Week 9 homework 3_Alex_C_Parra

Alex Parra

19/7/2022

```
library(tsibble)
```

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

```
##
## Attaching package: 'tsibble'
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```
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 ggplot2     3.3.5
## v dplyr       1.0.7     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
```

```
## 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 lubridate::interval() masks tsibble::interval()
## x dplyr::lag()          masks stats::lag()
## x tsibble::setdiff()    masks base::setdiff()
## x tsibble::union()      masks base::union()
```

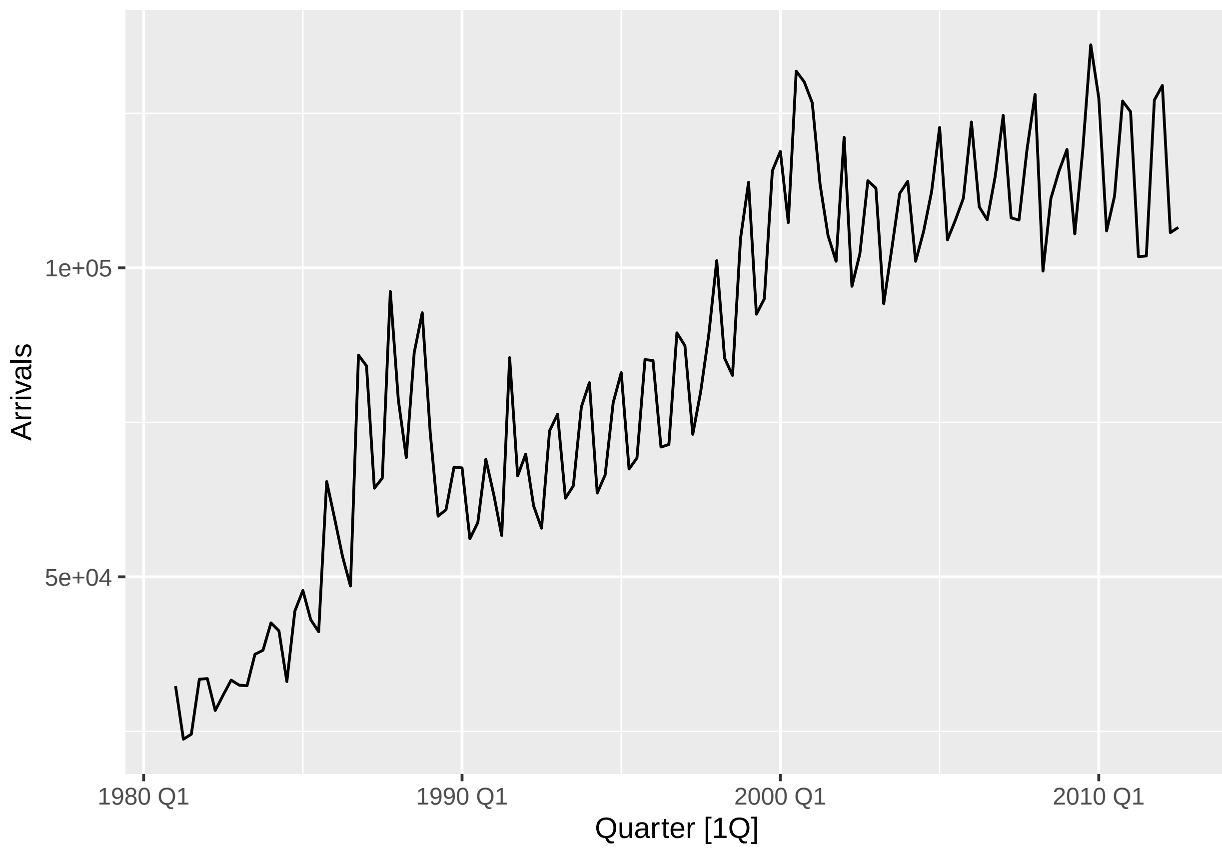
9.11 exercise 9

```r
us_aus_arrivals <- aus_arrivals %>%
  filter(Origin=='US' )

us_aus_arrivals
```

```
## # A tsibble: 127 x 3 [1Q]
## # Key:       Origin [1]
##    Quarter Origin Arrivals
##      <qtr> <chr>     <int>
##  1 1981 Q1 US        32316
##  2 1981 Q2 US        23721
##  3 1981 Q3 US        24533
##  4 1981 Q4 US        33438
##  5 1982 Q1 US        33527
##  6 1982 Q2 US        28366
##  7 1982 Q3 US        30856
##  8 1982 Q4 US        33293
##  9 1983 Q1 US        32472
## 10 1983 Q2 US        32369
## # ... with 117 more rows
```

a)

```r
us_aus_arrivals %>%
  autoplot(Arrivals)
```



We can see an ascending trend in the data with some seasonality factor in play, after 200 we can see that the trend slows down and even stop.

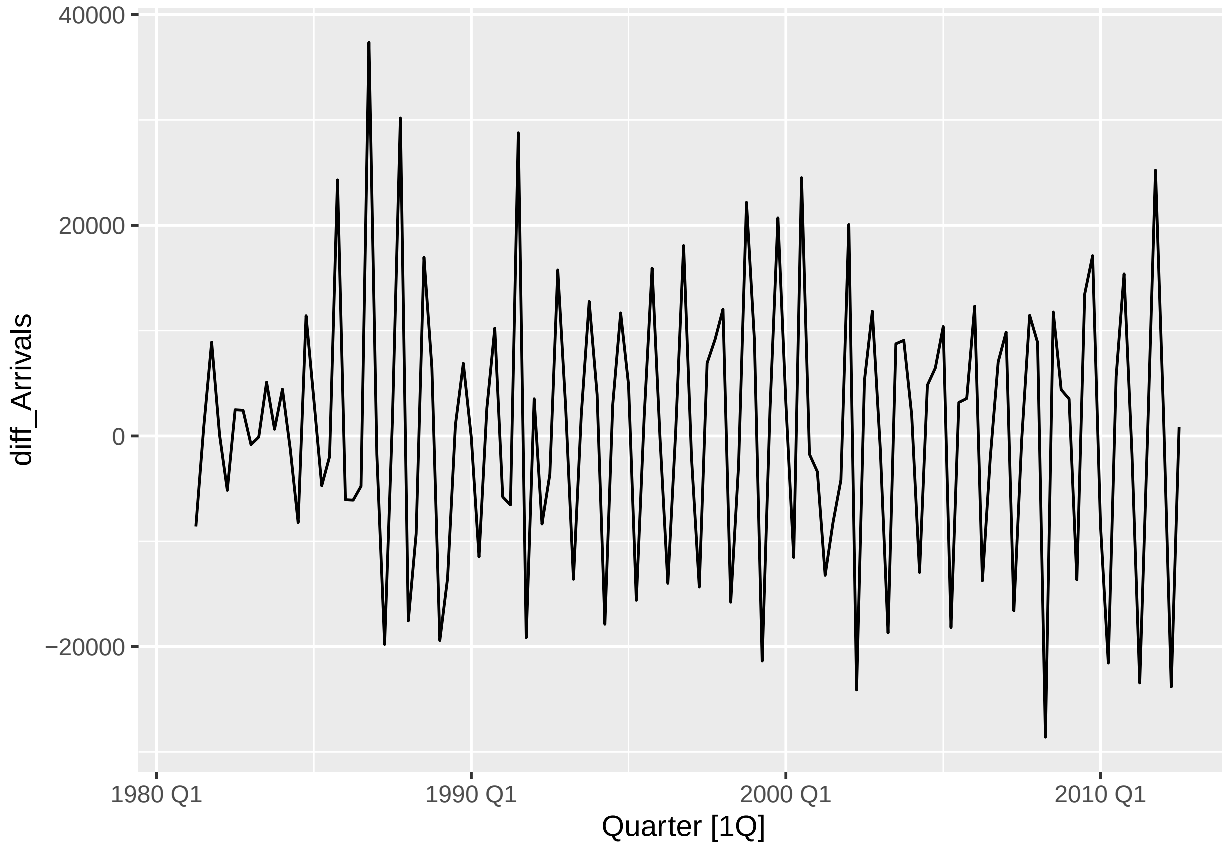
b)

```
us_aus_arrivals %>%
  mutate(diff_Arrivals = difference(Arrivals)) -> us_aus_arrivals_diff
us_aus_arrivals_diff
```

```
## # A tsibble: 127 x 4 [1Q]
## # Key:       Origin [1]
##    Quarter Origin Arrivals diff_Arrivals
##      <qtr> <chr>     <int>         <int>
##  1 1981 Q1 US        32316            NA
##  2 1981 Q2 US        23721         -8595
##  3 1981 Q3 US        24533           812
##  4 1981 Q4 US        33438          8905
##  5 1982 Q1 US        33527            89
##  6 1982 Q2 US        28366         -5161
##  7 1982 Q3 US        30856          2490
##  8 1982 Q4 US        33293          2437
##  9 1983 Q1 US        32472          -821
## 10 1983 Q2 US        32369          -103
## # ... with 117 more rows
```

