Stats 326 A3

Richard Choi

2022-05-15

```
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.1.3
```

```
## -- Attaching packages -------------------------------------------- fpp3 0.4.0 --
```

```
## v tibble      3.1.6     v tsibble     1.1.1
## v dplyr       1.0.8     v tsibbledata 0.4.0
## v tidyr       1.2.0     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tsibble' was built under R version 4.1.3
```

```
## Warning: package 'tsibbledata' was built under R version 4.1.3
```

```
## Warning: package 'feasts' was built under R version 4.1.3
```

```
## Warning: package 'fabletools' was built under R version 4.1.3
```

```
## Warning: package 'fable' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v readr   2.1.2     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x tsibble::intersect()     masks lubridate::intersect(), base::intersect()
## x tsibble::interval()      masks lubridate::interval()
## x dplyr::lag()             masks stats::lag()
## x tsibble::setdiff()       masks lubridate::setdiff(), base::setdiff()
## x tsibble::union()         masks lubridate::union(), base::union()
```

```
library(lubridate)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

Question 1

Plot the data and comment on what you can observe

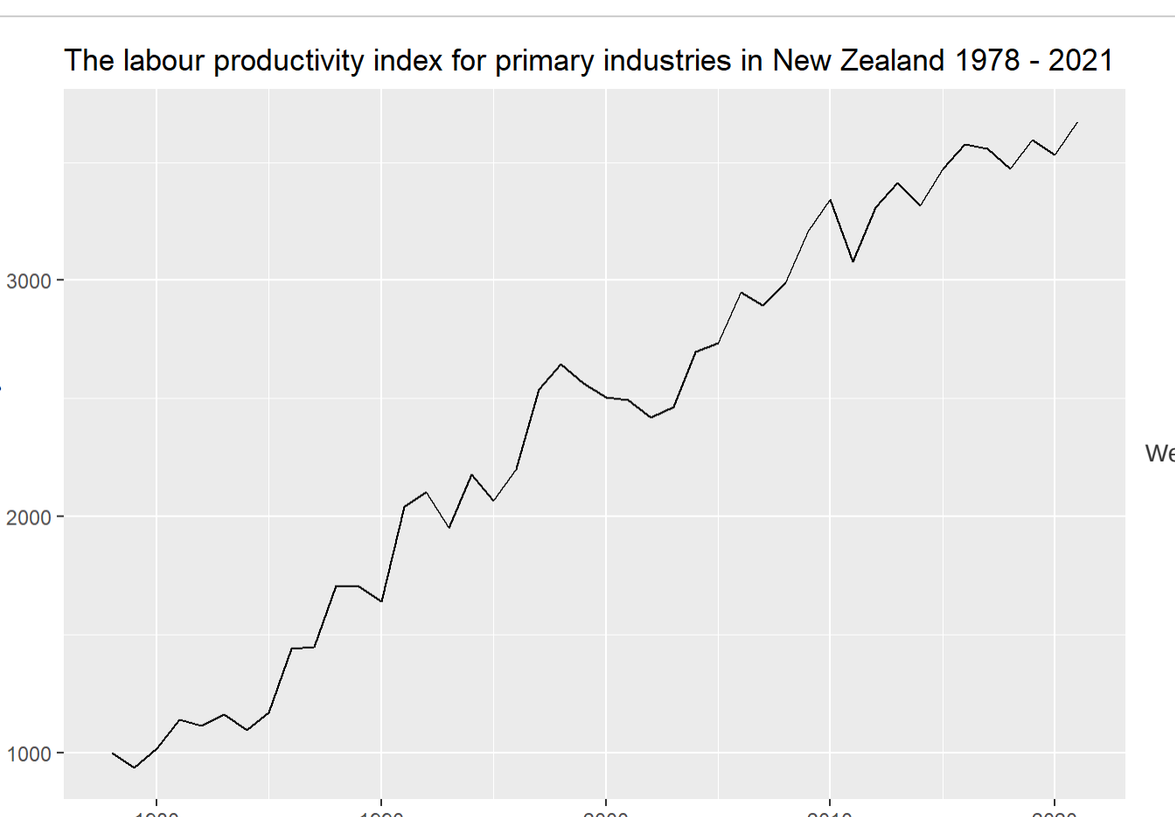
```
productivity.df = read_csv("productivity.csv")
```

```
## Rows: 44 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (2): Year, Productivity
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
productivity.tsibble = productivity.df %>%
  as_tsibble(index='Year')

productivity.tsibble %>% autoplot() + xlab("Year") +
  ggtitle("The labour productivity index for primary industries in New Zealand 1978 - 2021")
```

```
## Plot variable not specified, automatically selected `.vars = Productivity`
```



We can see that there is a strong positive linear trend in the plot. This means that the New Zealand productivity in primary industries is being more efficient as the time progresses.

Question 2

Create a training set that contains data from the years 1978–2016.

```
productivityTraining = productivity.df %>%
  filter(Year < 2017) %>%
  as_tsibble(index='Year')
```

Fit Holt's linear trend model and Holt's damped linear trend model to the training data.

