DATA624: Homework 5

Donald Butler

2023-10-08

Contents

Homework 5

```
library(fpp3)
library(tidyverse)
```

Exercise 8.1

Consider the number of pigs slaughtered in Victoria, available in the `aus_livestock` dataset.

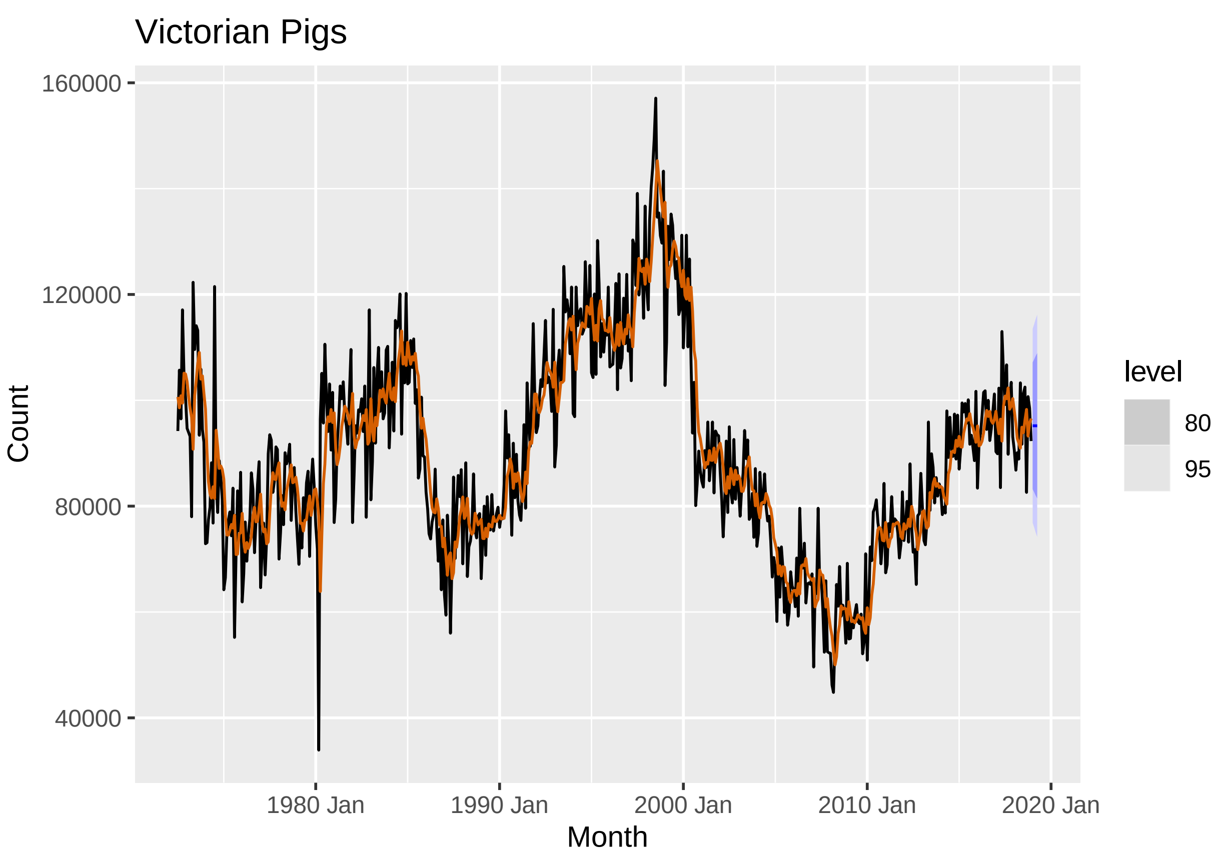
a. Use the `ETS()` function to estimate the equivalent model for simple exponential smoothing. Find the optimal values of α and ℓ_0, and generate forecasts for the next four months.

```
victoria_pigs <- aus_livestock |>
  filter(Animal == 'Pigs',
         State == 'Victoria')

fit <- victoria_pigs |>
  model(ETS(Count ~ error("A") + trend("N") + season("N")))

report(fit)
```

```
## Series: Count
## Model: ETS(A,N,N)
##   Smoothing parameters:
##     alpha = 0.3221247
##
##   Initial states:
##      l[0]
##  100646.6
##
##   sigma^2:  87480760
##
##      AIC     AICc      BIC
## 13737.10 13737.14 13750.07
```

The optimal values are: * $\alpha = 0.322$ * $\ell_0 = 100646$

```
fc <- fit |>
  forecast(h = 4)

fc |>
  autoplot(victoria_pigs) +
  geom_line(aes(y = .fitted),
            col = '#D55E00',
            data = augment(fit)) +
  labs(title = 'Victorian Pigs')
```



```r
fc |>
  select(Month, .mean, Count)
```

```
## # A fable: 4 x 3 [1M]
##      Month  .mean            Count
##      <mth>  <dbl>           <dist>
## 1 2019 Jan 95187. N(95187, 8.7e+07)
## 2 2019 Feb 95187. N(95187, 9.7e+07)
## 3 2019 Mar 95187. N(95187, 1.1e+08)
## 4 2019 Apr 95187. N(95187, 1.1e+08)
```

b. Compute a 95% prediction interval for the first forecast using $\hat{y} \pm 1.96s$ where s is the standard deviation of the residuals. Compare your interval with the interval produced by R.

```r
y_hat <- fc[1,] |> pull(.mean)
sd <- sd(residuals(fit)$.resid)
cat('[',
    format(y_hat-1.96*sd, digits = 7),
    ', ',
    format(y_hat+1.96*sd,digits = 7),
    ']95',sep = '')
```

```
## [76871.01, 113502.1]95
```

```r
fc[1,] |>
  hilo() |>
  select('95%')
```

```
## # A tsibble: 1 x 2 [1M]
##                    `95%`    Month
##                   <hilo>    <mth>
## 1 [76854.79, 113518.3]95 2019 Jan
```

