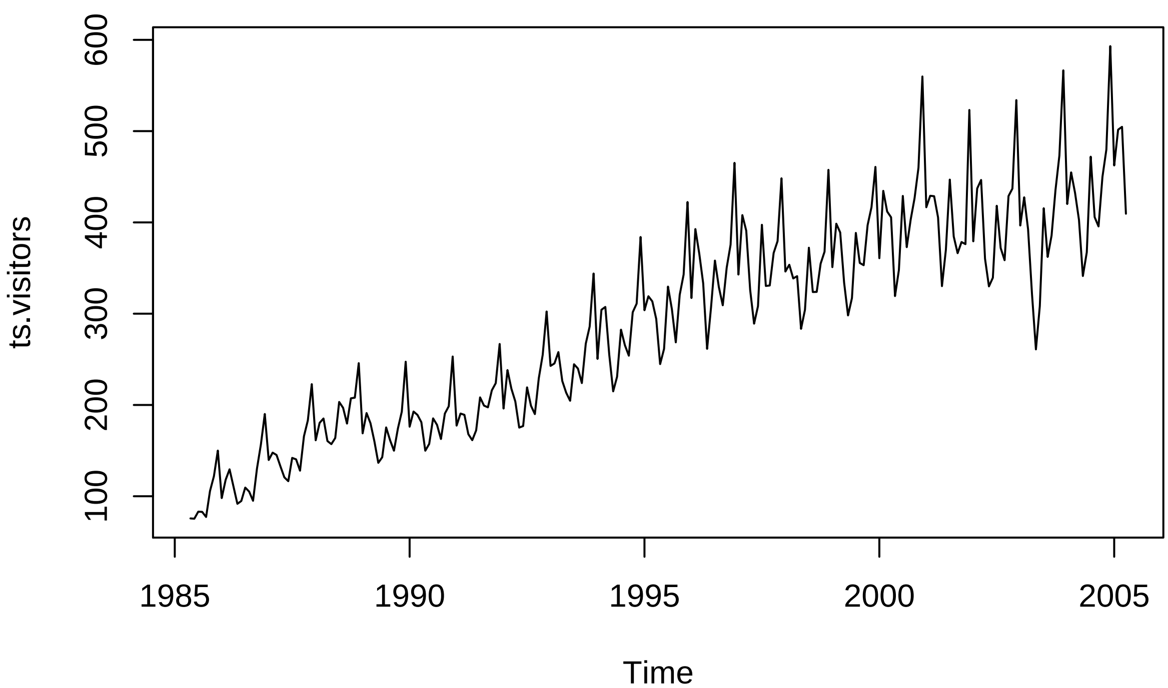
For this, use the monthly Australian short-term overseas visitors data from May 1985 to April 2005 from `Forecasting: principles and practice` the Hyndeman and Athanasopoulos text.

```
ts.visitors = visitors  # comes from fpp package
df.visitors = as.data.frame(visitors)
```

Part A

Make a time plot of your data and describe the main features of the series.

```
plot(ts.visitors)
```



The series appears to have an upward trend and a monthly seasonal component. The series appears to peak around Feb or March of each year.

Part B

Forecast the next two years using Holt-Winters' multiplicative method.

```
aust = window(visitors)
fit_multi = hw(aust, seasonal="multiplicative")
print(fit_multi)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## May 2005       357.9775 333.0657 382.8893 319.8783 396.0767
## Jun 2005       388.1893 355.6186 420.7601 338.3766 438.0020
## Jul 2005       472.5084 427.1341 517.8827 403.1145 541.9024
## Aug 2005       425.8257 380.3844 471.2669 356.3293 495.3220
## Sep 2005       422.5612 373.3933 471.7292 347.3654 497.7571
## Oct 2005       477.2950 417.5369 537.0531 385.9029 568.6871
## Nov 2005       502.0552 435.0774 569.0329 399.6215 604.4888
## Dec 2005       616.7458 529.7315 703.7601 483.6689 749.8227
## Jan 2006       460.6378 392.3153 528.9602 356.1476 565.1279
## Feb 2006       513.2142 433.5758 592.8526 391.4178 635.0106
## Mar 2006       502.5685 421.3040 583.8329 378.2853 626.8517
## Apr 2006       443.4652 368.9948 517.9356 329.5725 557.3578
## May 2006       377.8345 312.1256 443.5435 277.3414 478.3277
## Jun 2006       409.6231 336.0362 483.2100 297.0817 522.1646
## Jul 2006       498.4784 406.1751 590.7817 357.3126 639.6442
## Aug 2006       449.1232 363.5640 534.6824 318.2717 579.9747
## Sep 2006       445.5752 358.3943 532.7562 312.2435 578.9070
## Oct 2006       503.1725 402.2090 604.1361 348.7621 657.5830
## Nov 2006       529.1527 420.4123 637.8931 362.8486 695.4568
## Dec 2006       649.8845 513.2752 786.4937 440.9586 858.8103
## Jan 2007       485.2782 381.0489 589.5076 325.8732 644.6833
## Feb 2007       540.5452 422.0364 659.0540 359.3017 721.7888
## Mar 2007       529.2143 410.8910 647.5377 348.2544 710.1743
## Apr 2007       466.8740 360.5104 573.2375 304.2050 629.5430
```

Part C

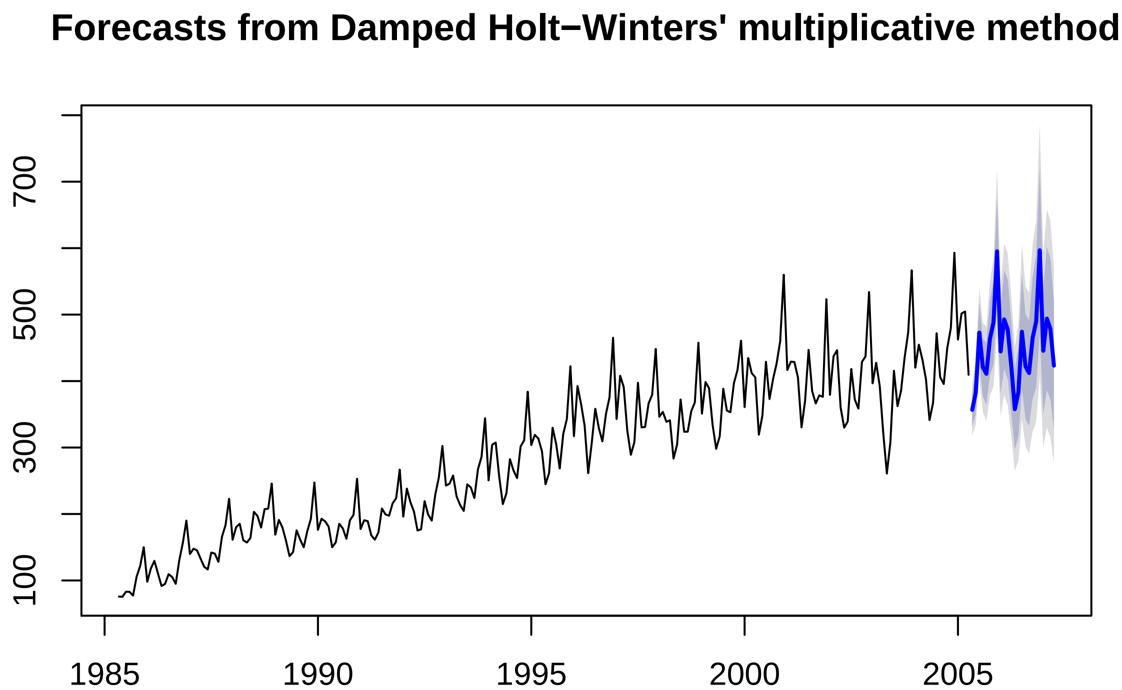
Why is multiplicative seasonality necessary here?