```
HoltFit = productivityTraining %>%
  model(Holt = ETS(Productivity ~ error("A") + trend("A") + season("N")))

DampedFit = productivityTraining %>%
  model(Damped = ETS(Productivity ~ error("A") + trend("Ad") + season("N")))

report(HoltFit)
```

```
## Series: Productivity
## Model: ETS(A,A,N)
##   Smoothing parameters:
##     alpha = 0.4889942
##     beta  = 0.0001000043
##
##   Initial states:
##      l[0]     b[0]
##  880.7263 68.67191
##
##   sigma^2:  19414.32
##
##      AIC     AICc      BIC
## 533.7355 535.5536 542.0533
```

```
report(DampedFit)
```

```
## Series: Productivity
## Model: ETS(A,Ad,N)
##   Smoothing parameters:
##     alpha = 0.5939802
##     beta  = 0.0001001014
##     phi   = 0.98
##
##   Initial states:
##     l[0]     b[0]
##  879.452 86.63342
##
##   sigma^2:  20988.5
##
##      AIC     AICc      BIC
## 537.6455 540.2705 547.6269
```

Interpret the estimates for the model parameters (α,β∗,ϕ) of Holt's damped linear trend model.

α represents smoothing parameter for the level. α = 0.59 shows that the level reacts moderately to each new observation.

Beta* = 0.0001001014/0.5939802 = 0.0001685265 shows less weight to the changing level and more weight to the previous trend. This means that the slope of the model change over time by small degree.

phi represents the dampness in the Holt's damped linear trend model. Phi value is close to 1 which means that the forecast trend is not much damped and the forecast is trended.

Compare AICc. Which model has a better fit to the training data?

Comparing AICc, Holt model seems better than Holt's dampged linear trend model at fitting to the training data.

Question 3

Based on the models fitted in part 2, do the following: [9 Marks]

Forecast the next 5 years into the future.

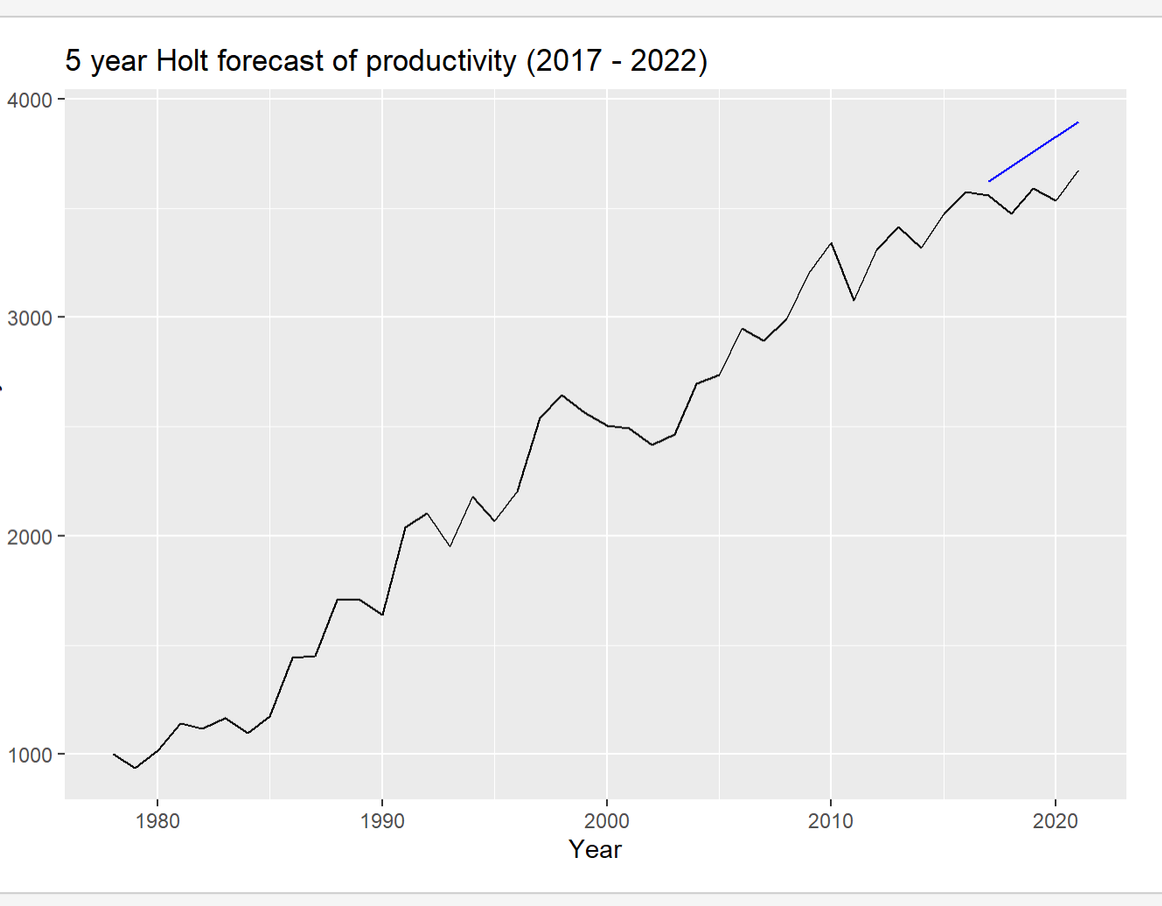
```
HoltFc = HoltFit %>%
  forecast(h=5)