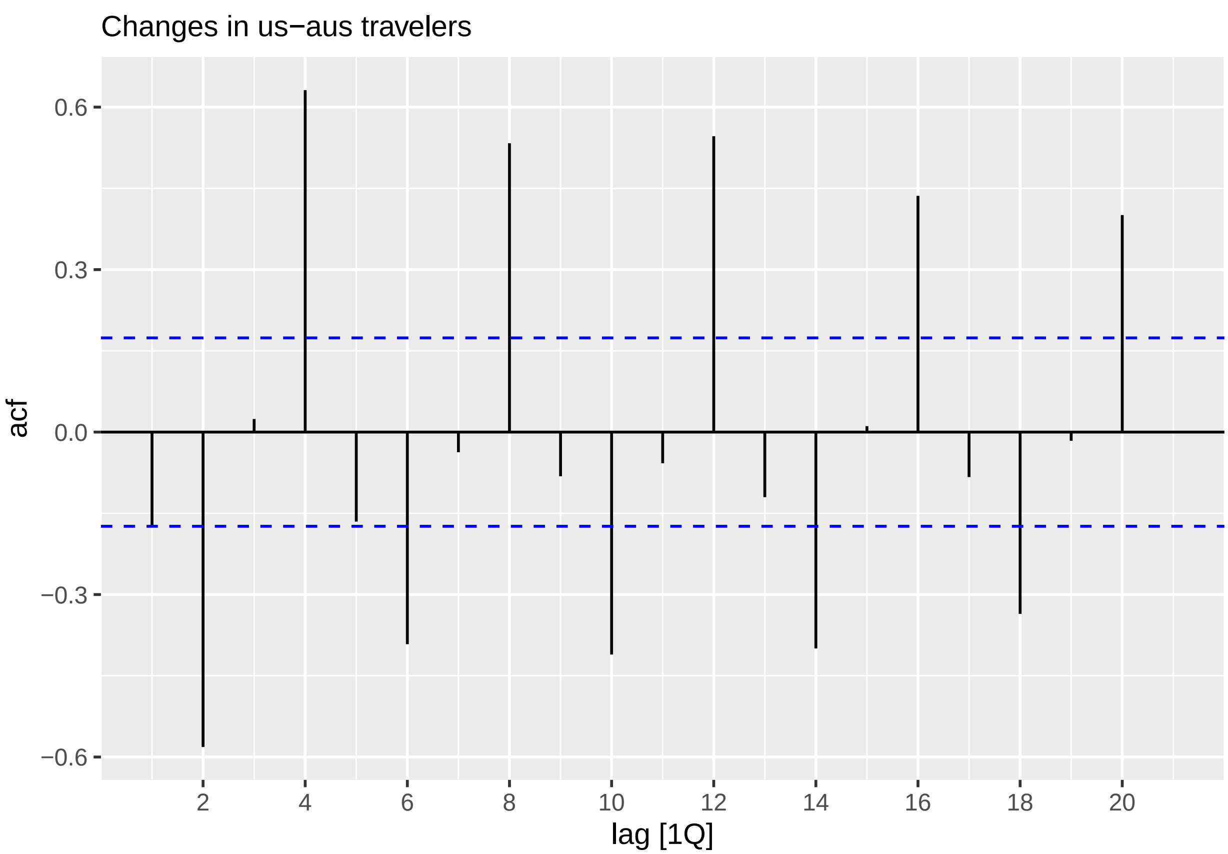
```r
us_aus_arrivals_diff %>%
  autoplot(diff_Arrivals)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



c)

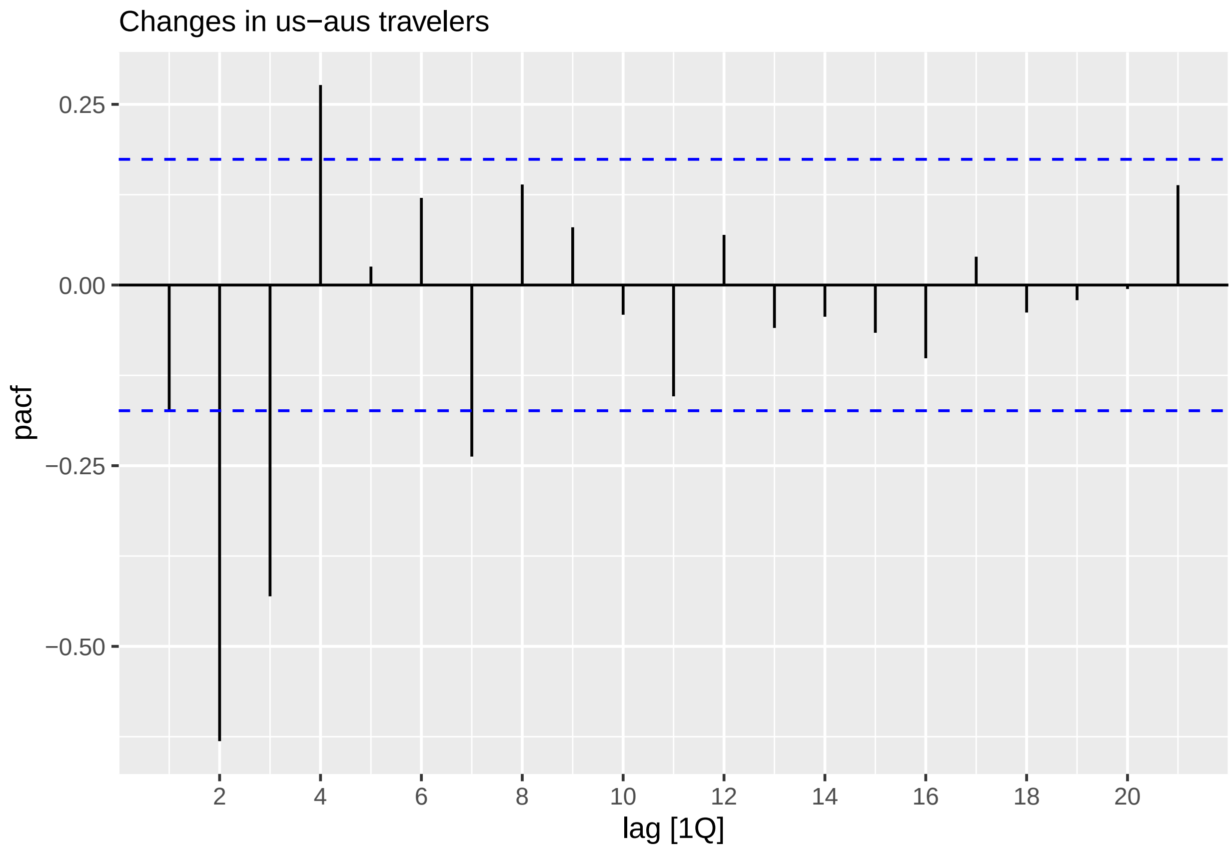
```r
us_aus_arrivals_diff %>% ACF(diff_Arrivals) %>%
  autoplot() + labs(subtitle = "Changes in us-aus travelers")
```



We can see that there are multiple autocorrelation that lies outside the 95% limits, this repeats every 2 observation, First in the negative at lag 2, then positive at lag 4, then again negative at lag 6, and so on.

d)

```
us_aus_arrivals_diff %>% PACF(diff_Arrivals) %>%
  autoplot() + labs(subtitle = "Changes in us-aus travelers")
```



If we look instead at the partial correlation, we can see that there are multiple spikes at the biggening, at lag 2, 3, 4 and then one last at 7, after that we can see that there isn't any other.

e) We can see that the ACF is sinusoidal, and there is a spike at lag 7 in the PACF, meaning that the model is most likely p=7, d=0, q=0 ARIMA(7,0,0). d is 0 because we have already differentiated before.

f)

I tried creating the model with p=7, but it gave me and error, so I tried 4

```
fit_410 <- us_aus_arrivals_diff %>%
  model('arima410' = ARIMA(diff_Arrivals ~ pdq(4,0,0)))

report(fit_410)
```

```
## Series: diff_Arrivals
## Model: ARIMA(4,0,0)(0,1,1)[4]
##
## Coefficients:
##           ar1      ar2     ar3     ar4     sma1
##       -0.4379  -0.3472  0.0001  0.1923  -0.8918
## s.e.   0.0922   0.0989  0.1025  0.0989   0.0660
##
## sigma^2 estimated as 56674753:  log likelihood=-1262.88
## AIC=2537.76   AICc=2538.48   BIC=2554.63
```

```r
print('')
```

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

```r
fit_auto <- us_aus_arrivals_diff %>%
  model('auto' = ARIMA(diff_Arrivals))

report(fit_auto)
```

```
## Series: diff_Arrivals
## Model: ARIMA(2,0,0)(1,1,1)[4]
##
## Coefficients:
##           ar1      ar2    sar1     sma1
##       -0.4155  -0.3843  0.1890  -0.8869
## s.e.   0.0834   0.0837  0.1088   0.0637
##
## sigma^2 estimated as 56796711:  log likelihood=-1263.35
## AIC=2536.71   AICc=2537.22   BIC=2550.77
```

We can see that the model with the best AIC and AICc is the auto model. But the difference is not by much, the main difference between the auto model and the model we chose is that the p component in the auto model is 2 and not 4.

9.11 exercise 10