Multiplicative method is preferred when the seasonal variations are changing proportionally to the level of the series. In this series, it appears that the variations are growing.

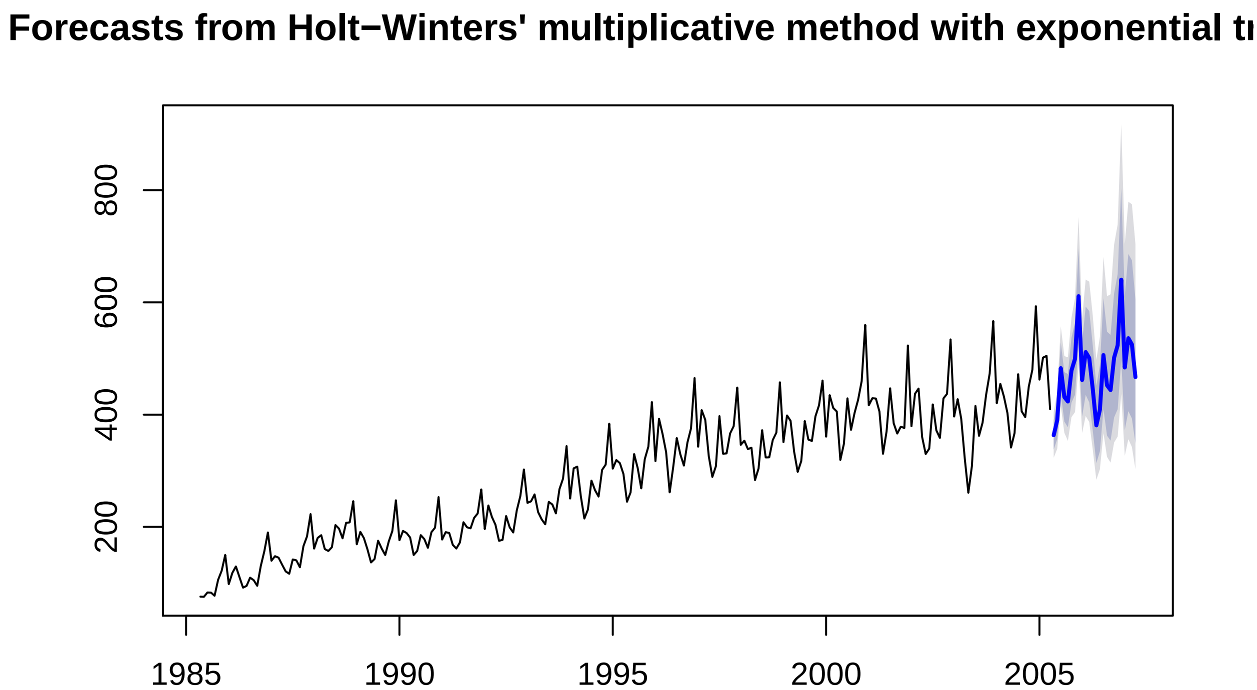
Part D

Experiment with making the trend exponential and/or damped.

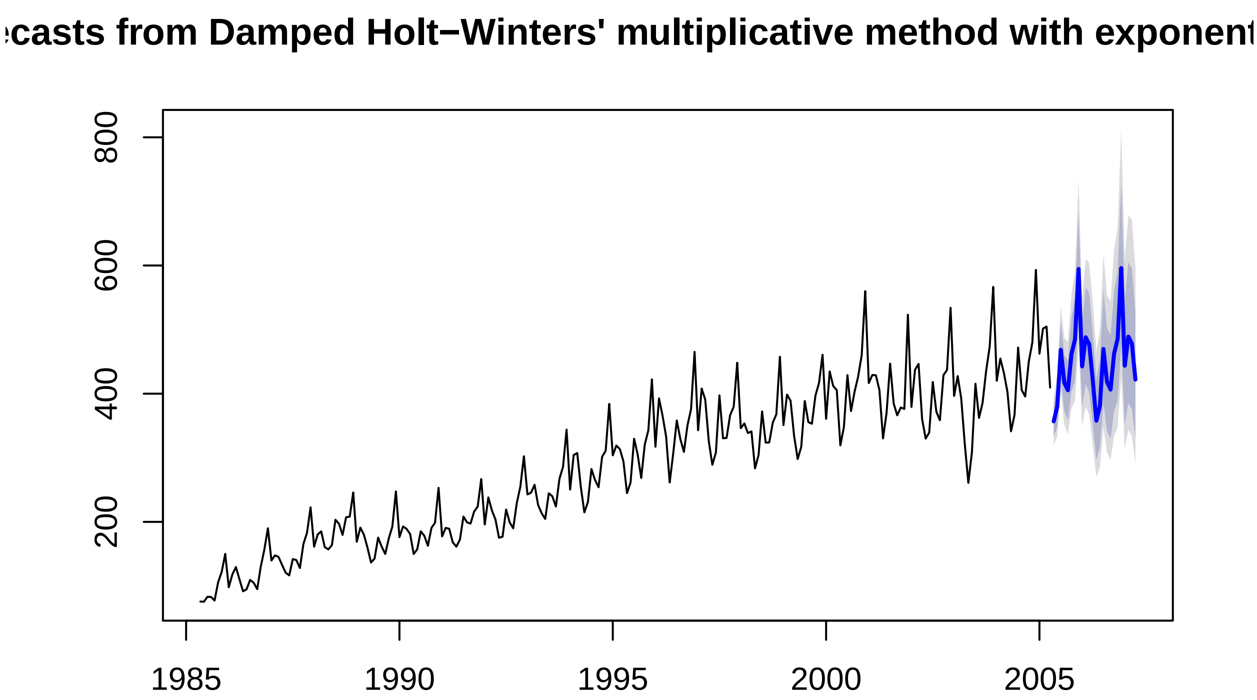
```
fit_multi_damped = hw(aust, seasonal="multiplicative", damped=TRUE)
plot(forecast(fit_multi_damped))
```




```r
fit_multi_exp = hw(aust, seasonal="multiplicative", exponential=TRUE)
plot(forecast(fit_multi_exp))
```




```r
fit_multi_exp_damped = hw(aust, seasonal="multiplicative", exponential=TRUE, damped=TRUE)
plot(forecast(fit_multi_exp_damped))
```