DampedFc = DampedFit %>%
  forecast(h=5)
```

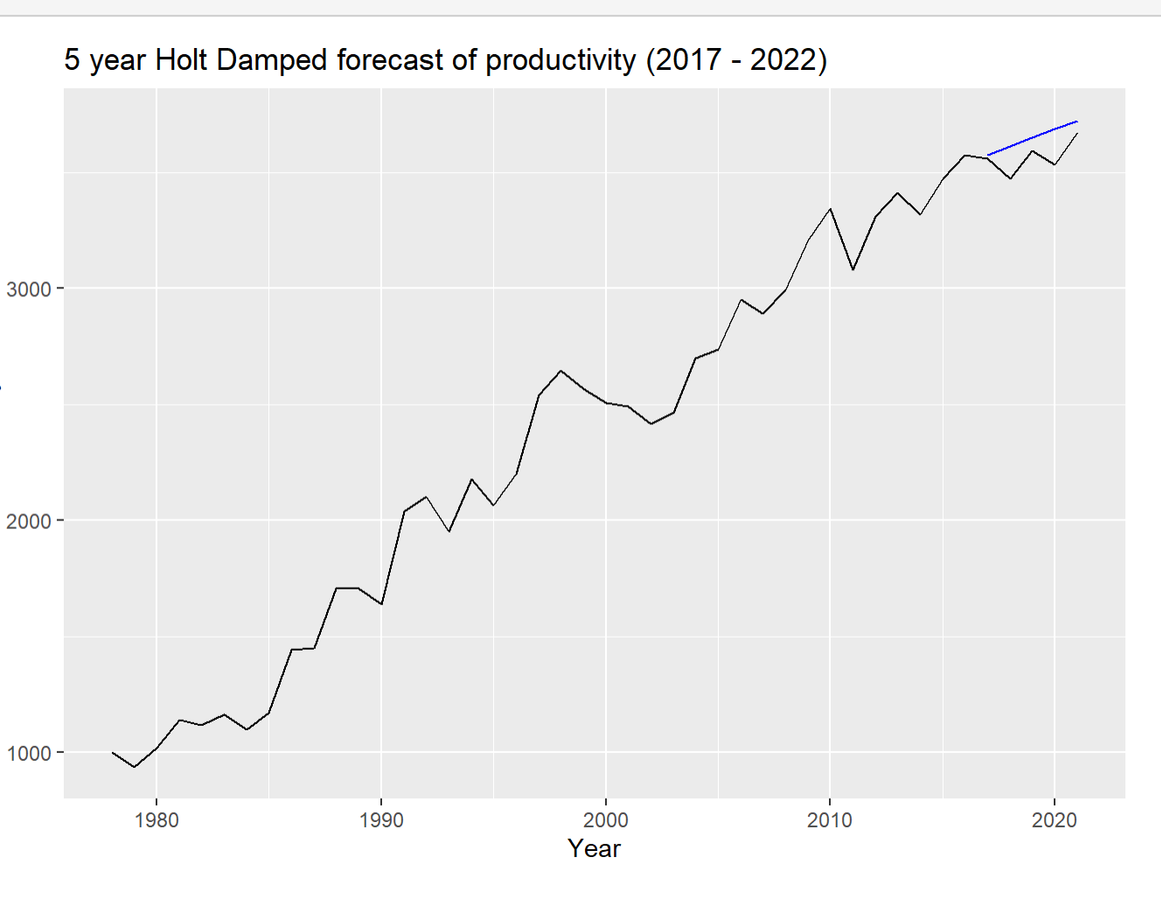
Create a plot where you overlay the point forecasts on the original data.

```
productivity.tsibble = productivity.df %>%
  as_tsibble(index="Year")

HoltFc %>% autoplot(productivity.tsibble, level=NULL) + ggtitle("5 year Holt forecast of productivity (2017 - 202
2)")
```




```
DampedFc %>% autoplot(productivity.tsibble, level=NULL) + ggtitle("5 year Holt Damped forecast of productivity (2
017 - 2022)")
```



Compute appropriate measures of forecast accuracy. Comment on which model provides better forecasts.

```
accuracy(HoltFc, productivity.tsibble)
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Holt   Test  -194.  208.  194. -5.43  5.43  1.52  1.31 -0.115
```

```
accuracy(DampedFc, productivity.tsibble)
```

```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Damped Test  -83.8  99.4  83.8 -2.37  2.37 0.659 0.628 -0.657
```

The root mean squared error (RMSE) is 99.45 for Damped model compared to Holt's model RMSE of 208.13. This means that the Holt's damped model is a much more accurate model for forecast.

Report 95% prediction interval for the year 2022 for the model with the better forecasts. Interpret this in plain English.

```
DampedFit %>%
  forecast(h=6) %>%
  hilo(level=95) %>%
  filter(Year == '2022')
```

```
## # A tsibble: 1 x 5 [1Y]
## # Key:       .model [1]
##   .model  Year   Productivity .mean                  `95%`
##   <chr>  <dbl>         <dist> <dbl>                 <hilo>
## 1 Damped  2022 N(3758, 58049) 3758. [3285.731, 4230.176]95
```

We estimate that the year 2022 on average will have the productivity index between 3285.73 and 4230.18.

Using the data you have available, discuss how could you reduce the forecast uncertainty?

Using the data available, we could perform cross validation to reduce the forecast uncertainty. Cross validation such as evaluation on a rolling forecasting origin can help to choose a good forecasting model which can reduce the forecast uncertainty.

Problem 2

Manually determine an appropriate non-seasonal ARIMA model for the productivity training data (from Problem 1), by doing the following: [9 Marks]

Conduct a KPSS unit root test on the training data. Keep differencing the data until it is stationary. What is the order of differencing d?