```r
us_employment %>% filter(Series_ID == 'CEU0500000001') -> ex10
ex10
```

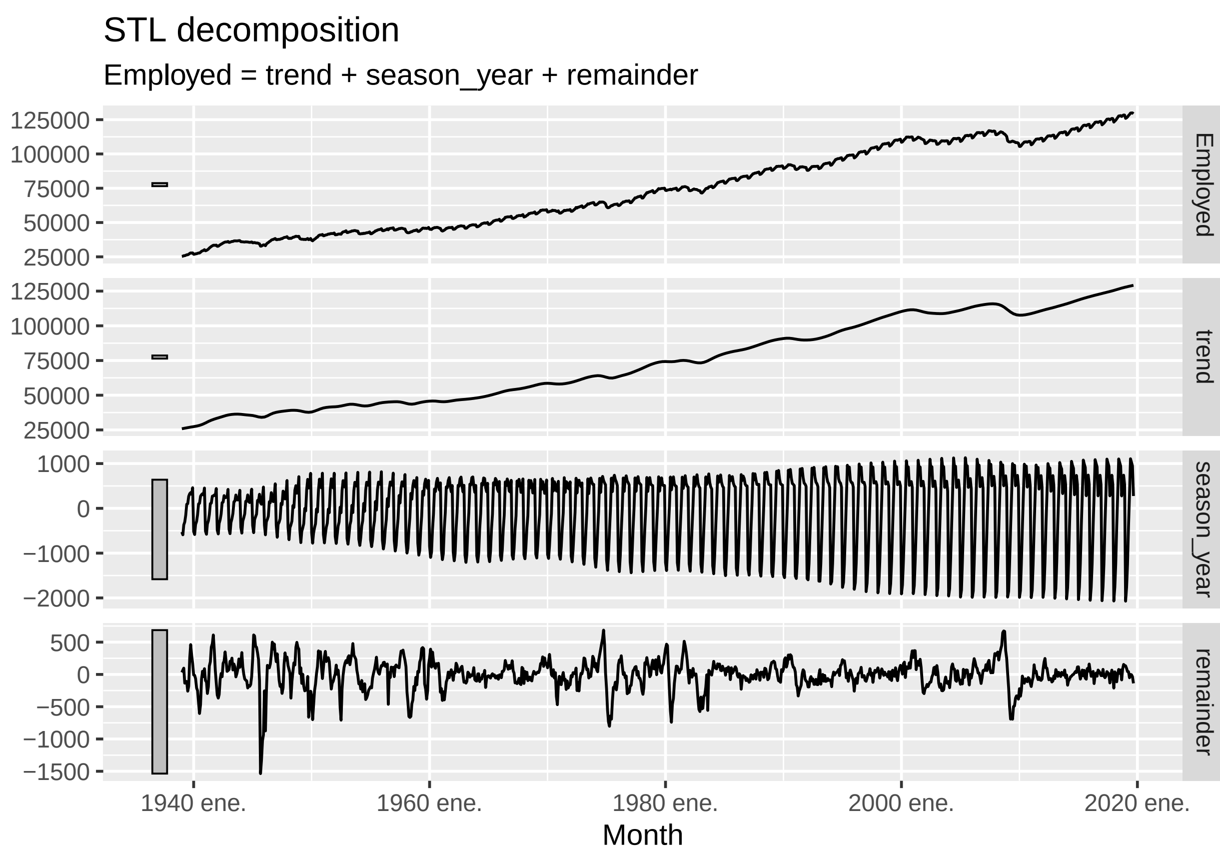
```
## # A tsibble: 969 x 4 [1M]
## # Key:       Series_ID [1]
##        Month Series_ID     Title         Employed
##        <mth> <chr>         <chr>            <dbl>
##  1 1939 ene. CEU0500000001 Total Private    25338
##  2 1939 feb. CEU0500000001 Total Private    25447
##  3 1939 mar. CEU0500000001 Total Private    25833
##  4 1939 abr. CEU0500000001 Total Private    25801
##  5 1939 may. CEU0500000001 Total Private    26113
##  6 1939 jun. CEU0500000001 Total Private    26485
##  7 1939 jul. CEU0500000001 Total Private    26481
##  8 1939 ago. CEU0500000001 Total Private    26848
##  9 1939 sep. CEU0500000001 Total Private    27468
## 10 1939 oct. CEU0500000001 Total Private    27830
## # ... with 959 more rows
```

a)

```r
dcmp <- ex10 %>%
  model(stl = STL(Employed))
components(dcmp)
```

```
## # A dable: 969 x 8 [1M]
## # Key:     Series_ID, .model [1]
## # :        Employed = trend + season_year + remainder
##    Series_ID     .model     Month Employed  trend season_year remainder
##    <chr>         <chr>      <mth>    <dbl>  <dbl>       <dbl>     <dbl>
##  1 CEU0500000001 stl    1939 ene.    25338 25835.      -528.       31.8
##  2 CEU0500000001 stl    1939 feb.    25447 25970.      -590.       67.3
##  3 CEU0500000001 stl    1939 mar.    25833 26105.      -370.       97.4
##  4 CEU0500000001 stl    1939 abr.    25801 26240.      -305.     -135.
##  5 CEU0500000001 stl    1939 may.    26113 26372.      -152.     -108.
##  6 CEU0500000001 stl    1939 jun.    26485 26504.        90.4    -110.
##  7 CEU0500000001 stl    1939 jul.    26481 26636.       104.     -260.
##  8 CEU0500000001 stl    1939 ago.    26848 26763.       268.     -184.
##  9 CEU0500000001 stl    1939 sep.    27468 26890.       327.      251.
## 10 CEU0500000001 stl    1939 oct.    27830 27017.       351.      462.
## # ... with 959 more rows, and 1 more variable: season_adjust <dbl>
```

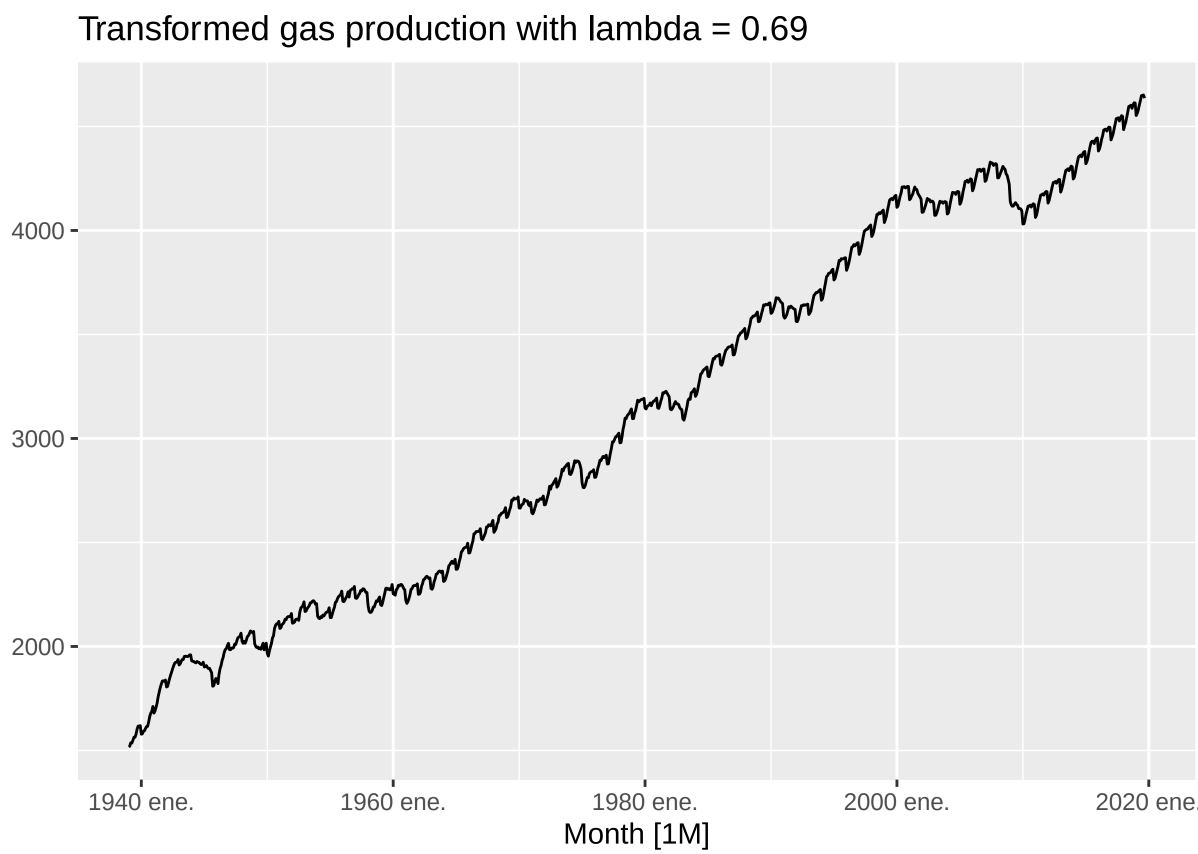
```r
components(dcmp) %>% autoplot()
```



We can see a positive trend in the data, that increase in all the time series. In the seasonality we can see and alternating pattern, a pattern that get amplified as time goes on, meaning that the seasons are more variables the later in time we look

b)

```
lambda <- ex10 %>%
  features(Employed, features = guerrero) %>%
  pull(lambda_guerrero)
ex10 %>%
  autoplot(box_cox(Employed, lambda)) +
  labs(y = "",
       title = paste0(
         "Transformed gas production with lambda = ",
         round(lambda,2)))
```



The Box-Cox seams like a good transformation as it will allow us to make the seasonality the same across the hole time series