Exercise 8.5

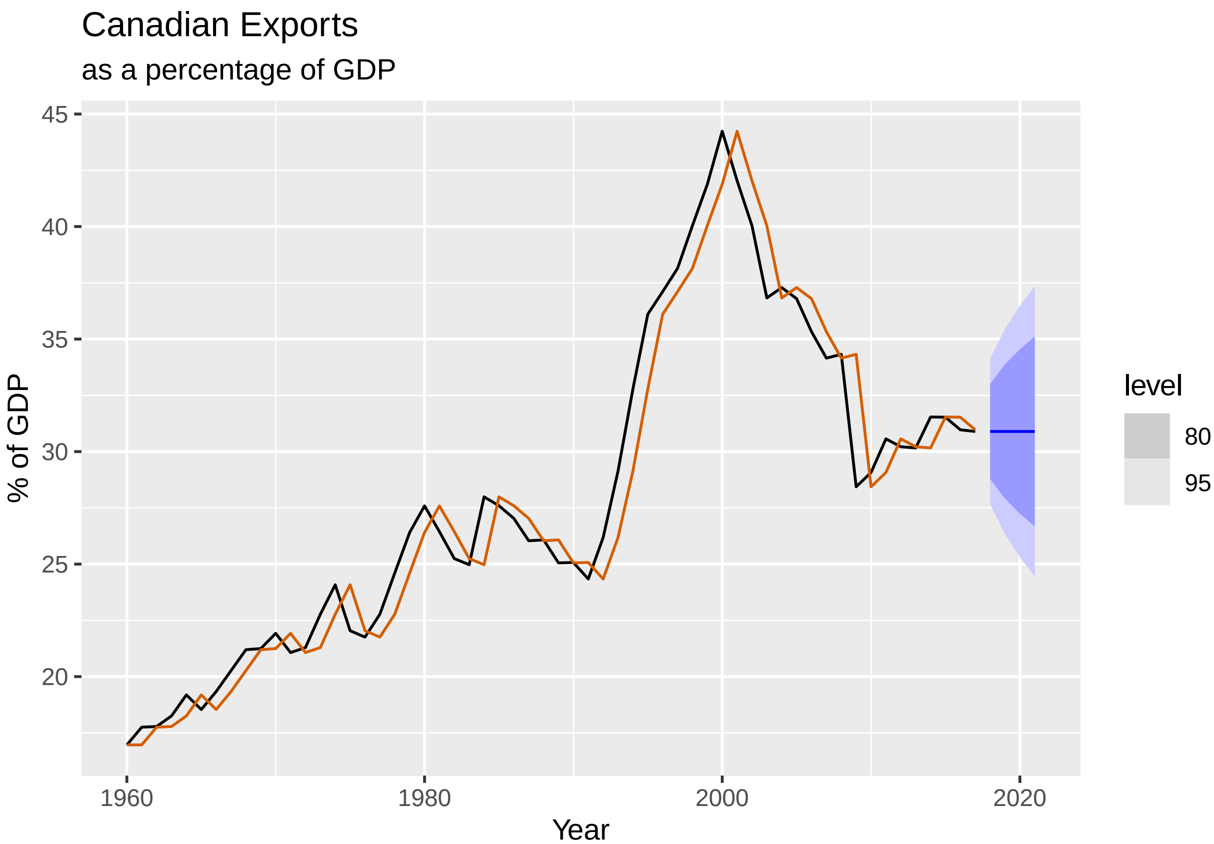
Dataset `global_economy` contains the annual Exports from many countries. Select one country to analyse.

a. Plot the Exports series and discuss the main features of the data.

```r
CanadianExports <- global_economy |>
  filter(Country == 'Canada')

CanadianExports |>
  autoplot(Exports) +
  labs(title = 'Canadian Exports',
       subtitle = 'as a percentage of GDP',
       y = '% of GDP')
```

b. Use an ETS(A,N,N) model to forecast the series, and plot the forecasts.

```
Canadian_fit <- CanadianExports |>
  model(ETS(Exports ~ error('A') + trend('N') + season('N')))

Canadian_fc <- Canadian_fit |>
  forecast(h = 4)

Canadian_fc |>
  autoplot(CanadianExports) +
  geom_line(aes(y = .fitted),
            col = '#D55E00',
            data = augment(Canadian_fit)) +
  labs(title = 'Canadian Exports',
       subtitle = 'as a percentage of GDP',
       y = '% of GDP')
```



c. Compute the RMSE values for the training data.

```
Canadian_fit |>
  accuracy() |>
  select(RMSE)
```

```
## # A tibble: 1 x 1
##    RMSE
##   <dbl>
## 1  1.62
```

d. Compare the results to those from an ETS(A,A,N) model. Remember that the trended model is using one more parameter than the simpler model. Discuss the merits of the two forecasting methods for this dataset.