```
productivityTraining %>% features(Productivity, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      1.06        0.01
```

```
productivityTraining %>%
mutate(diff_productivity = difference(Productivity)) %>%
features(diff_productivity, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0520         0.1
```

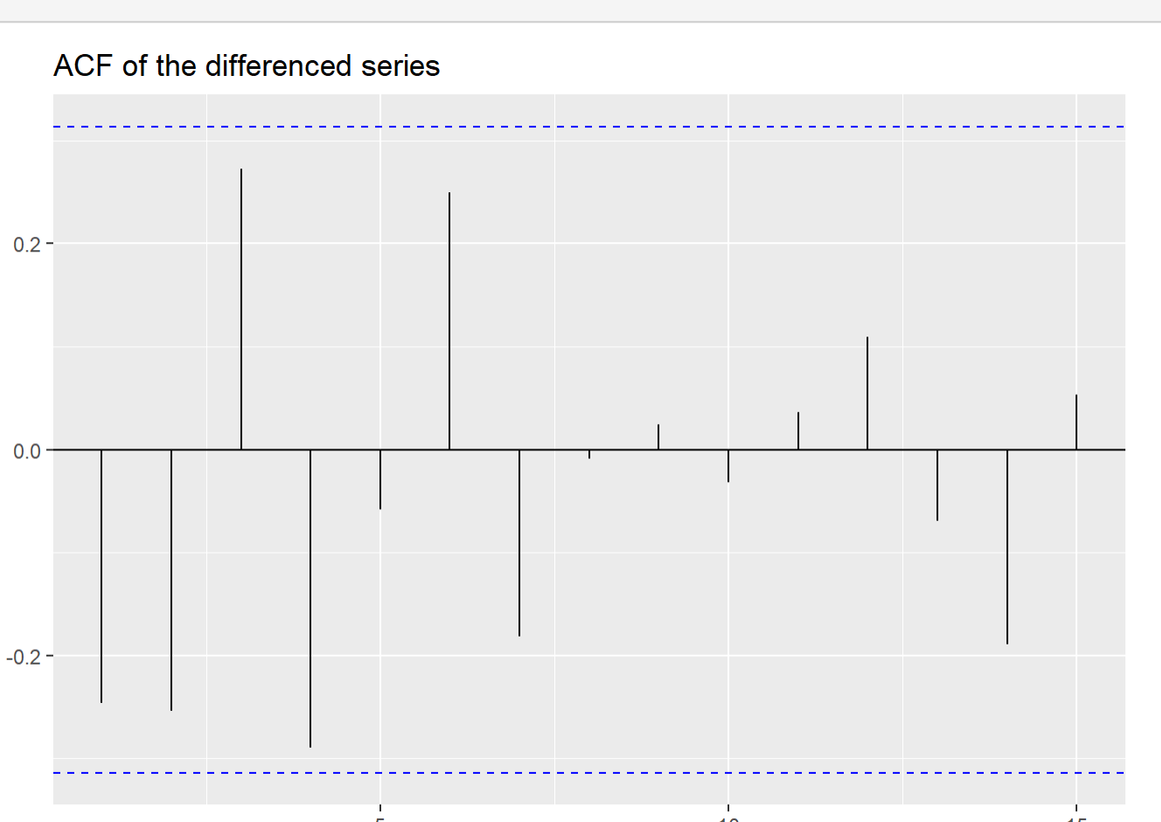
```
productivityTraining %>% features(Productivity, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