Part E

Compare the RMSE of the one-step forecasts from the various methods. Which is preferred?

```r
accuracy(fit_multi)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.8614726 14.52211 10.86884 -0.4799156 4.168399 0.4013761
##                     ACF1
## Training set -0.03448764
```

```r
accuracy(fit_multi_damped)
```

```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 1.523643 14.40219 10.64283 0.3591333 4.057262 0.3930297
##                    ACF1
## Training set 0.01526565
```

```r
accuracy(fit_multi_exp)
```

```
##                      ME    RMSE      MAE        MPE     MAPE     MASE
## Training set -0.6175624 14.6899 11.00618 -0.3558085 4.230296 0.406448
##                    ACF1
## Training set 0.08654357
```

```r
accuracy(fit_multi_exp_damped)
```

```
##                     ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.5595893 14.46091 10.66091 -0.07611252 4.075176 0.3936972
##                    ACF1
## Training set -0.0268311
```

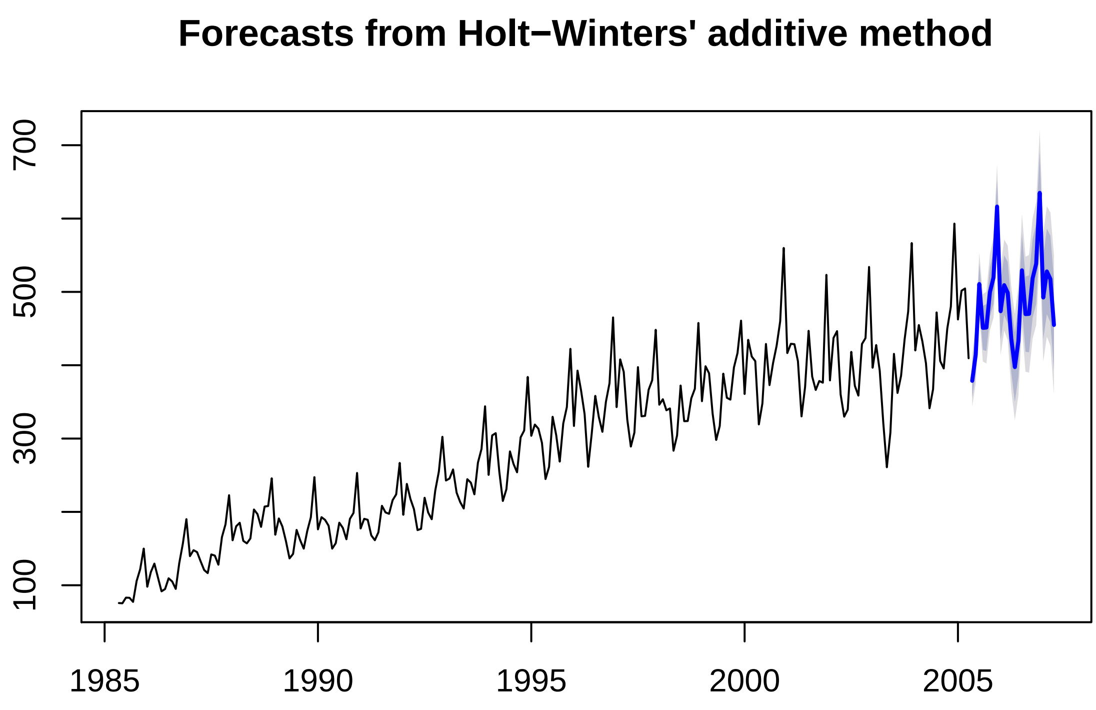
It appears that the lowest RMSE was within tthe Multiplicative and Damped model, which fit the data best.

Part F

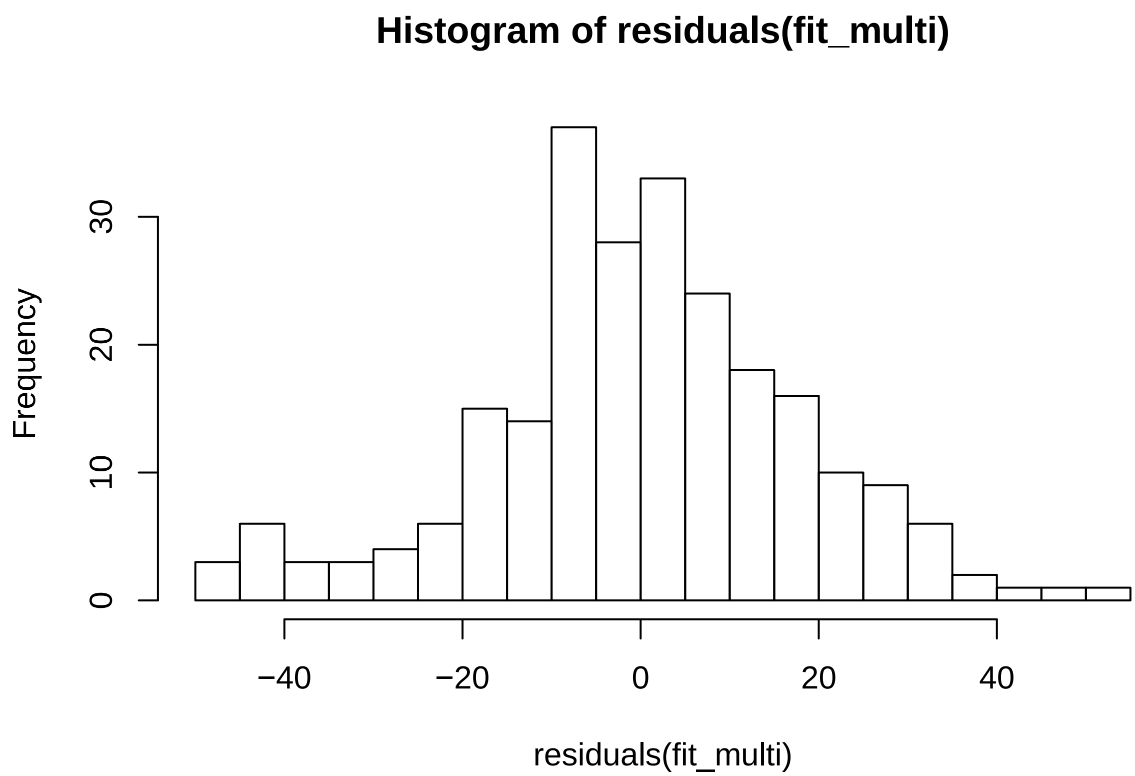
Fit each of the following models to the same data, examine the residual diagnostics and compare the forecasts for the next two years:

Multiplicative Holt-Winters' Method

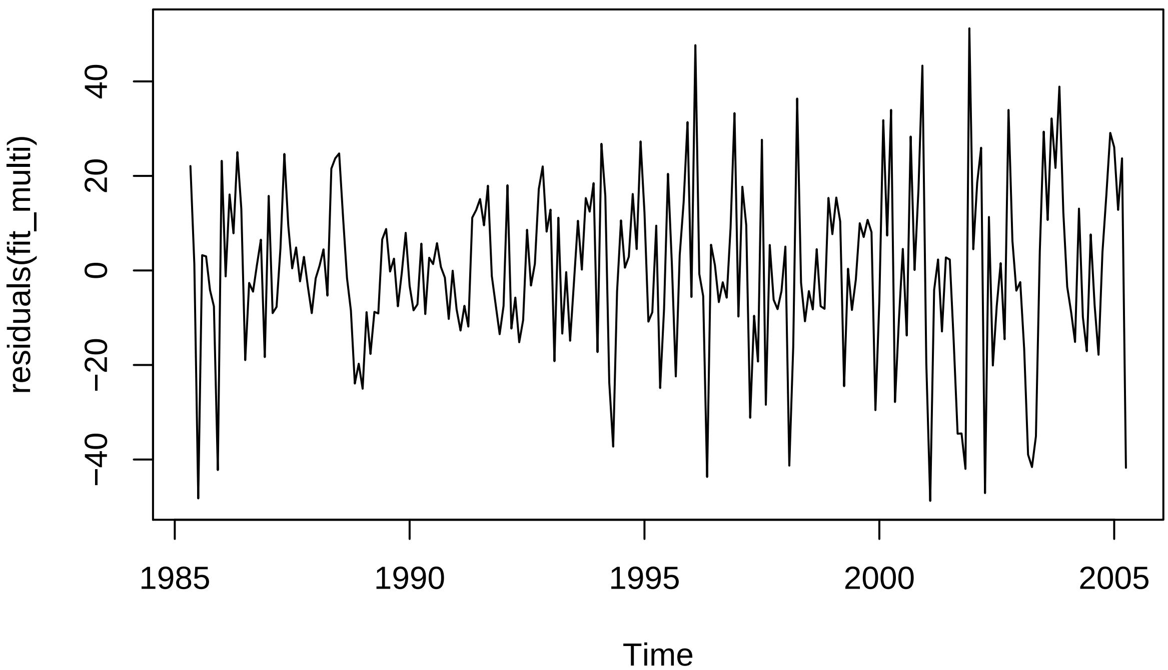
```r
fit_multi = hw(aust, multiplicative=TRUE)
plot(fit_multi)
```



```r
hist(residuals(fit_multi), nclass=20)
```




```r
plot(residuals(fit_multi))
```



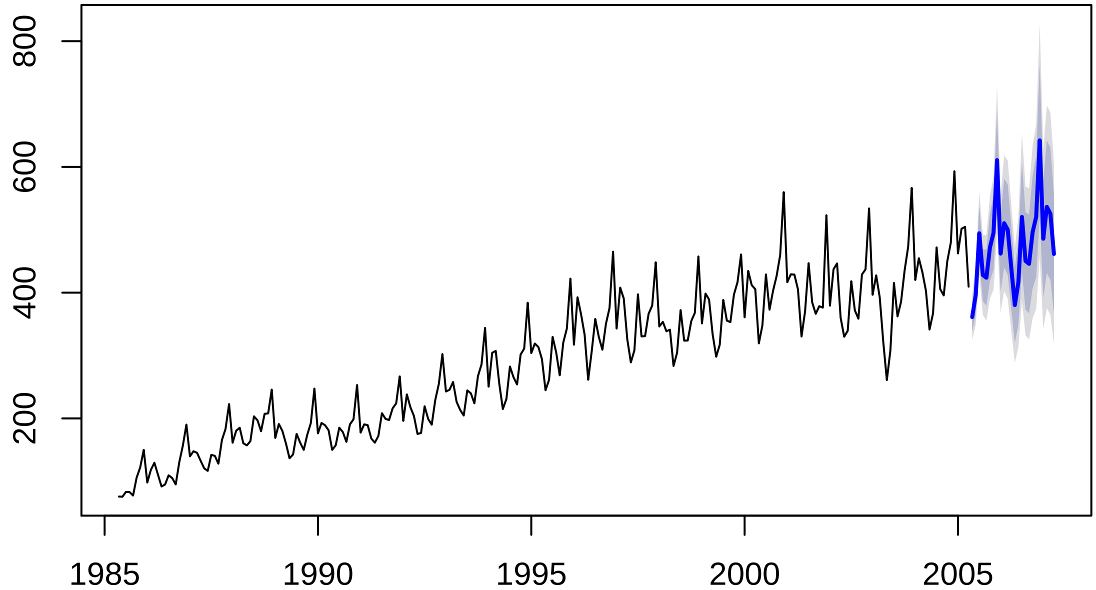

```r
accuracy(fit_multi)
```

```
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.0893009 17.96425 13.68053 -0.2196562 5.342406 0.5052092
##                   ACF1
## Training set 0.1284181
```

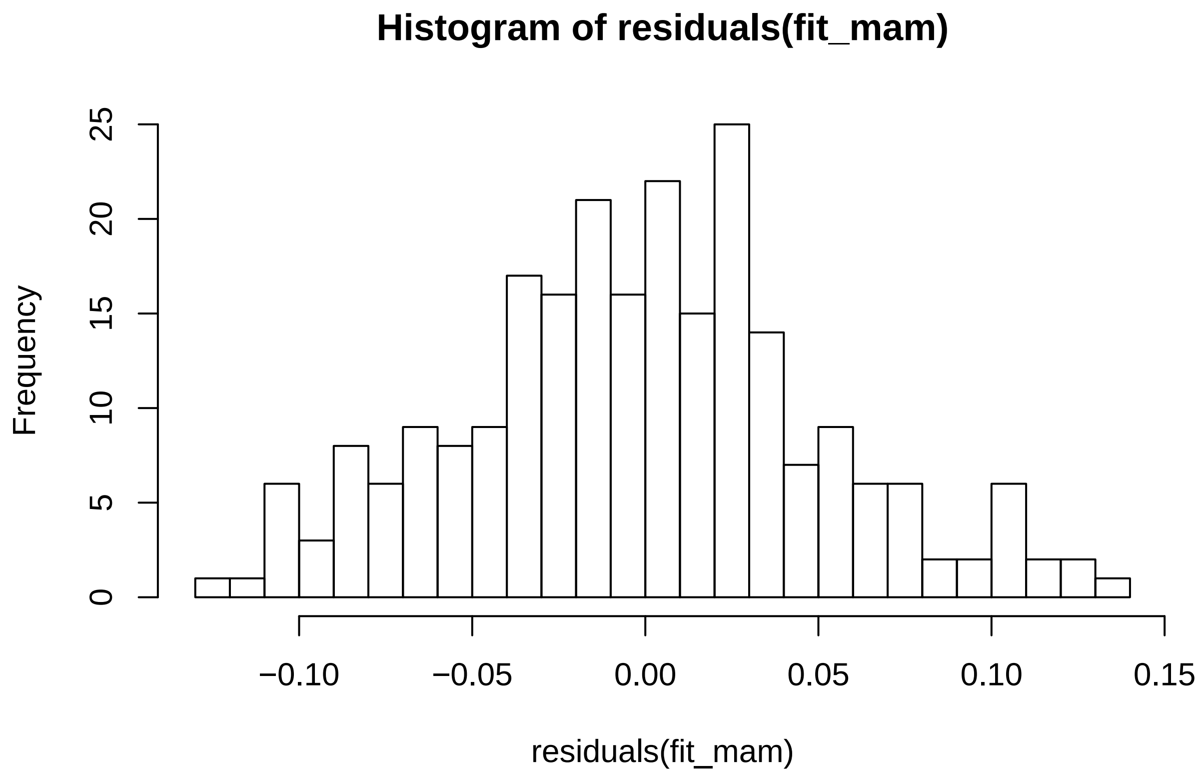
an ETS Model

```r
fit_mam = ets(visitors, model="ZZZ")
plot(forecast(fit_mam))
```