```
ex10 %>%
  mutate(box_cox_Employed = box_cox(Employed, lambda)) -> ex10_box_cox
ex10_box_cox
```

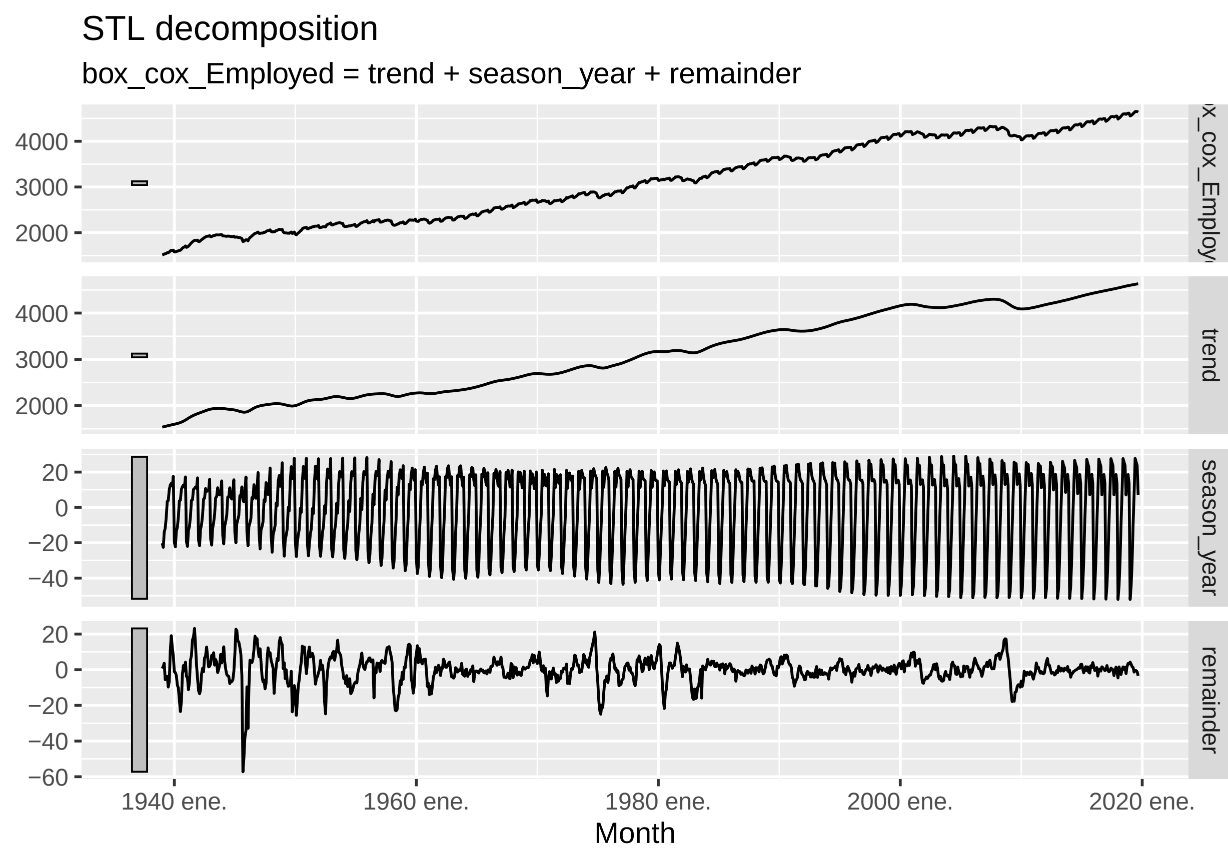
```
## # A tsibble: 969 x 5 [1M]
## # Key:       Series_ID [1]
##        Month Series_ID     Title         Employed box_cox_Employed
##        <mth> <chr>         <chr>            <dbl>            <dbl>
##   1 1939 ene. CEU0500000001 Total Private    25338            1517.
##   2 1939 feb. CEU0500000001 Total Private    25447            1521.
##   3 1939 mar. CEU0500000001 Total Private    25833            1537.
```

```
##  4 1939 abr. CEU0500000001 Total Private    25801            1536.
##  5 1939 may. CEU0500000001 Total Private    26113            1548.
##  6 1939 jun. CEU0500000001 Total Private    26485            1563.
##  7 1939 jul. CEU0500000001 Total Private    26481            1563.
##  8 1939 ago. CEU0500000001 Total Private    26848            1578.
##  9 1939 sep. CEU0500000001 Total Private    27468            1603.
## 10 1939 oct. CEU0500000001 Total Private    27830            1617.
## # ... with 959 more rows
```

```
dcmp <- ex10_box_cox %>%
  model(stl = STL(box_cox_Employed))
components(dcmp)
```

```
## # A dable: 969 x 8 [1M]
## # Key:     Series_ID, .model [1]
## # :        box_cox_Employed = trend + season_year + remainder
##    Series_ID     .model     Month box_cox_Employed trend season_year remainder
##    <chr>         <chr>      <mth>            <dbl> <dbl>       <dbl>     <dbl>
##  1 CEU0500000001 stl    1939 ene.            1517. 1536.      -20.3      0.663
##  2 CEU0500000001 stl    1939 feb.            1521. 1542.      -22.6      1.91
##  3 CEU0500000001 stl    1939 mar.            1537. 1547.      -14.1      3.62
##  4 CEU0500000001 stl    1939 abr.            1536. 1553.      -11.7     -5.62
##  5 CEU0500000001 stl    1939 may.            1548. 1558.       -5.75    -4.28
##  6 CEU0500000001 stl    1939 jun.            1563. 1564.        3.41    -3.75
##  7 CEU0500000001 stl    1939 jul.            1563. 1569.        3.81    -9.72
##  8 CEU0500000001 stl    1939 ago.            1578. 1574.       10.1     -6.42
##  9 CEU0500000001 stl    1939 sep.            1603. 1580.       12.6     10.8
## 10 CEU0500000001 stl    1939 oct.            1617. 1585.       13.6     19.1
## # ... with 959 more rows, and 1 more variable: season_adjust <dbl>
```

```
components(dcmp) %>% autoplot()
```



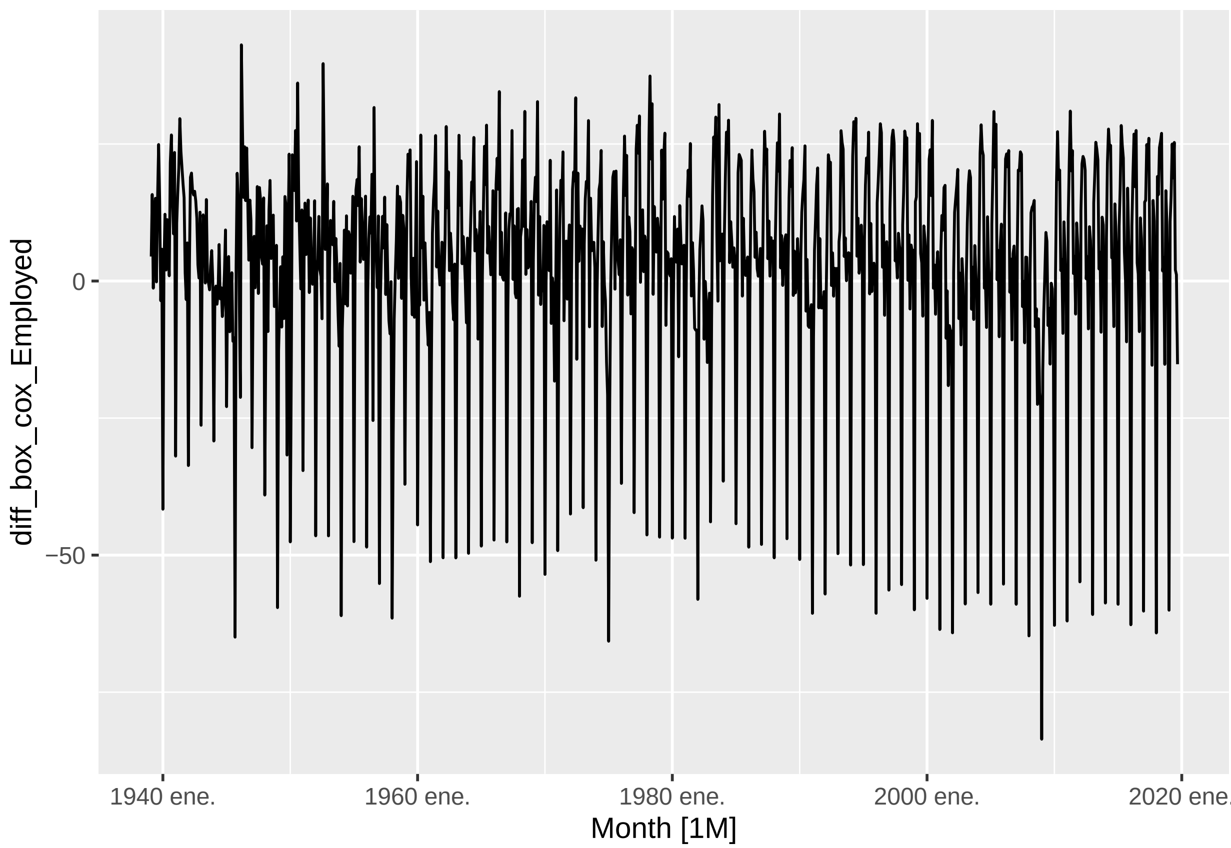
c)

```
ex10_box_cox %>%
  mutate(diff_box_cox_Employed = difference(box_cox_Employed)) -> ex10_box_cox_diff
ex10_box_cox_diff
```

```
## # A tsibble: 969 x 6 [1M]
## # Key:       Series_ID [1]
##         Month Series_ID     Title      Employed box_cox_Employed diff_box_cox_Em~
##         <mth> <chr>         <chr>         <dbl>            <dbl>            <dbl>
##   1 1939 ene. CEU0500000001 Total Pri~    25338            1517.           NA
##   2 1939 feb. CEU0500000001 Total Pri~    25447            1521.            4.47
##   3 1939 mar. CEU0500000001 Total Pri~    25833            1537.           15.8
##   4 1939 abr. CEU0500000001 Total Pri~    25801            1536.           -1.31
##   5 1939 may. CEU0500000001 Total Pri~    26113            1548.           12.7
##   6 1939 jun. CEU0500000001 Total Pri~    26485            1563.           15.1
##   7 1939 jul. CEU0500000001 Total Pri~    26481            1563.           -0.162
##   8 1939 ago. CEU0500000001 Total Pri~    26848            1578.           14.8
##   9 1939 sep. CEU0500000001 Total Pri~    27468            1603.           24.9
## 10 1939 oct. CEU0500000001 Total Pri~    27830            1617.           14.5
## # ... with 959 more rows
```