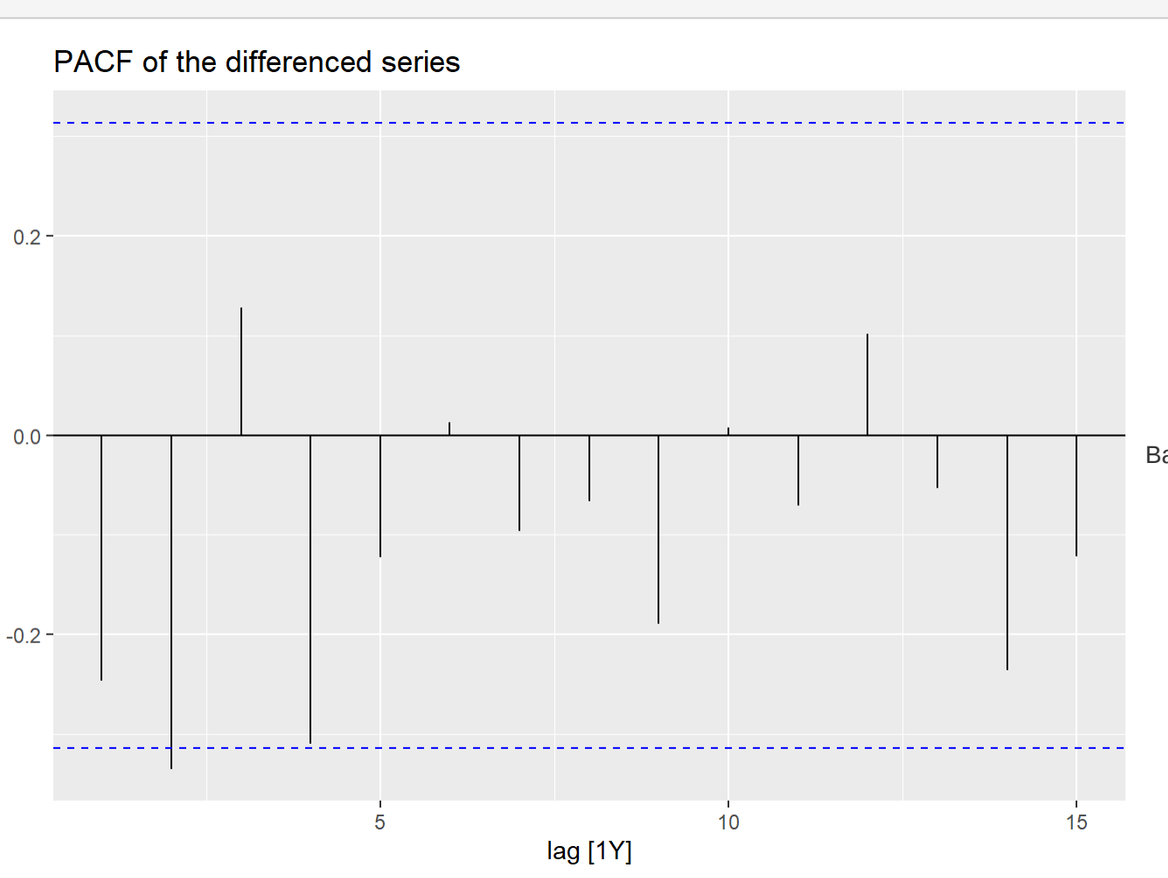
The KPSS test has a null hypothesis that the data is stationery and non-seasonal. We have a p - value = 0.01 which means that we have a strong evidence that the data is not stationery and seasonal. Therefore, differencing was applied so that p value is 0.1 which means that we have a strong evidence that the data is stationery. The order of differencing is 1.

Plot the ACF and PACF plots for the differenced data and comment on what you observe.

```
productivityTraining %>% ACF(Productivity %>% difference(1)) %>% autoplot() +
  labs(y = "ACF", title = "ACF of the differenced series")
```




```
productivityTraining %>% PACF(Productivity %>% difference(1)) %>% autoplot() +
  labs(y = "PACF", title = "PACF of the differenced series")
```



Based on ACF and PACF plots, we can determine the working model.

We see a a dampening sinusoidal pattern in the ACF plot, and a significant spike at lag 2 but none beyond lag 2.

Based on what you have learned on lectures, what ARIMA model would you suggest fitting to the training data?

Therefore, p = 2, and d = 1 so we should fit working model ARIMA(2,1,0) to the training data.

Write the equation of this model using backshift notation.

$(1-\phi_1 B-\phi_2 B^2)*(1-B)y_t = c$

Fit the following ARIMA models, compare them using information criteria, and write down the equation of the best model using backshift notation: [6 Marks]

```
fit = productivityTraining %>%
  model(ARIMA(Productivity ~ pdq(2,1,0)))

fit %>% report()
```

```
## Series: Productivity
## Model: ARIMA(2,1,0) w/ drift
##
## Coefficients:
##           ar1      ar2  constant
##       -0.3320  -0.3321  113.5754
## s.e.   0.1532   0.1503   21.4622
##
## sigma^2 estimated as 18401:  log likelihood=-239.09
## AIC=486.18   AICc=487.39   BIC=492.73
```

Your suggested model from part 1.

An automatic model using the stepwise algorithm.

```
autostepfit <- productivityTraining %>%
  model(stepwise = ARIMA(Productivity))

autostepfit %>% report()
```

```
## Series: Productivity
## Model: ARIMA(0,1,1) w/ drift
##
## Coefficients:
##           ma1  constant
##       -0.4888   68.8464
## s.e.   0.2121   11.4006
##
## sigma^2 estimated as 18873:  log likelihood=-240.09
## AIC=486.19   AICc=486.89   BIC=491.1
```

An automatic model without using the stepwise algorithm.

```
autofit <- productivityTraining %>%
  model(stepwise = ARIMA(Productivity, stepwise = FALSE))

autofit %>% report()
```

```
## Series: Productivity
## Model: ARIMA(0,1,1) w/ drift
##
## Coefficients:
##           ma1  constant
##       -0.4888   68.8464
## s.e.   0.2121   11.4006
##
## sigma^2 estimated as 18873:  log likelihood=-240.09
## AIC=486.19   AICc=486.89   BIC=491.1
```