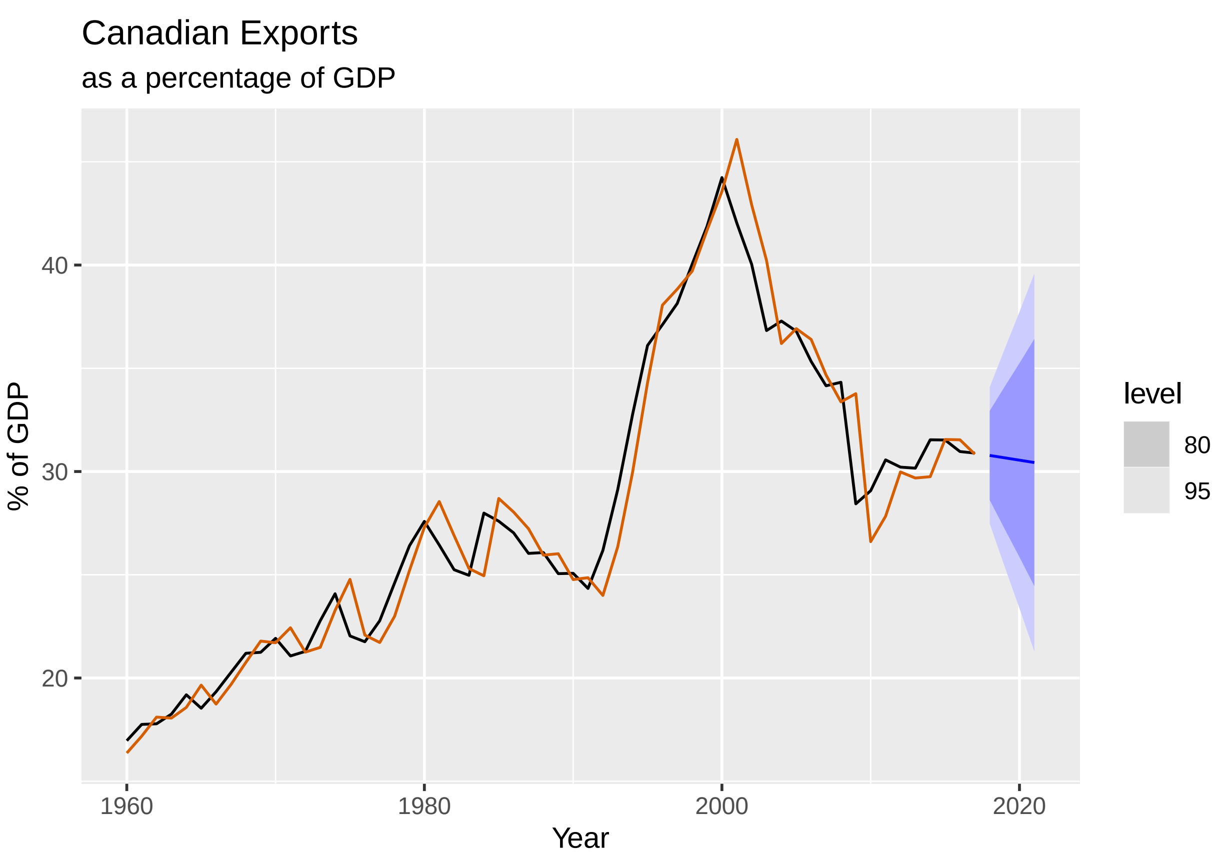
```
Canadian_fitCompare <- CanadianExports |>
  model(ANN = ETS(Exports ~ error('A') + trend('N') + season('N')),
        AAN = ETS(Exports ~ error('A') + trend('A') + season('N')))
accuracy(Canadian_fitCompare) |>
  select(.model, .type, RMSE)
```

```
## # A tibble: 2 x 3
##   .model .type     RMSE
##   <chr>  <chr>    <dbl>
## 1 ANN    Training  1.62
## 2 AAN    Training  1.63
```

Looking at the Canadian Exports data, there doesn't seem to be an overall trend that can be exploited for forecasting. Due to this, the AAN model isn't really any better than the ANN model.

e. Compare the forecasts from both methods. Which do you think is best?

```
Canadian_fit2 <- CanadianExports |>
  model(ETS(Exports ~ error('A') + trend('A') + season('N')))

Canadian_fc2 <- Canadian_fit2 |>
  forecast(h = 4)

Canadian_fc2 |>
  autoplot(CanadianExports) +
  geom_line(aes(y = .fitted),
            col = '#D55E00',
            data = augment(Canadian_fit2)) +
  labs(title = 'Canadian Exports',
       subtitle = 'as a percentage of GDP',
       y = '% of GDP')
```



The AAN model is producing a forecast that is trending lower based on the previous values, but since there is little indication that the trend in the data plays any role, the trending forecast isn't likely any better than the ANN model forecast.

f. Calculate a 95% prediction interval for the first forecast for each model, using RMSE values and assuming normal errors. Compare your intervals with those produces using R.

Confidence Interval for ETS(A,N,N)

```
y_hat <- Canadian_fc[1,] |> pull(.mean)
sd <- sd(residuals(Canadian_fit)$.resid)
cat('[',
    format(y_hat-1.96*sd, digits = 7),
    ', ',
    format(y_hat+1.96*sd,digits = 7),
    ']95',sep = '')
```

```
## [27.73107, 34.057]95
```

```
Canadian_fc[1,] |>
  hilo() |>
  select('95%')
```

```
## # A tsibble: 1 x 2 [1Y]
##                   `95%`  Year
##                  <hilo> <dbl>
## 1 [27.66725, 34.12082]95  2018
```

Confidence Interval for ETS(A,A,N)

```
y_hat <- Canadian_fc2[1,] |> pull(.mean)
sd <- sd(residuals(Canadian_fit2)$.resid)
cat('[',
    format(y_hat-1.96*sd, digits = 7),
    ', ',
    format(y_hat+1.96*sd,digits = 7),
    ']95',sep = '')
```

```
## [27.56593, 33.99359]95
```

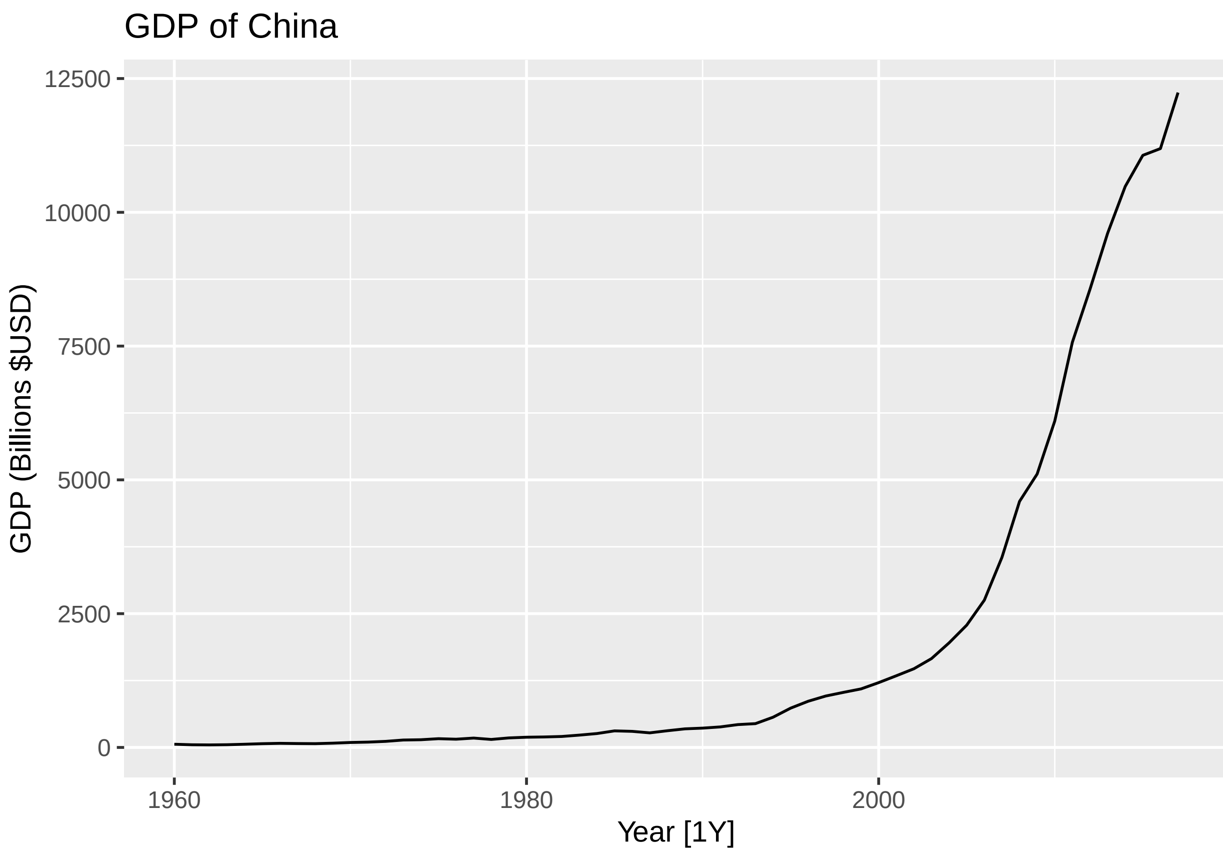
```
Canadian_fc[1,] |>
  hilo() |>
  select('95%')
```

```
## # A tsibble: 1 x 2 [1Y]
##                   `95%`  Year
##                  <hilo> <dbl>
## 1 [27.66725, 34.12082]95  2018
```