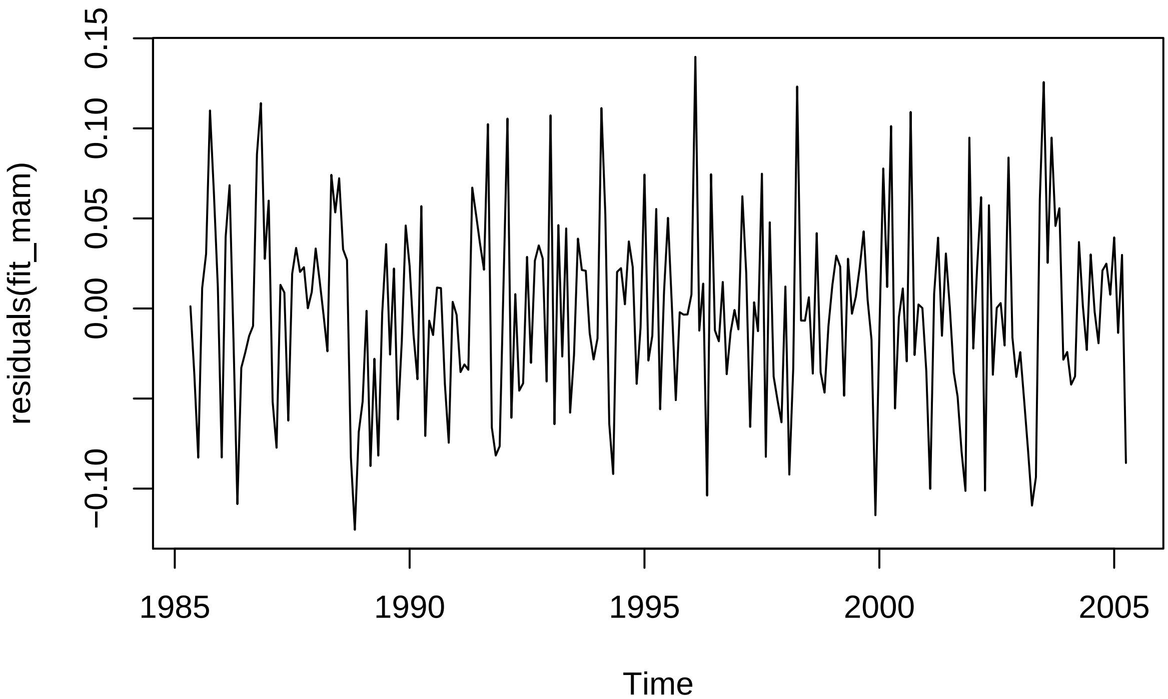
Forecasts from ETS(M,A,M)




```
hist(residuals(fit_mam), nclass=20)
```




```
plot(residuals(fit_mam))
```

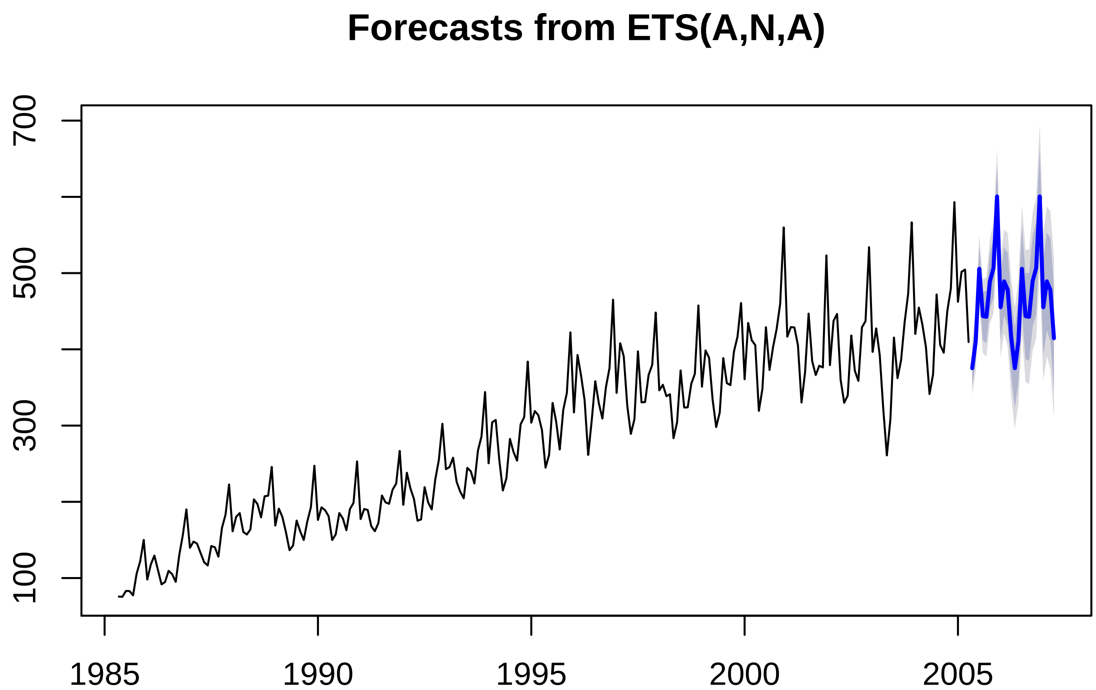



```r
accuracy(fit_mam)
```

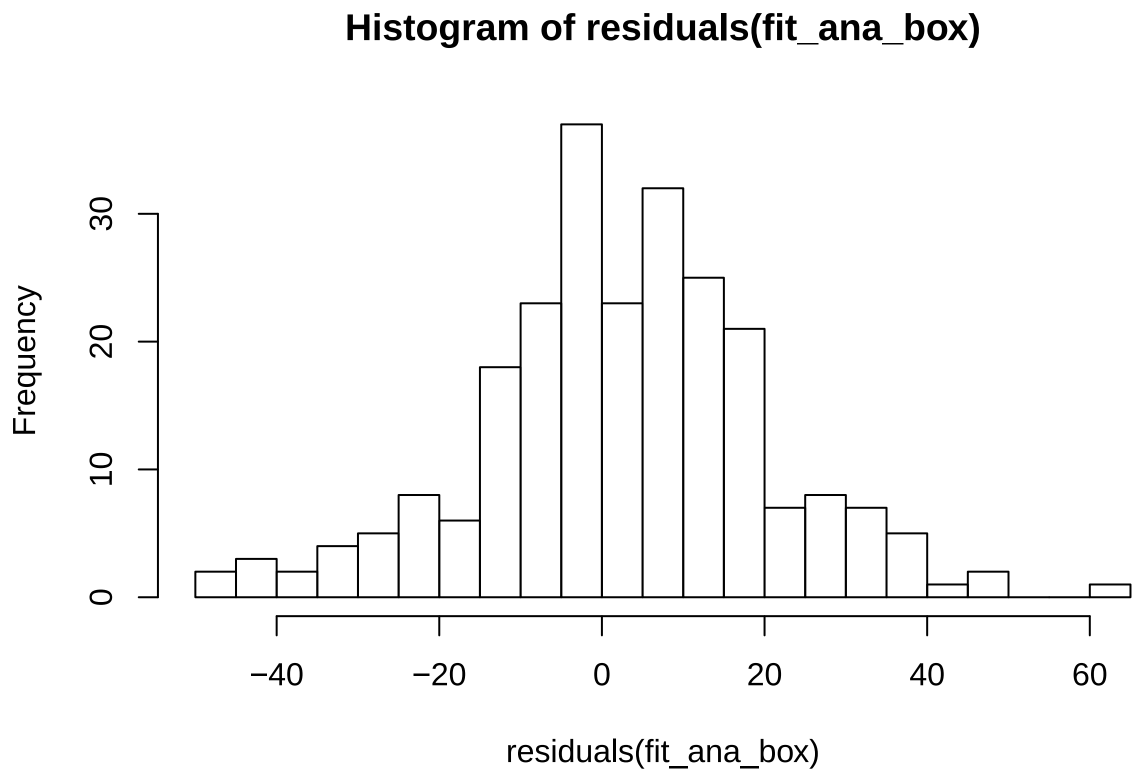
```
##                      ME   RMSE     MAE        MPE     MAPE      MASE
## Training set -0.9564743 15.847 11.5215 -0.4307078 4.075378 0.4254781
##                    ACF1
## Training set 0.02434609
```