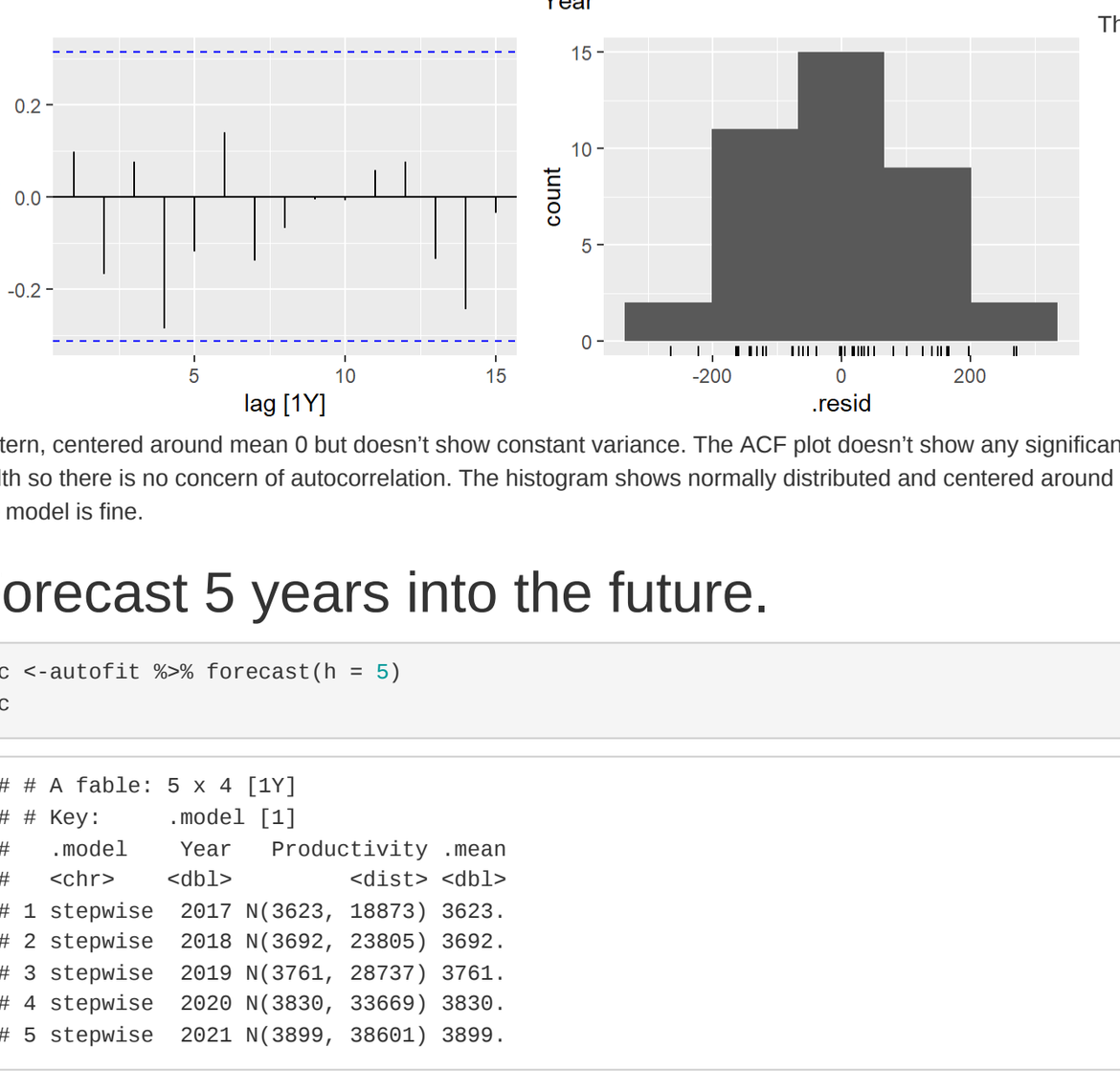
The automatic model using step wise and without step wise have both fitted ARIMA(0,1,1). They both have the same information criteria and AICc value of 486.89 which is lower than the manual fit (AICc=487.39). The best model using backshift notation $(1-B)y_t = c + (1+\phi_1 B)\epsilon_t$

Using the best model found in part 2, do the following: [5 Marks]

Conduct a diagnostic check on the residuals. Discuss whether or not you have any concerns about the model assumptions.

Based on the AICc criteria,

```
autofit %>% gg_tsresiduals()
```



The residual plot shows no apparent pattern, centered around mean 0 but doesn't show constant variance. The ACF plot doesn't show any significant point and they are within the band width so there is no concern of autocorrelation. The histogram shows normally distributed and centered around mean 0. Overall the assumption of the model is fine.