Exercise 8.6

Forecast the Chinese GDP from the `global_economy` dataset using an ETS model. Experiment with the various options in the `ETS()` function to see how much the forecasts change with damped trend, or with a Box-Cox transformation. Try to develop an intuition of what each is doing to the forecasts.

```r
ChinaGDP <- global_economy |>
  filter(Country == 'China') |>
  mutate(GDP = GDP / 10^9)

ChinaGDP |>
  autoplot(GDP) +
  labs(title = 'GDP of China',
       y = 'GDP (Billions $USD)')
```



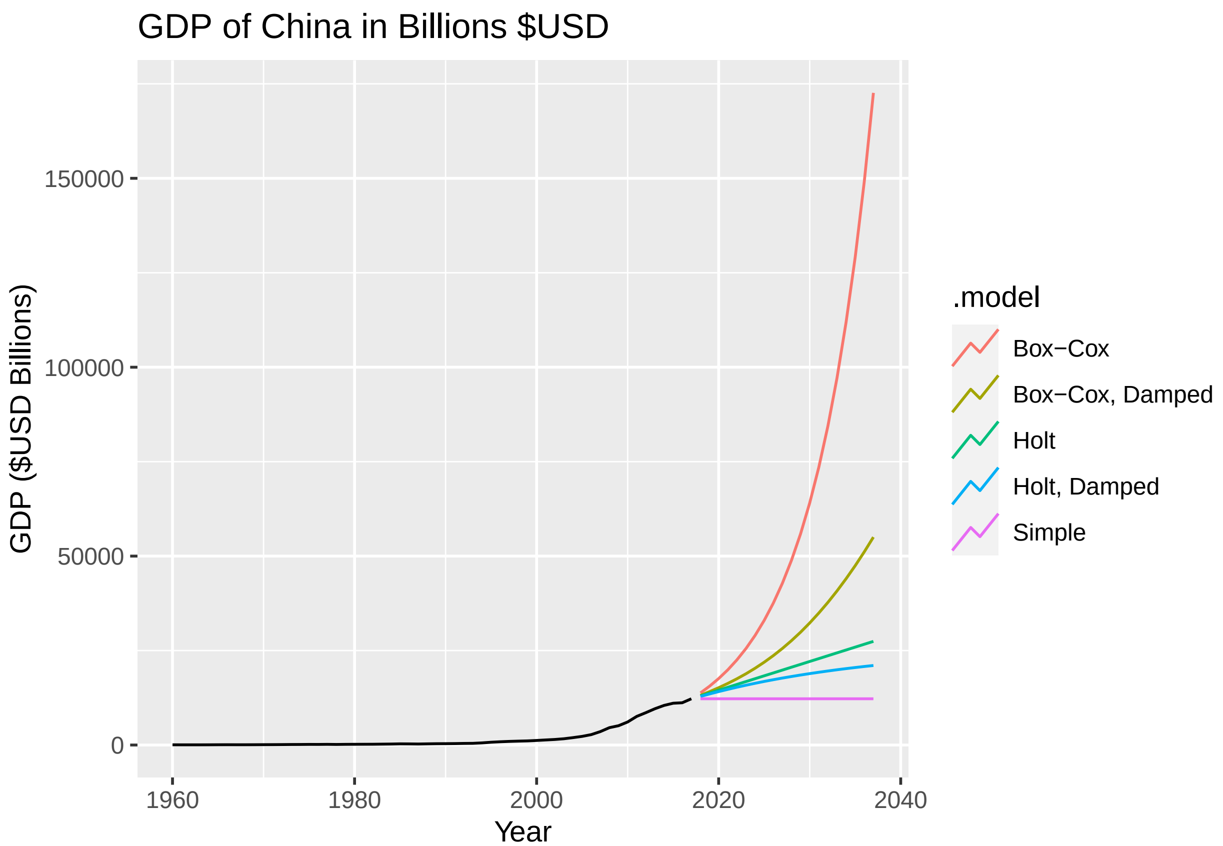

```r
lambda <- ChinaGDP |>
  features(GDP, features = guerrero) |>
  pull(lambda_guerrero)
lambda
```

```
## [1] -0.03446284
```

With a negative value so near zero, will set $\lambda = 0$ and let the Box-Cox transformation use the natural log.

```r
lambda = 0

ChinaGDP_fit <- ChinaGDP |>
  model(Simple = ETS(GDP ~ error('A') + trend('N') + season('N')),
        Holt = ETS(GDP ~ error('A') + trend('A') + season('N')),
        `Holt, Damped` =  ETS(GDP ~ error('A') + trend('Ad', phi = .95) + season('N')),
        `Box-Cox` = ETS(box_cox(GDP, lambda) ~ error('A') + trend('A') + season('N')),
```