```
ex10_box_cox_diff %>%
  autoplot(diff_box_cox_Employed)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

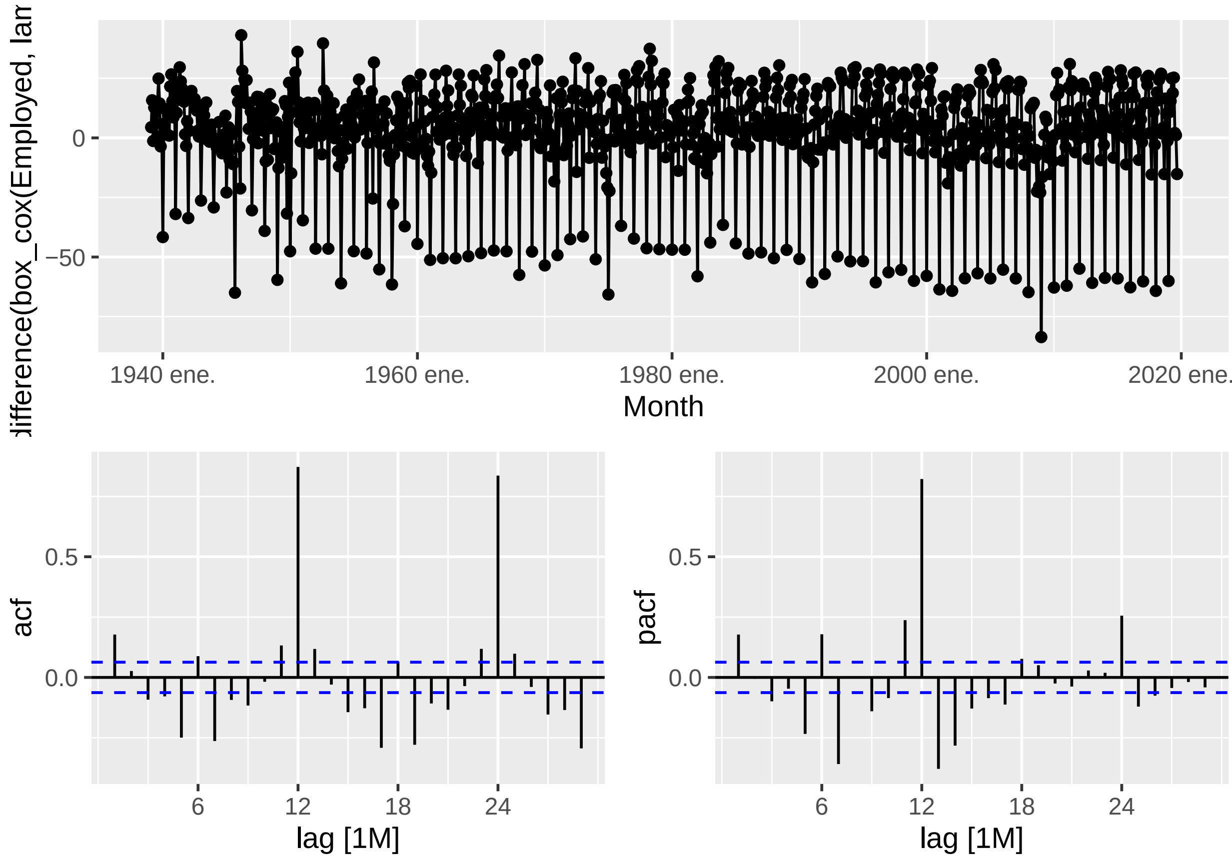


d)

```
ex10 %>%
  gg_tsdisplay(difference(box_cox(Employed, lambda)), plot_type='partial')
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
fit <- ex10 %>%
  model('arima010' = ARIMA(box_cox(Employed, lambda) ~ pdq(0,1,0)),
        'arima110' = ARIMA(box_cox(Employed, lambda) ~ pdq(1,1,0)),
        'arima210' = ARIMA(box_cox(Employed, lambda) ~ pdq(2,1,0)),
        'arima011' = ARIMA(box_cox(Employed, lambda) ~ pdq(0,1,1)),
        'arima012' = ARIMA(box_cox(Employed, lambda) ~ pdq(0,1,1)),
        'auto'     = ARIMA(box_cox(Employed, lambda), stepwise=FALSE))
report (fit)
```

```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 6 x 9
##   Series_ID     .model   sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>         <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 CEU0500000001 arima010   61.4  -3331. 6667. 6667. 6682. <cpl [0]>  <cpl [24]>
## 2 CEU0500000001 arima110   50.3  -3233. 6474. 6474. 6493. <cpl [1]>  <cpl [24]>
## 3 CEU0500000001 arima210   46.4  -3193. 6399. 6400. 6433. <cpl [26]> <cpl [24]>
## 4 CEU0500000001 arima011   54.2  -3270. 6548. 6548. 6567. <cpl [0]>  <cpl [25]>
## 5 CEU0500000001 arima012   54.2  -3270. 6548. 6548. 6567. <cpl [0]>  <cpl [25]>
## 6 CEU0500000001 auto       44.9  -3182. 6380. 6380. 6419. <cpl [26]> <cpl [13]>
```

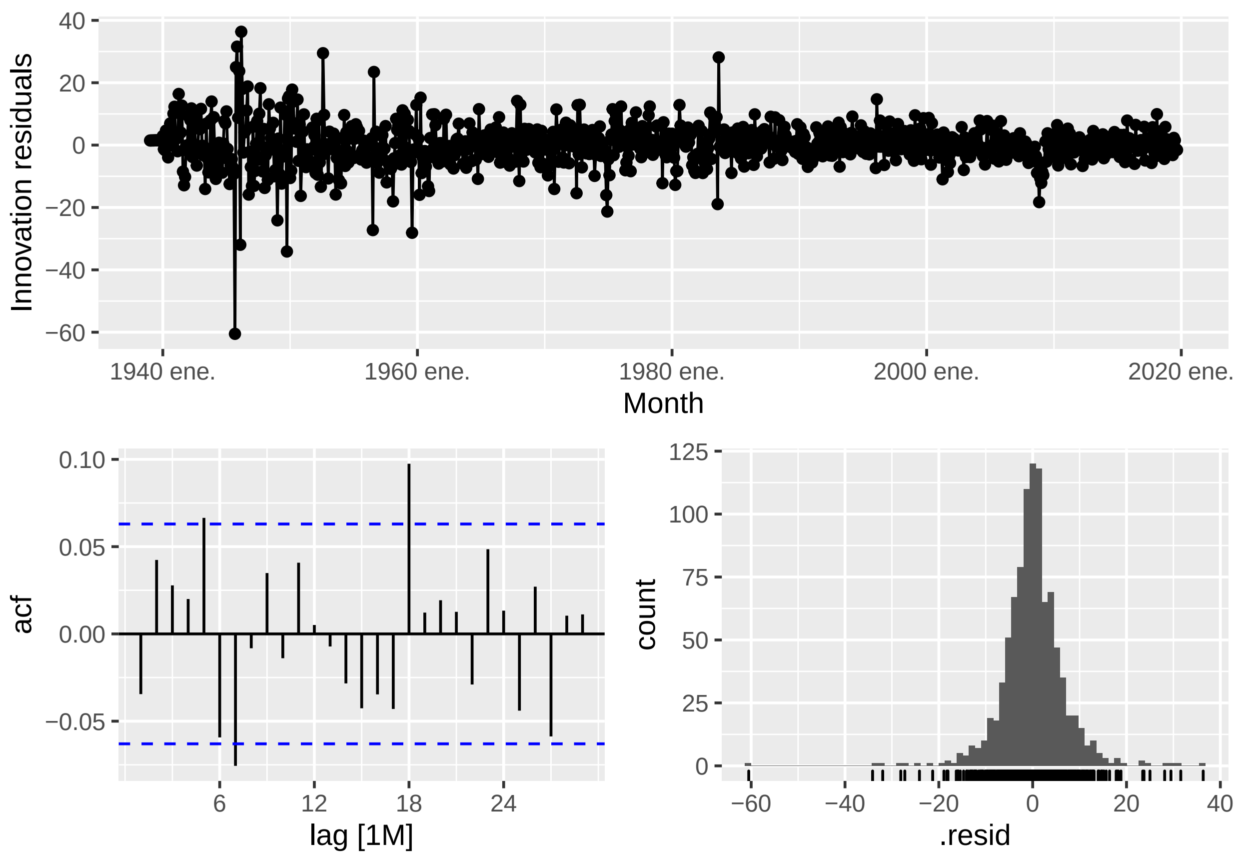
We can see that the best model is the auto model with the lowest AIC (6380.17) and AICc (6380.32), followed by the ARIMA 210.

e)

```
fit <- ex10 %>%
  model(ARIMA(box_cox(Employed, lambda), stepwise=FALSE))

report(fit)
```

```
## Series: Employed
## Model: ARIMA(2,0,1)(2,1,1)[12] w/ drift
## Transformation: box_cox(Employed, lambda)
##
## Coefficients:
##          ar1      ar2      ma1    sar1    sar2     sma1  constant
##       1.8687  -0.8742  -0.5781  0.1660  0.0655  -0.8390    0.1582
## s.e.  0.0241   0.0240   0.0388  0.0446  0.0402   0.0293    0.0153
##
## sigma^2 estimated as 44.94:  log likelihood=-3182.09
## AIC=6380.17   AICc=6380.32   BIC=6419.08
```