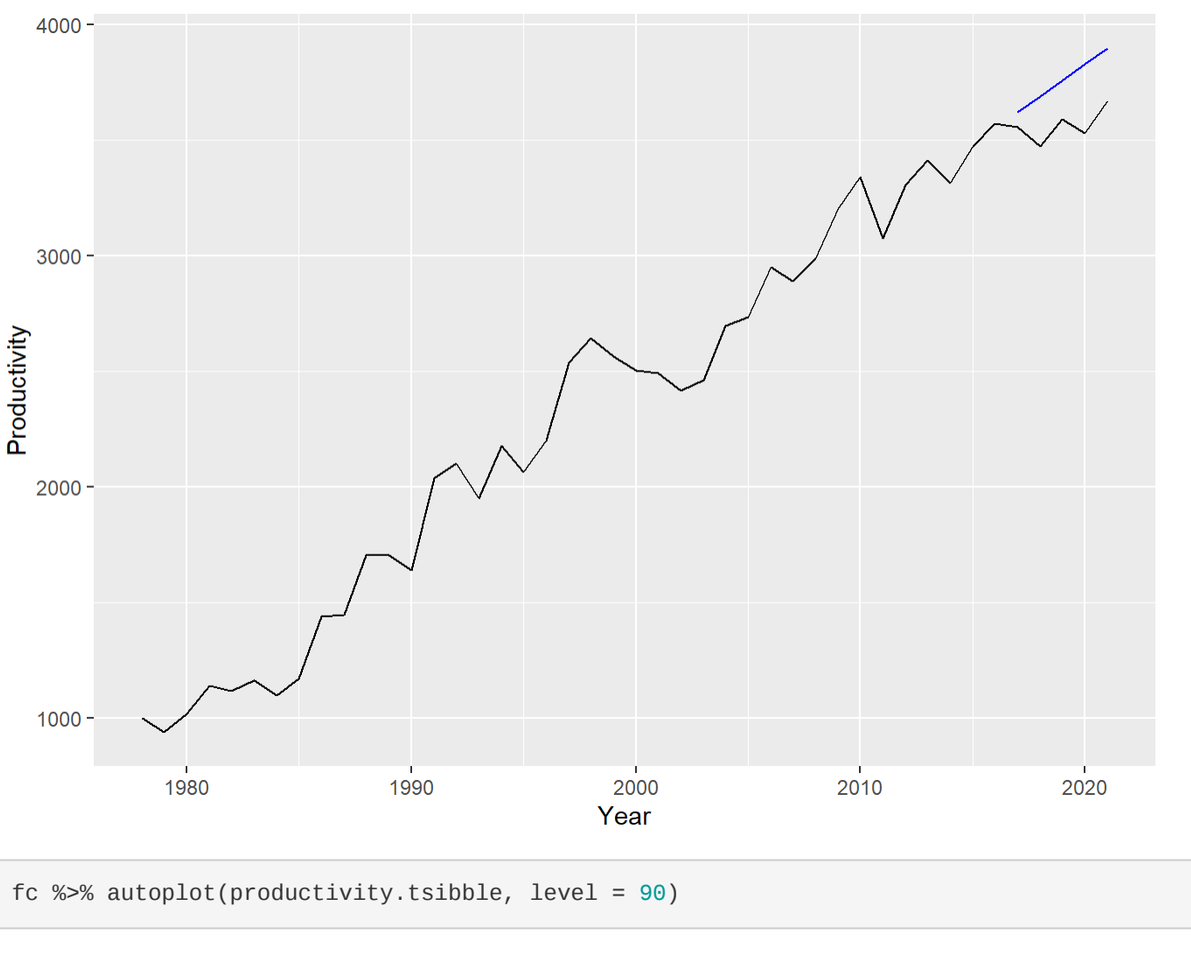
Forecast 5 years into the future.

```
fc <-autofit %>% forecast(h = 5)
fc
```

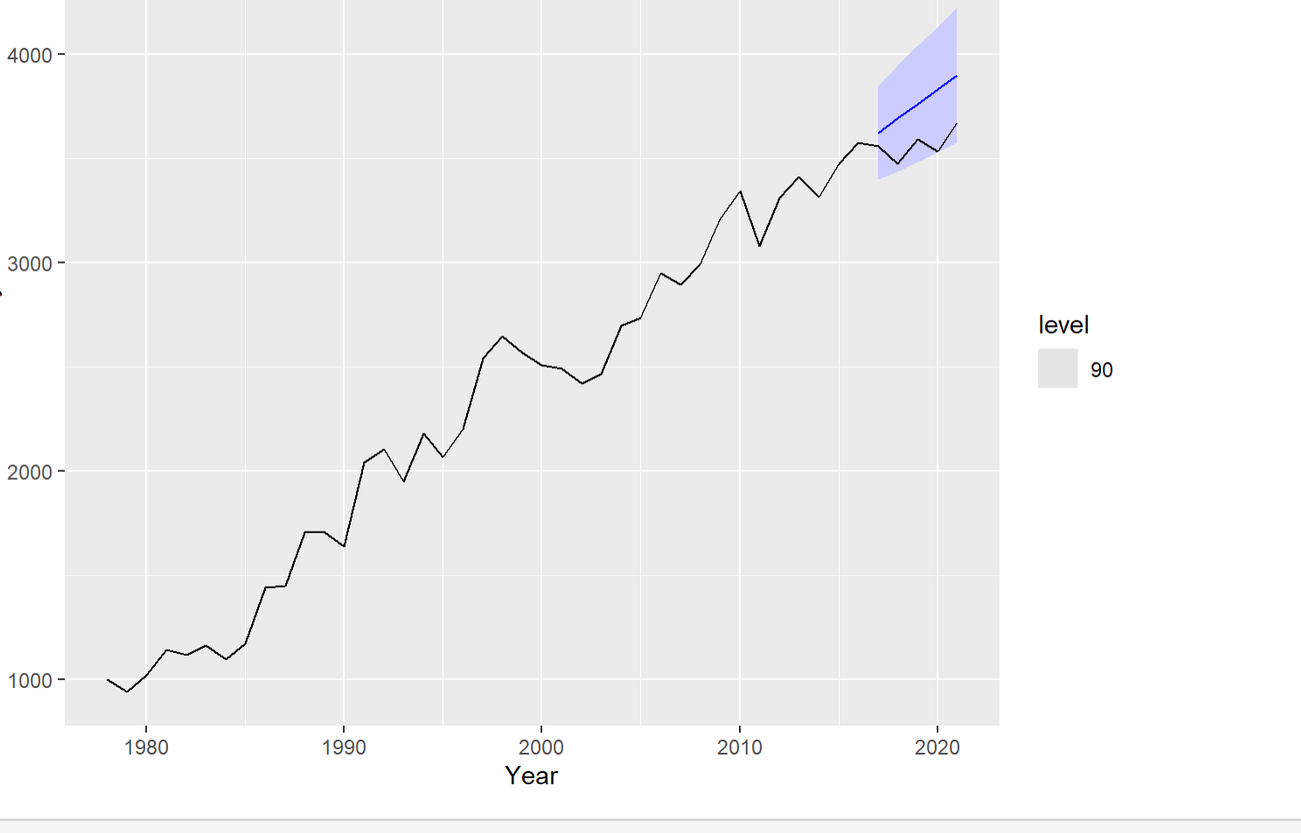
```
## # A fable: 5 x 4 [1Y]
## # Key:     .model [1]
##   .model    Year   Productivity .mean
##   <chr>    <dbl>         <dist> <dbl>
## 1 stepwise  2017 N(3623, 18873) 3623.
## 2 stepwise  2018 N(3692, 23805) 3692.
## 3 stepwise  2019 N(3761, 28737) 3761.
## 4 stepwise  2020 N(3830, 33669) 3830.
## 5 stepwise  2021 N(3899, 38601) 3899.
```