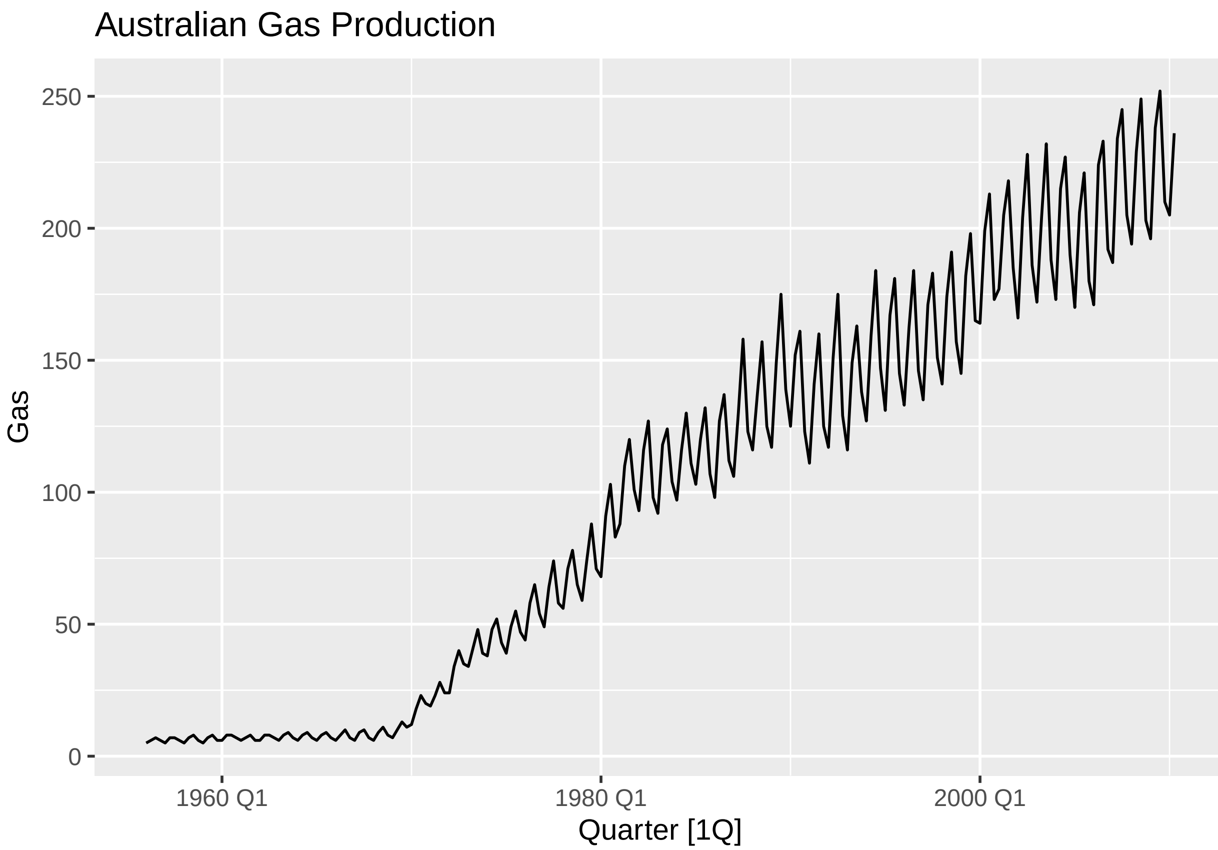
```
        `Box-Cox, Damped` = ETS(box_cox(GDP, lambda) ~ error('A') + trend('Ad', phi = .95) + season('N'
        )

ChinaGDP_fc <- ChinaGDP_fit |>
  forecast(h = 20)

ChinaGDP_fc |>
  autoplot(ChinaGDP, level = NULL) +
  labs(title = 'GDP of China in Billions $USD',
       y = 'GDP ($USD Billions)')
```



Exercise 8.7

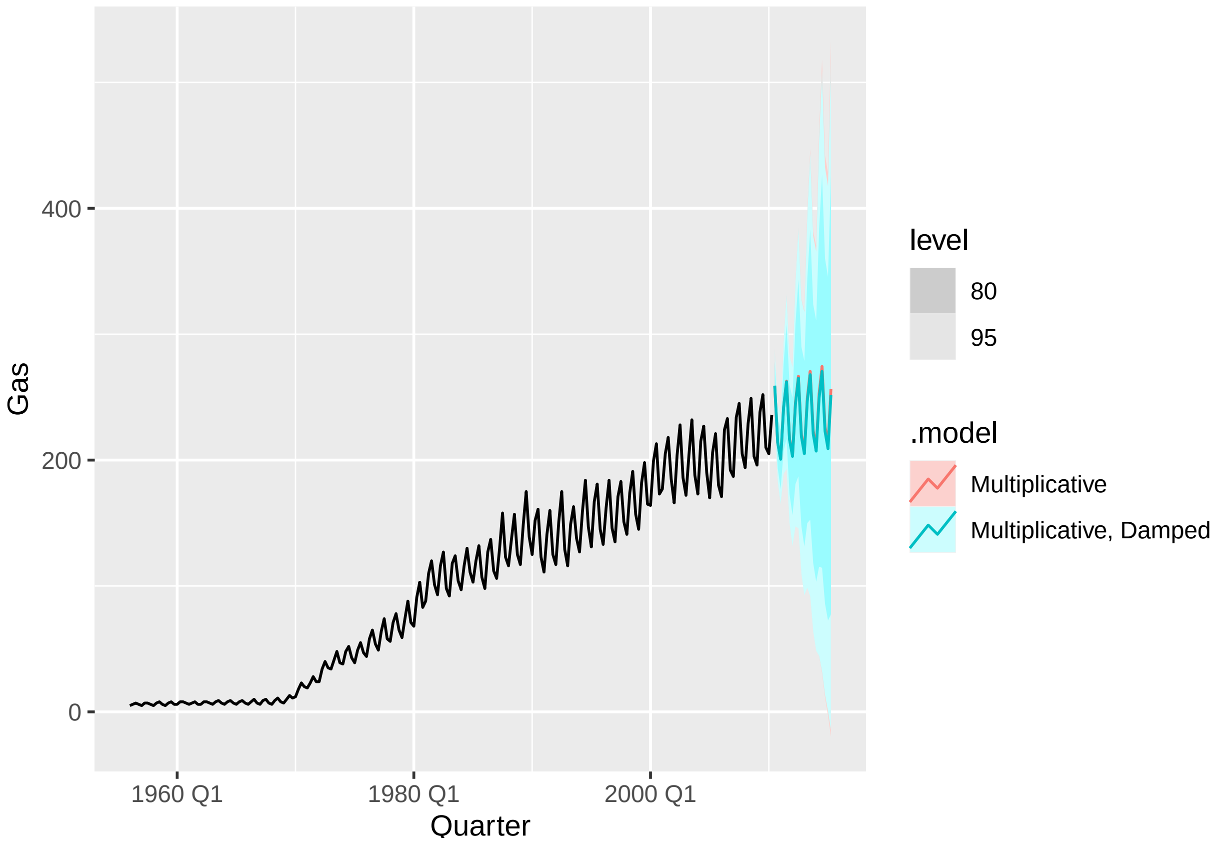
Find an ETS model for the Gas data from `aus_production` and forecast the next few years. Why is multiplicative seasonality necessary here? Experiment with making the trend damped. Does it improve the forecasts?

```
aus_production |>
  autoplot(Gas) +
  labs(title = 'Australian Gas Production')
```