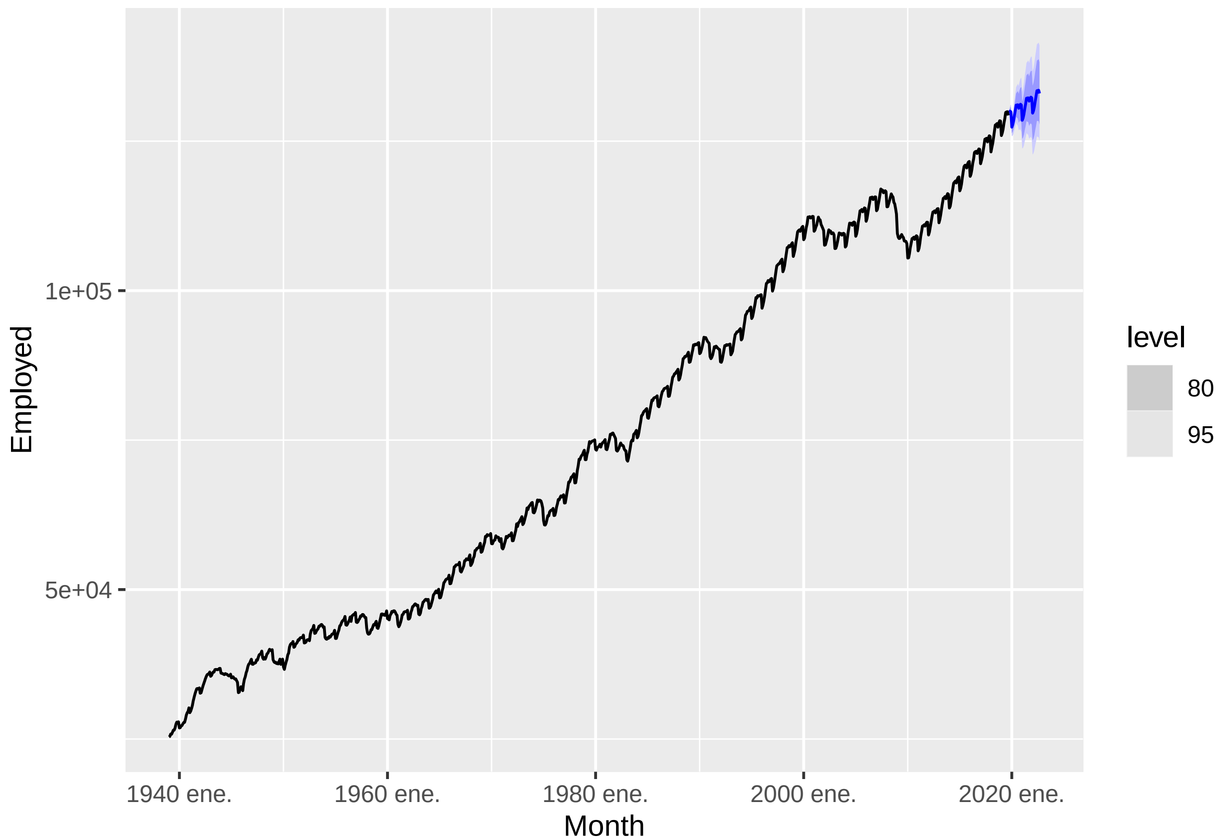
```
fit %>% gg_tsresiduals()
```



We can see that the auto model is a (2,0,1)(2,1,1) ARIMA. If we look at the residuals we can see that they follow a normal distribution, with most values around the 0 value, but there is a couples of autocorrelation at lag 7 and 18.

f)

```
fc <- fit %>%
  forecast(h = 36)

fc %>%
  autoplot(ex10)
```



g) The predictions intervals are going to grow as the time prediction increases, meaning that the further in the future we are the harder is to make a prediction. So I don't think there is a hard limit after that the predictions are unusable, is more gradual than that. But you can use it to make predictions with in the next year and get good results.

9.11 exercise 11

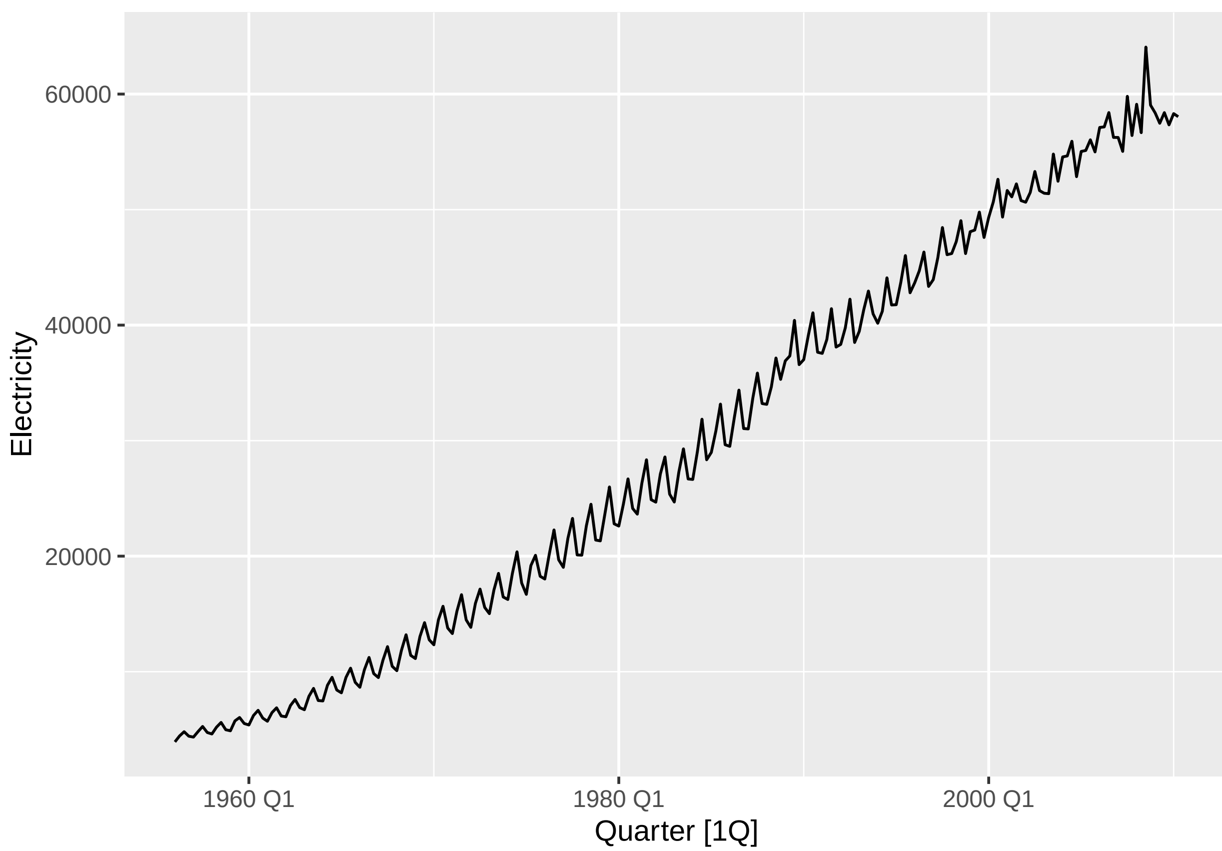
```
aus_production %>% select(Quarter, Electricity) -> ex11

ex11
```

```
## # A tsibble: 218 x 2 [1Q]
##    Quarter Electricity
##      <qtr>       <dbl>
##  1 1956 Q1        3923
##  2 1956 Q2        4436
##  3 1956 Q3        4806
```

```
##   4 1956 Q4        4418
##   5 1957 Q1        4339
##   6 1957 Q2        4811
##   7 1957 Q3        5259
##   8 1957 Q4        4735
##   9 1958 Q1        4608
##  10 1958 Q2        5196
## # ... with 208 more rows
```

a)

```
ex11 %>% autoplot(Electricity)
```



The data doesn't look like it needs a transformation.

b)