Additive ETS model applied to a Box-Cox transformed Series

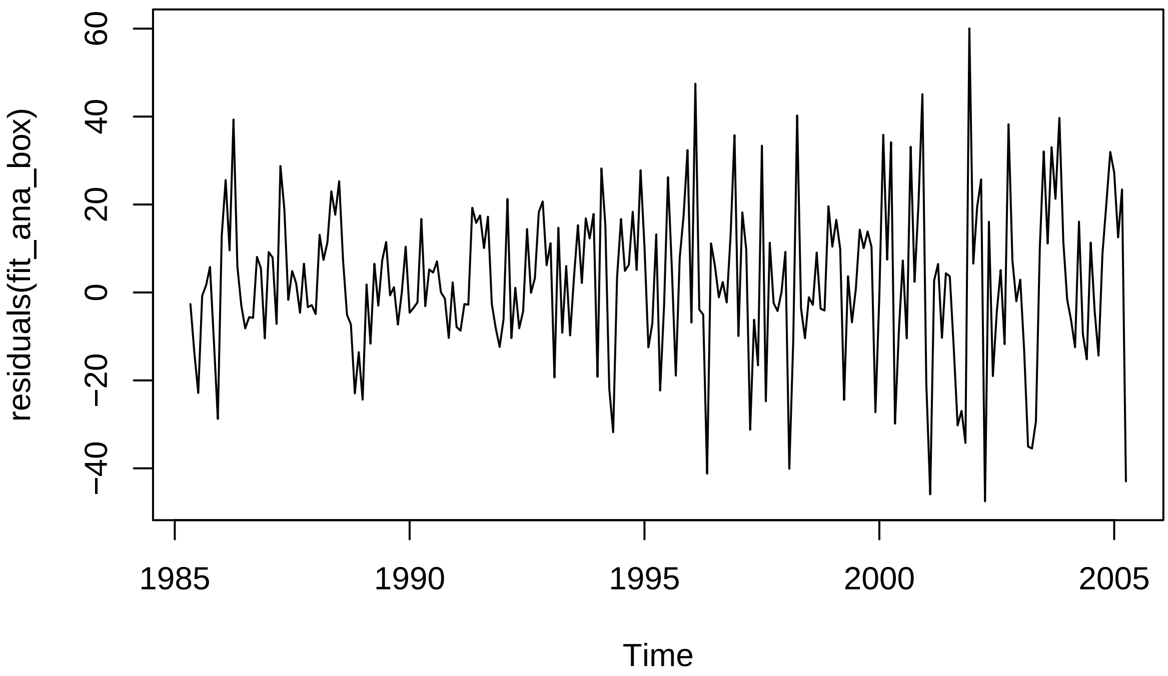
```r
fit_ana_box = ets(visitors, additive.only = TRUE, lambda = TRUE)
plot(forecast(fit_ana_box))
```




```
hist(residuals(fit_ana_box), nclass=20)
```




```r
plot(residuals(fit_ana_box))
```