Multiplicative seasonality is needed here because the seasonal variation is increasing as the time increases.

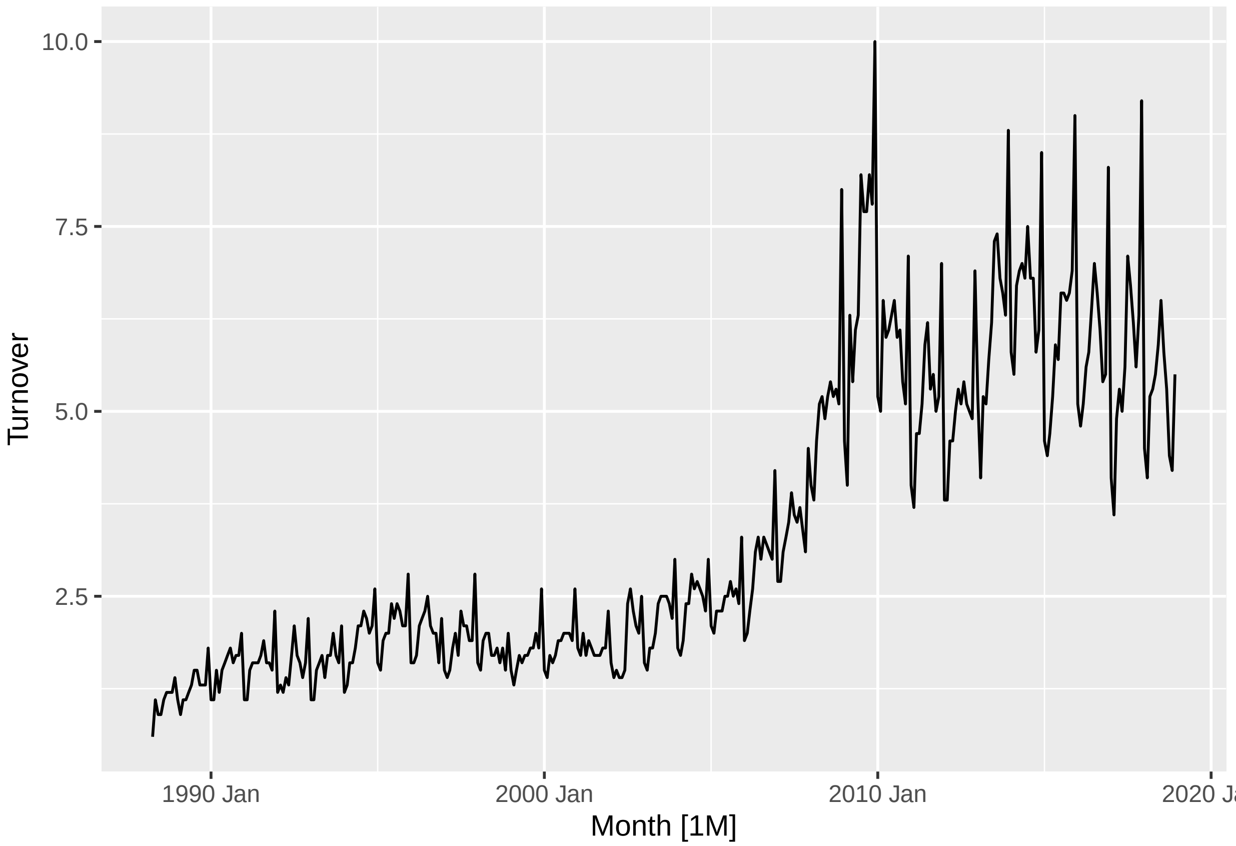
```
gas_fit <- aus_production |>
  model(Multiplicative = ETS(Gas ~ error('M') + trend('A') + season('M')),
        `Multiplicative, Damped` = ETS(Gas ~ error('M') + trend('Ad') + season('M')))

gas_fc <- gas_fit |>
  forecast(h = 20)

gas_fc |>
  autoplot(aus_production)
```



Exercise 8.8

Recall your retail time series data from Exercise 7 in Section 2.10.

```
set.seed(31415)
myseries <- aus_retail |>
  filter(`Series ID` == sample(aus_retail$`Series ID`, 1))

myseries |>
  autoplot(Turnover)
```