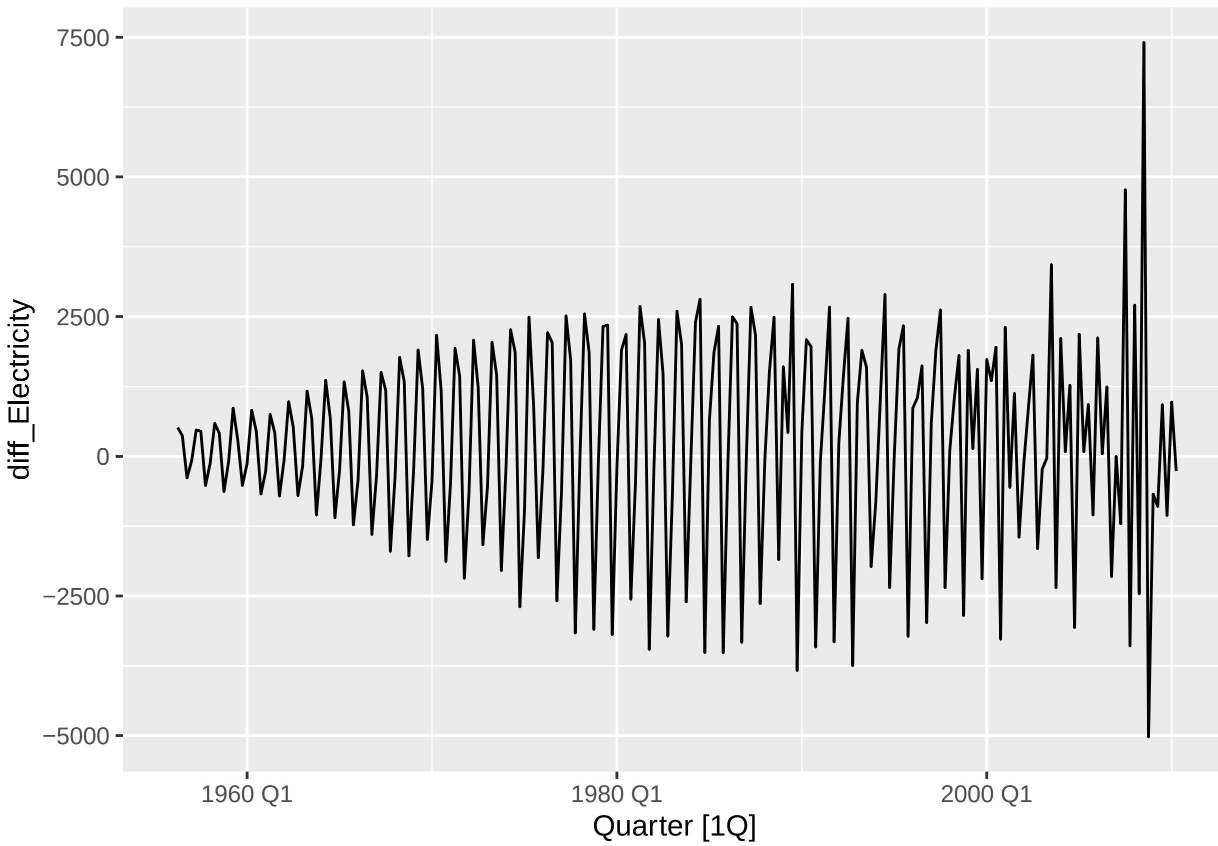
```
ex11 %>%
  mutate(diff_Electricity = difference(Electricity)) -> ex11_diff

ex11_diff
```

```
## # A tsibble: 218 x 3 [1Q]
##    Quarter Electricity diff_Electricity
##      <qtr>       <dbl>            <dbl>
##   1 1956 Q1        3923               NA
```

```
##   2 1956 Q2        4436              513
##   3 1956 Q3        4806              370
##   4 1956 Q4        4418             -388
##   5 1957 Q1        4339              -79
##   6 1957 Q2        4811              472
##   7 1957 Q3        5259              448
##   8 1957 Q4        4735             -524
##   9 1958 Q1        4608             -127
##  10 1958 Q2        5196              588
## # ... with 208 more rows
```

```
ex11_diff %>%
  autoplot(diff_Electricity)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

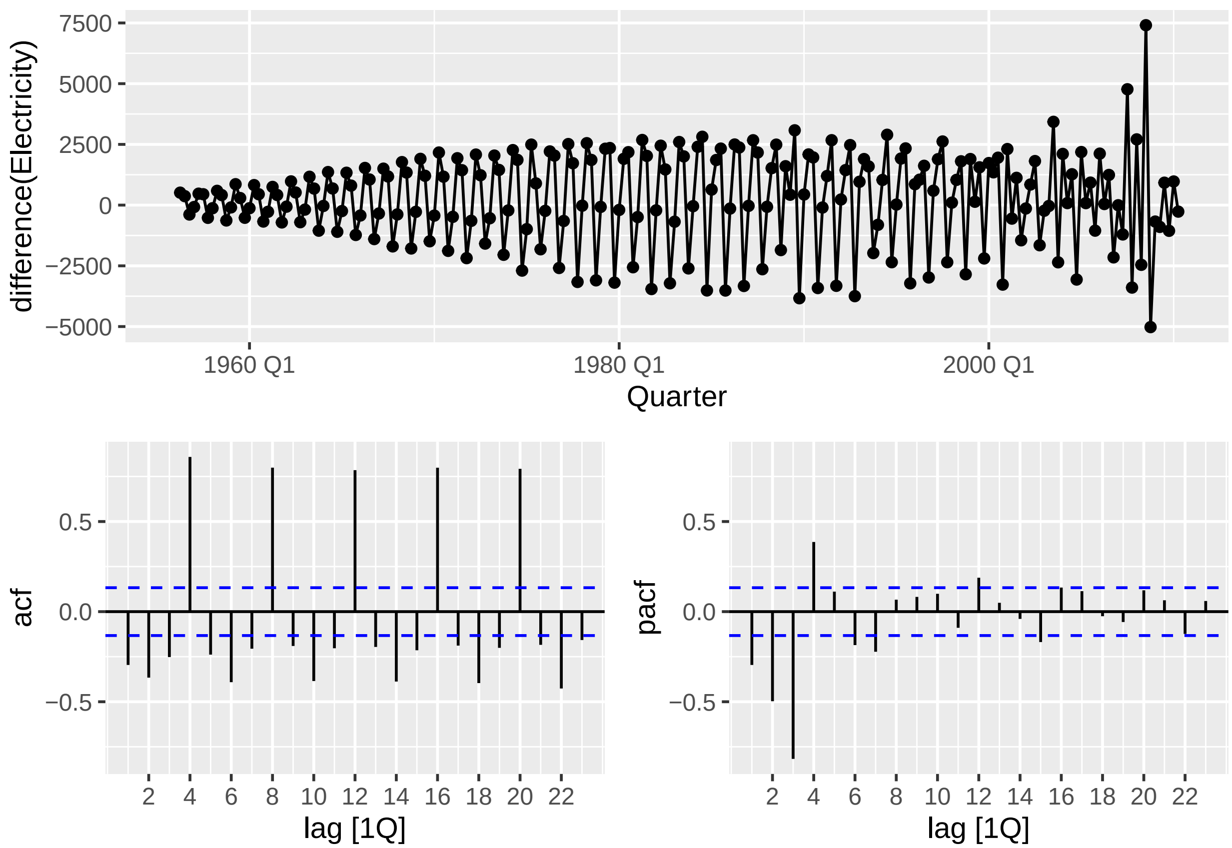


c)

```
ex11 %>%
  gg_tsdisplay(difference(Electricity), plot_type='partial')
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
fit <- ex11 %>%
  model('arima010' = ARIMA(Electricity ~ pdq(0,1,0)),
        'arima110' = ARIMA(Electricity ~ pdq(1,1,0)),
        'arima210' = ARIMA(Electricity ~ pdq(2,1,0)),
        'arima011' = ARIMA(Electricity ~ pdq(0,1,1)),
        'arima012' = ARIMA(Electricity ~ pdq(0,1,1)),
        'auto'     = ARIMA(Electricity, stepwise=FALSE))
report (fit)
```

```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 6 x 8
##   .model    sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 arima010 672392.   -1732. 3468. 3468. 3475. <cpl [0]> <cpl [4]>
## 2 arima110 550132.   -1710. 3431. 3431. 3447. <cpl [5]> <cpl [8]>
## 3 arima210 538128.   -1708. 3427. 3428. 3447. <cpl [6]> <cpl [8]>
## 4 arima011 517208.   -1704. 3419. 3419. 3436. <cpl [4]> <cpl [9]>
## 5 arima012 517208.   -1704. 3419. 3419. 3436. <cpl [4]> <cpl [9]>
## 6 auto     508179.   -1709. 3432. 3432. 3455. <cpl [5]> <cpl [9]>
```

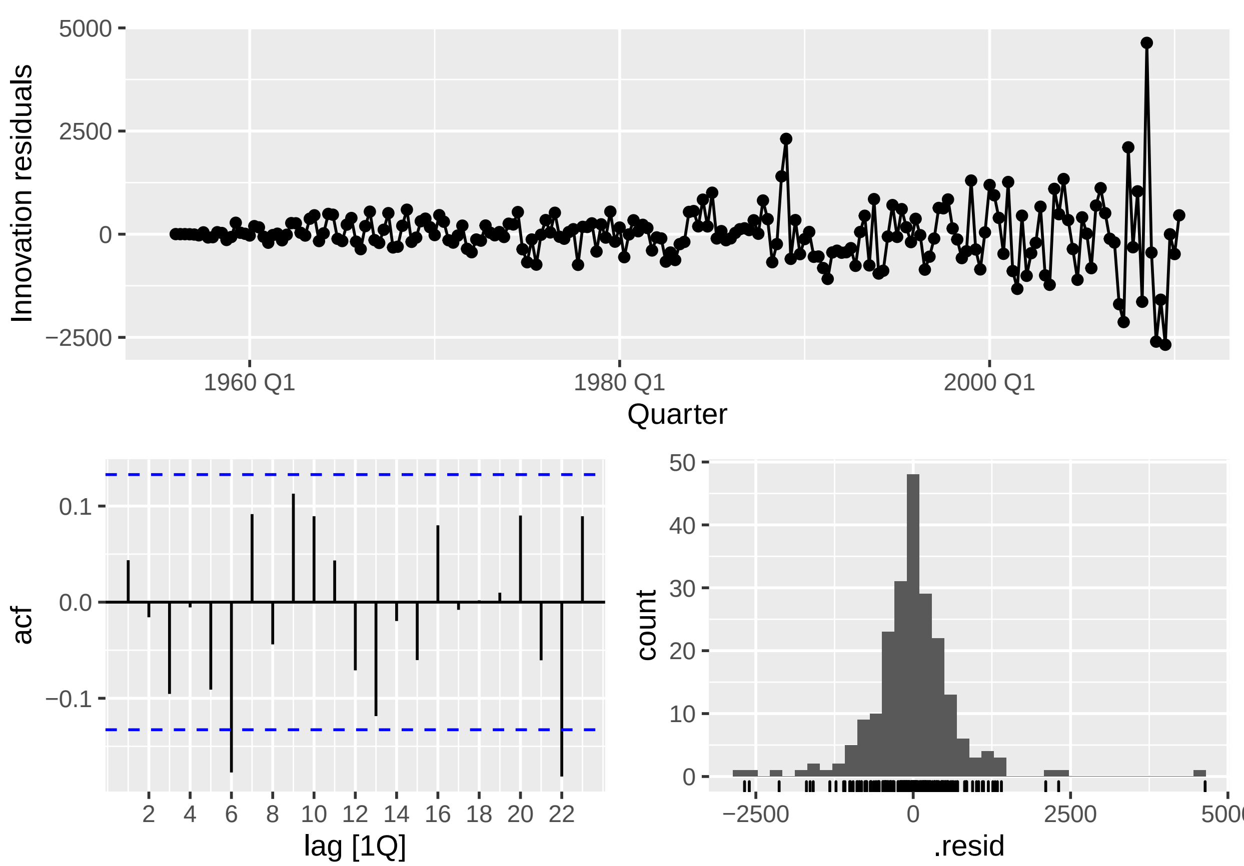
We can see that the best model is the arima011 and arima012 with the lowest AIC (3418.716) and AICc (3419)

d)