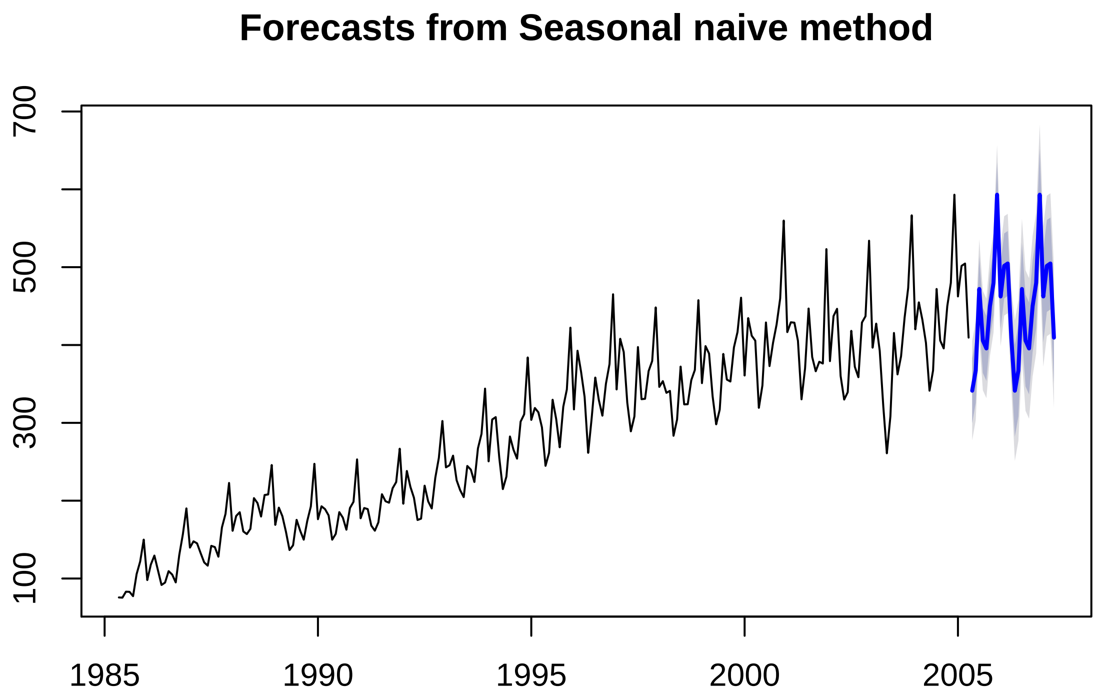



```r
accuracy(fit_ana_box)
```

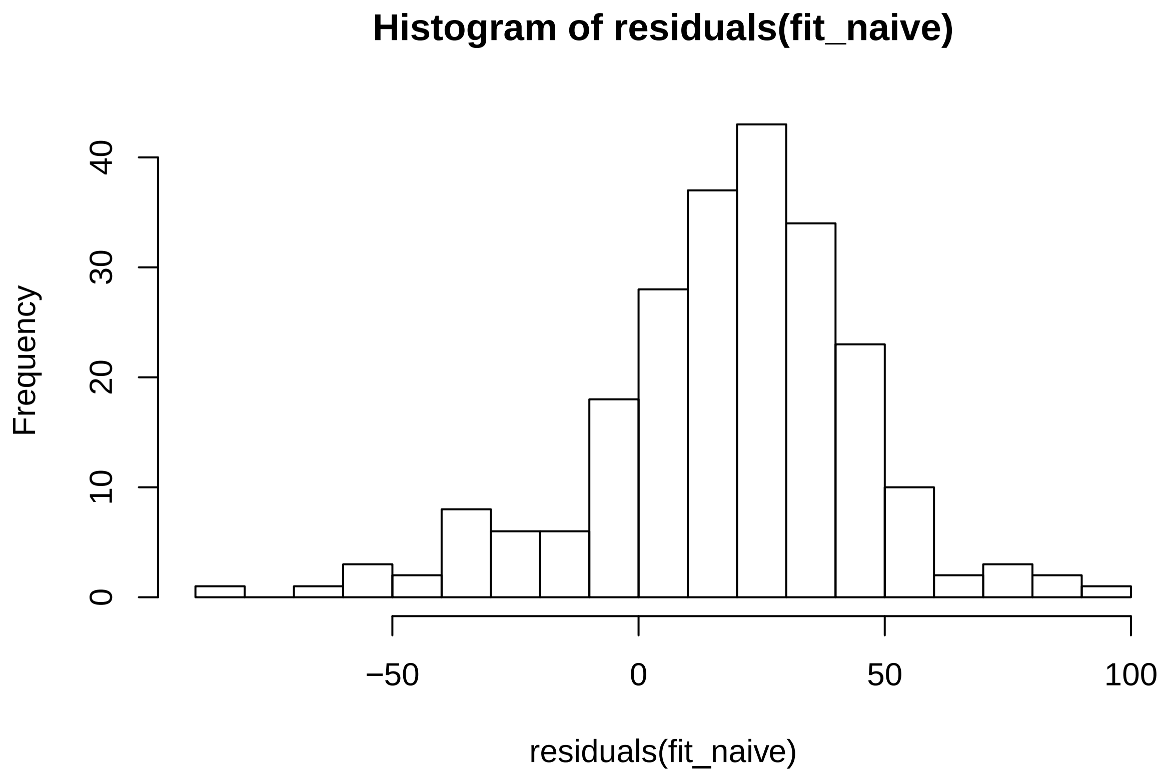
```
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 2.474204 17.75333 13.58897 0.7179028 5.123695 0.5018279
##                    ACF1
## Training set 0.08426191
```

Seasonal naive method applied to the Box-Cox transformed series

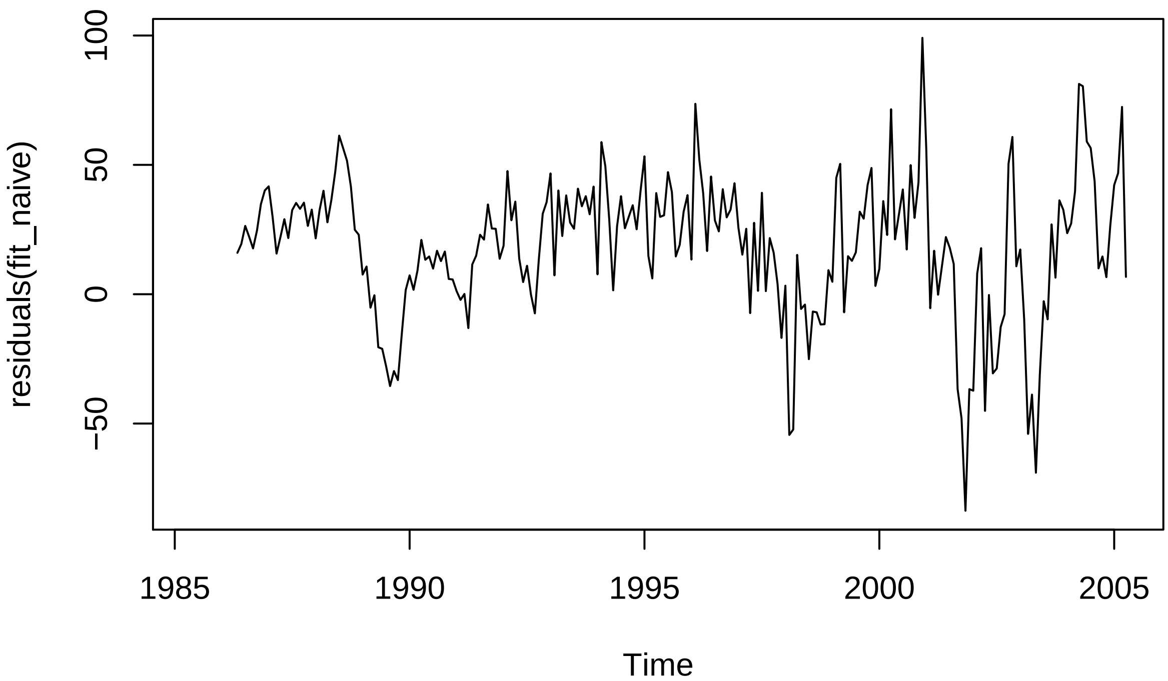
```r
fit_naive = snaive(visitors, lambda = TRUE)
plot(forecast(fit_naive))
```




```
hist(residuals(fit_naive), nclass=20)
```




```
plot(residuals(fit_naive))
```