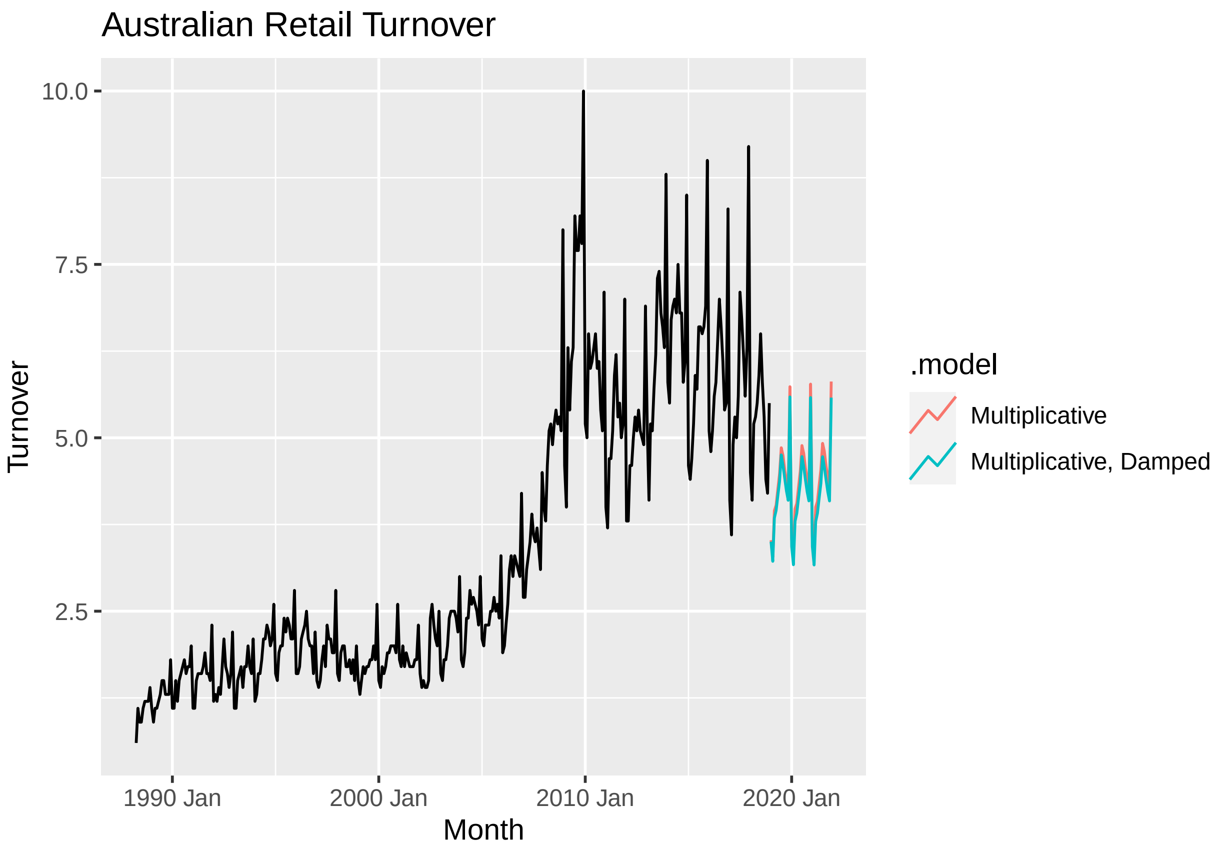


a. Why is multiplicative seasonality necessary for this series?

Multiplicative seasonality is ncessary for this series because the variation is increasing over time.

b. Apply Holt-Winters' multiplicative method to the data. Experiment with making the trend damped.

```
myseries_fit <- myseries |>
  model(Multiplicative = ETS(Turnover ~ error('M') + trend('A') + season('M')),
        `Multiplicative, Damped` = ETS(Turnover ~ error('M') + trend('Ad') + season('M')))

myseries_fc <- myseries_fit |>
  forecast(h = 36)

myseries_fc |>
  autoplot(myseries, level = NULL) +
  labs(title = 'Australian Retail Turnover')
```



c. Compare the RMSE of the one-step forecasts from the two methods. Which do you prefer?

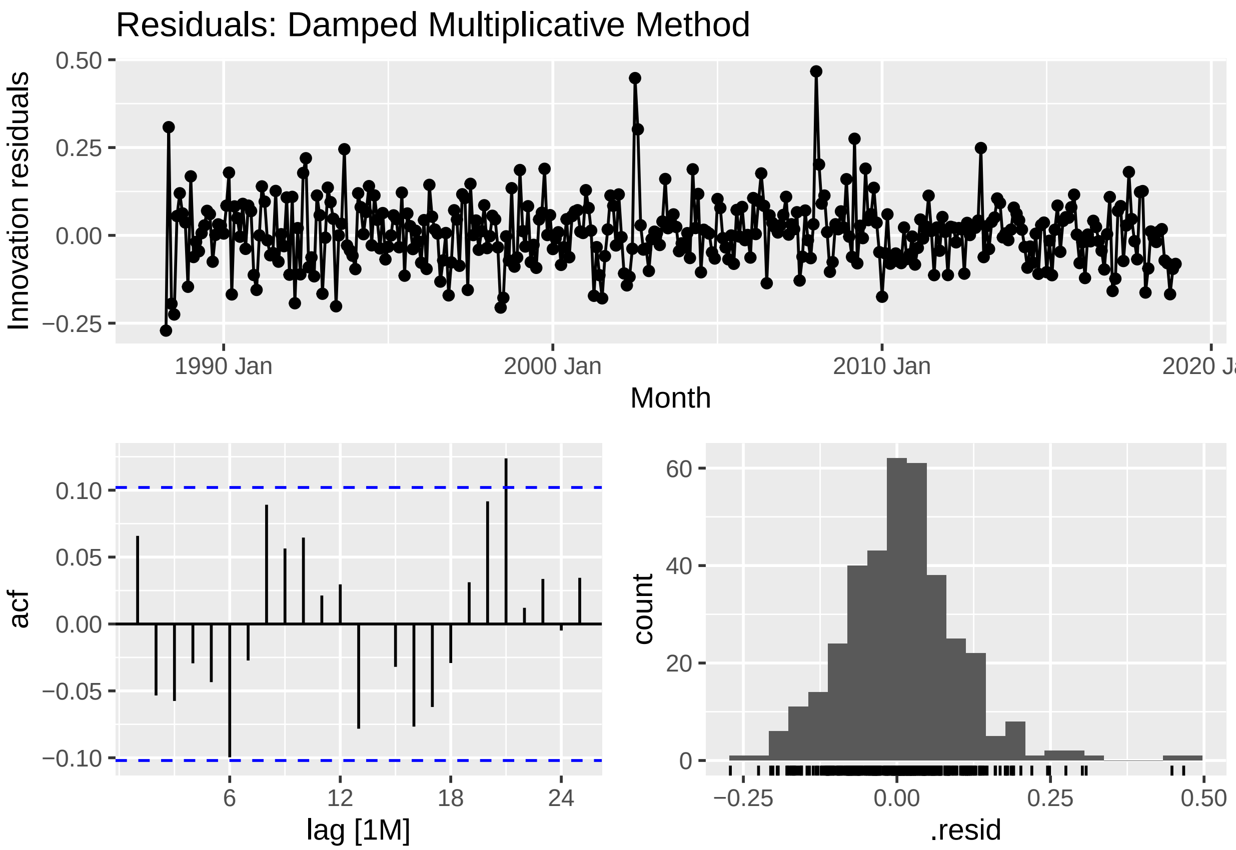
```
accuracy(myseries_fit) |>
  select(.model, RMSE)
```

```
## # A tibble: 2 x 2
##   .model                  RMSE
##   <chr>                  <dbl>
## 1 Multiplicative         0.319
## 2 Multiplicative, Damped 0.317
```