Overlay the original data with the point forecast and 90% and 99% prediction intervals.

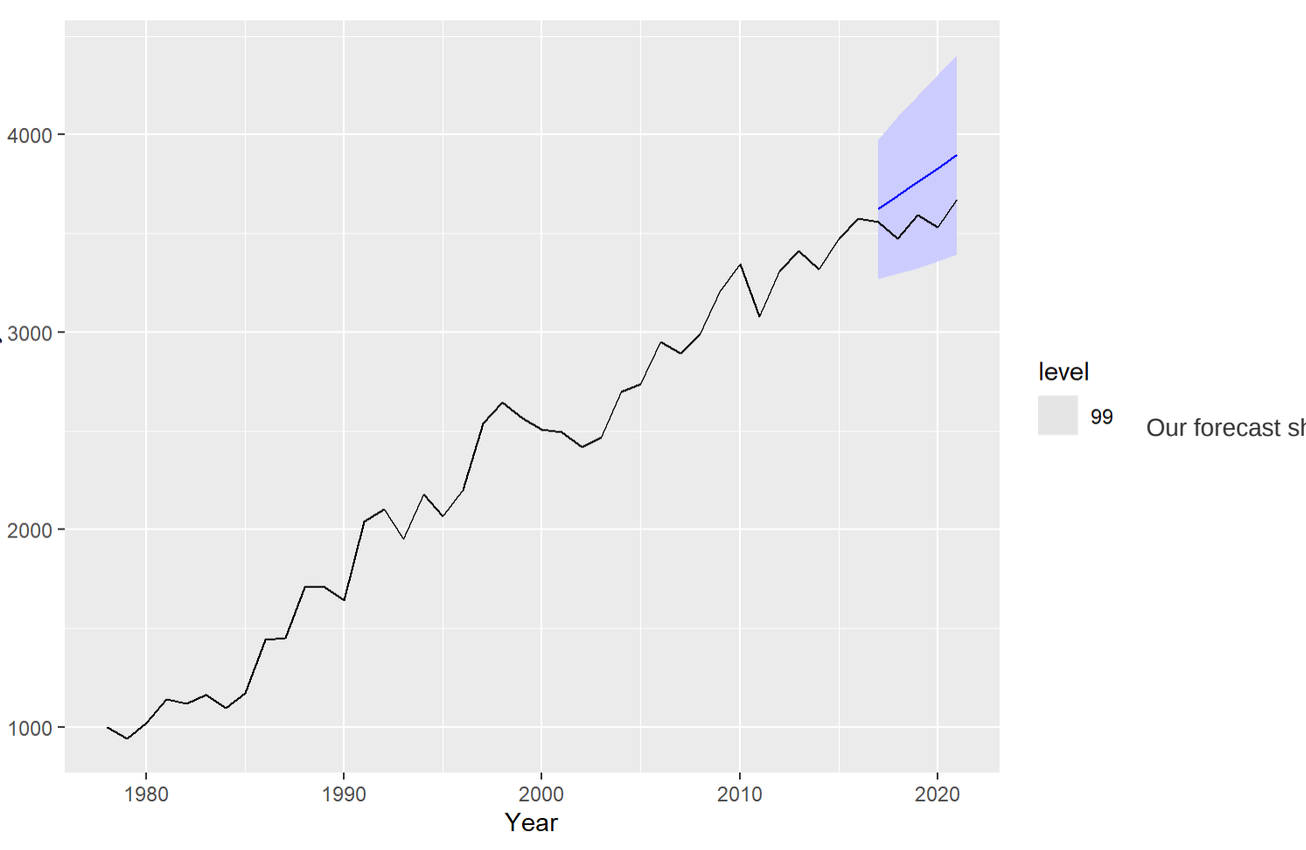
```
fc %>% autoplot(productivity.tsibble, level = NULL)
```




```
fc %>% autoplot(productivity.tsibble, level = 90)
```




```
fc %>% autoplot(productivity.tsibble, level = 99)
```



Our forecast shows overestimation than the actual data.