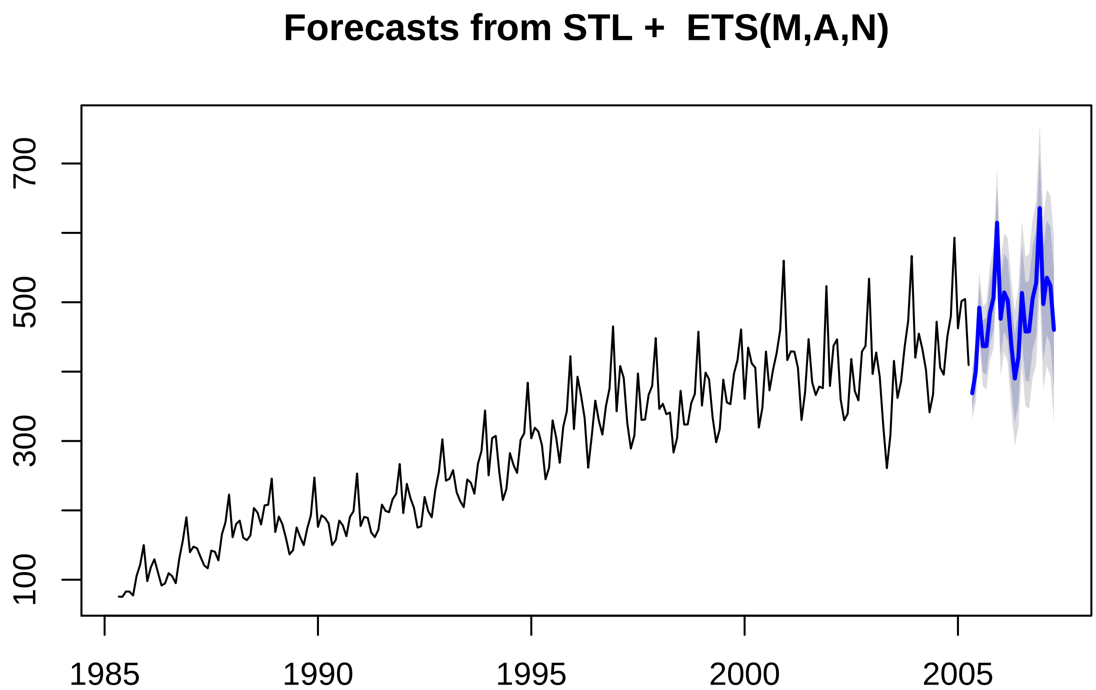



```r
accuracy(fit_naive)
```

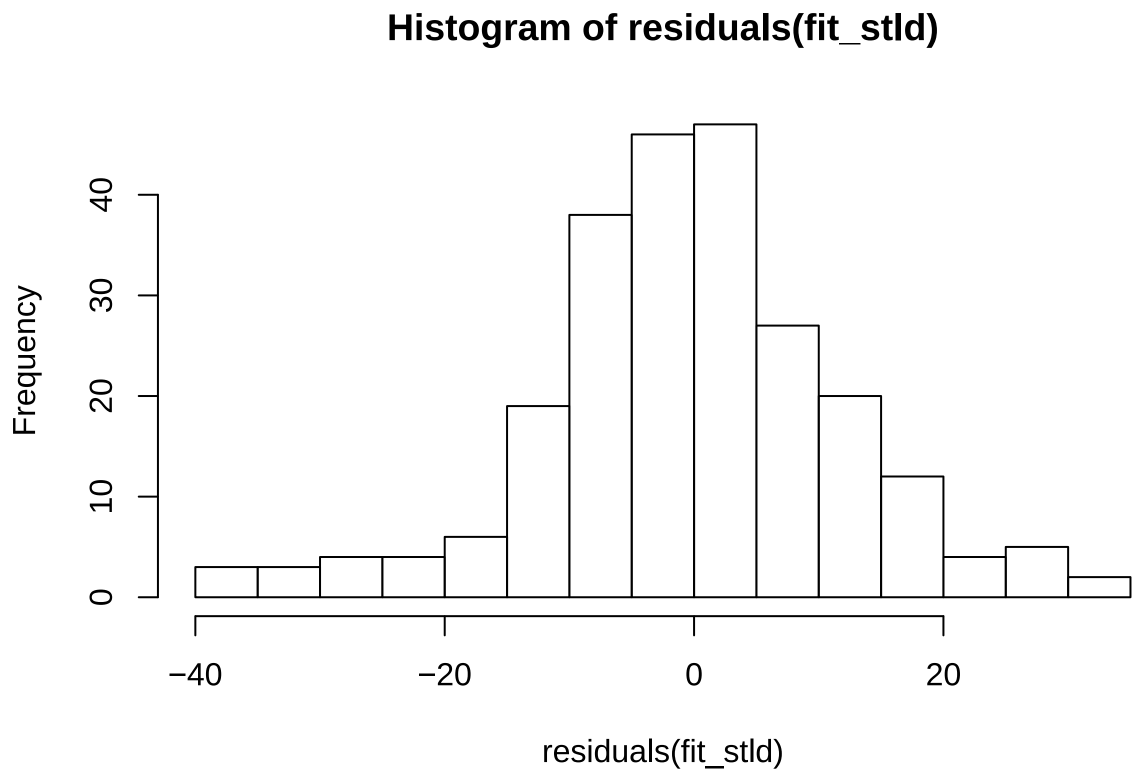
```
##                    ME     RMSE      MAE      MPE     MAPE MASE      ACF1
## Training set 18.22368 32.56941 27.07895 7.011798 10.12935    1 0.6600405
```

STL decomposition applied to the Box-Cox transformed data followed by an ETS model applied to the seasonally adjusted (transformed) data

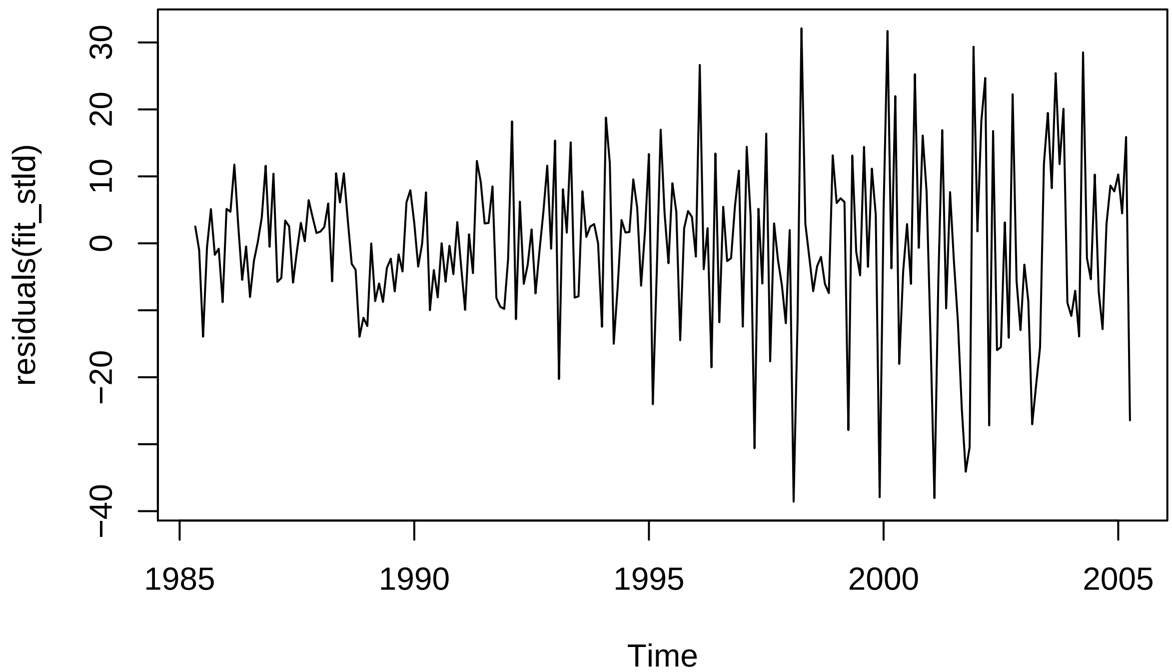
```r
fit_stld = stlf(visitors, method = "ets", lambda = TRUE)
plot(forecast(fit_stld))
```




```r
hist(residuals(fit_stld), nclass=20)
```




```
plot(residuals(fit_stld))
```




```r
accuracy(fit_stld)
```

```
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.3609864 12.17182 9.125237 -0.2344186 3.246417 0.3369864
##                     ACF1
## Training set -0.02552833
```

Part G

Which model from above do you prefer:

Looking through the forecasts, I'd rule out model 3 and 4 as the growth does not seem to match the upward trend. The residuals on the naive model in particular do not look normal or random. Although the RSME fit is best for the last model, the residual pattern does not look random (exhibits heteroschedascity). Therefore, I would choose the second model (ETS MAM) as it looks like the best balance of forecast quality, RMSE score, and no apparent issues in the residual diagnostics.