The RMSE for the damped model is slightly lower than the multiplicative model, but doesn't appear to be significant.

d. Check that the residuals from the best method look like white noise.

```
myseries_fit |>
  select(model = `Multiplicative, Damped`) |>
  gg_tsresiduals() +
  labs(title = 'Residuals: Damped Multiplicative Method')
```



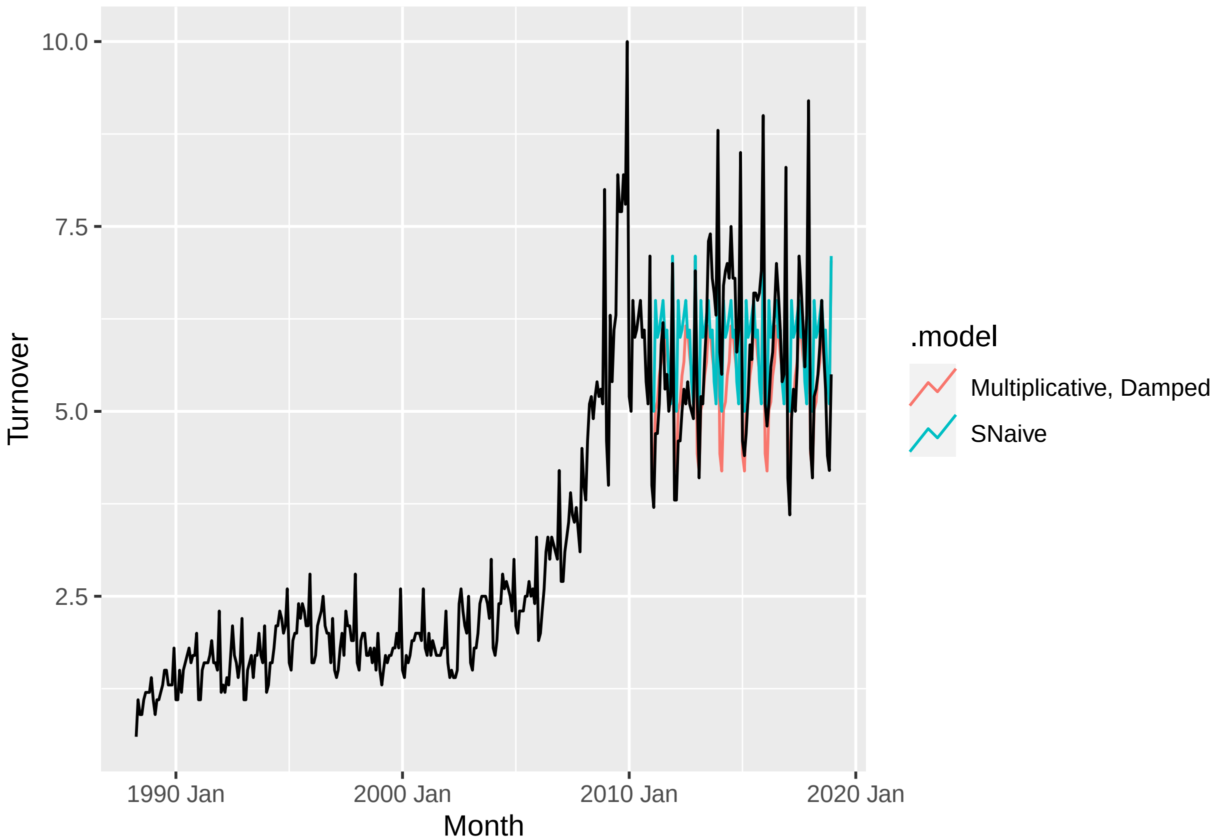

```
myseries_fit |>
  select(model = `Multiplicative, Damped`) |>
  augment() |>
  features(.innov, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 model     32.4     0.118
```

The p-value of .11 is not less than .05, and we can conclude that the residuals resemble white noise.

e. Now find the test set RMSE, while training the model to the end of 2010. Can you beat the seasonal naïve approach from Exercise 7 in Section 5.11?

```
myseries_train <- myseries |> filter(year(Month) < 2011)

train_fit <- myseries_train |>
  model(`Multiplicative, Damped` = ETS(Turnover ~ error('M') + trend('Ad') + season('M')),
        SNaive = SNAIVE(Turnover))

train_fc <- train_fit |>
  forecast(new_data = anti_join(myseries, myseries_train))

train_fc |>
  autoplot(myseries, level = NULL)
```



Exercise 8.9

For the same retail data, try an STL decomposition applied to the Box-Cox transformed series, followed by ETS on the seasonally adjusted data. How does that compare with your best previous forecasts on the test set?

```
myseries_lambda <- myseries |>
  features(Turnover, features = guerrero) |>
  pull(lambda_guerrero)

myseries_lambda
```

```
## [1] -0.0264685
```

With a small negative lambda value calculated using the guerrero feature, I'd select $\lambda = 0$, and apply a log transformation.

```
myseries_lambda <- 0

myseries_bc <- myseries |>
  mutate(Turnover_bc = box_cox(Turnover, myseries_lambda))

myseries_bc_fit <- myseries_bc |>
  model(`Box-Cox ETS` = ETS(Turnover_bc),
        `Box-Cox STL` = STL(Turnover_bc ~ season(window = 'periodic')))
```

```r
accuracy(myseries_bc_fit) |>
  select(.model, RMSE) |>
  rbind(accuracy(myseries_fit) |>
    select(.model, RMSE))
```

```
## # A tibble: 4 x 2
##   .model                   RMSE
##   <chr>                   <dbl>
## 1 Box-Cox ETS            0.0939
## 2 Box-Cox STL            0.0781
## 3 Multiplicative         0.319
## 4 Multiplicative, Damped 0.317
```