```r
fit <- ex11 %>%
  model('arima011' = ARIMA(Electricity ~ pdq(0,1,1)))

report(fit)
```

```
## Series: Electricity
## Model: ARIMA(0,1,1)(1,1,2)[4]
##
## Coefficients:
##           ma1    sar1     sma1    sma2
##       -0.5386  0.8528  -1.7176  0.7909
## s.e.   0.0722  0.1751   0.1709  0.1167
##
## sigma^2 estimated as 517208:  log likelihood=-1704.36
## AIC=3418.72   AICc=3419.01   BIC=3435.52
```

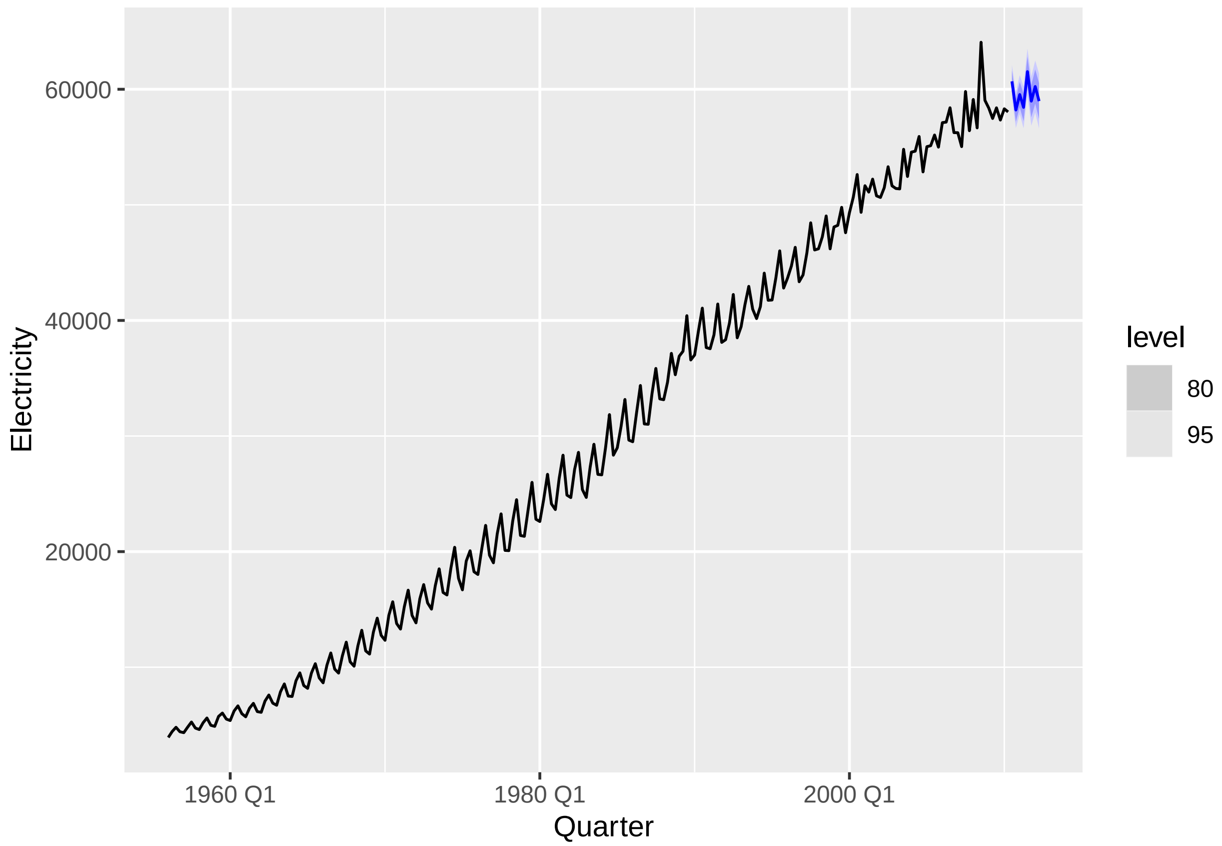
```r
fit %>% gg_tsresiduals()
```



We can see that the auto model is a (0,1,1)(1,1,2) ARIMA. If we look at the residuals, we can see that they follow a normal distribution, with most values around the 0 value although it looks like it have a long left tail, but there is a couples of autocorrelation at lag 6 and 22.

e)

```
fc <- fit %>%
  forecast(h = 8)

fc %>%
  autoplot(ex11)
```

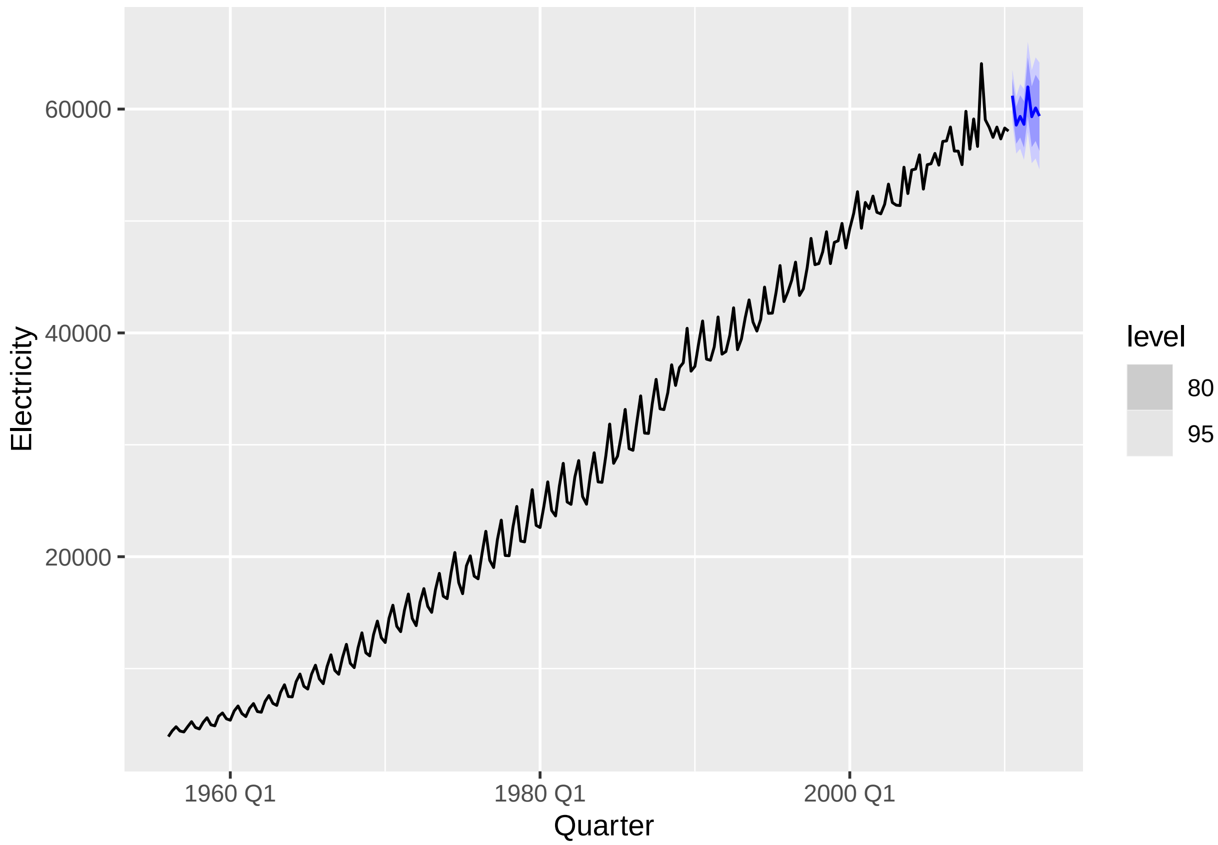


f)

```
fit <- ex11 %>% model(ETS(Electricity))
report(fit)
```

```
## Series: Electricity
## Model: ETS(M,A,M)
##   Smoothing parameters:
##     alpha = 0.5113977
##     beta  = 0.03629917
##     gamma = 0.3081703
##
##   Initial states:
##       l[0]    b[0]      s[0]     s[-1]     s[-2]      s[-3]
##  4150.421 107.077 0.9617614 1.082253 1.027738 0.9282474
##
##   sigma^2:  4e-04
##
##      AIC     AICc      BIC
## 3840.587 3841.453 3871.048
```

```
fc <- fit %>%
  forecast(h = 8)

fc %>%
  autoplot(ex11)
```



We can see that this model ETS, has a AIC of 3840 and an AICc of 3841, they are greater than the ARIMA model, meaning that the ARIMA is better

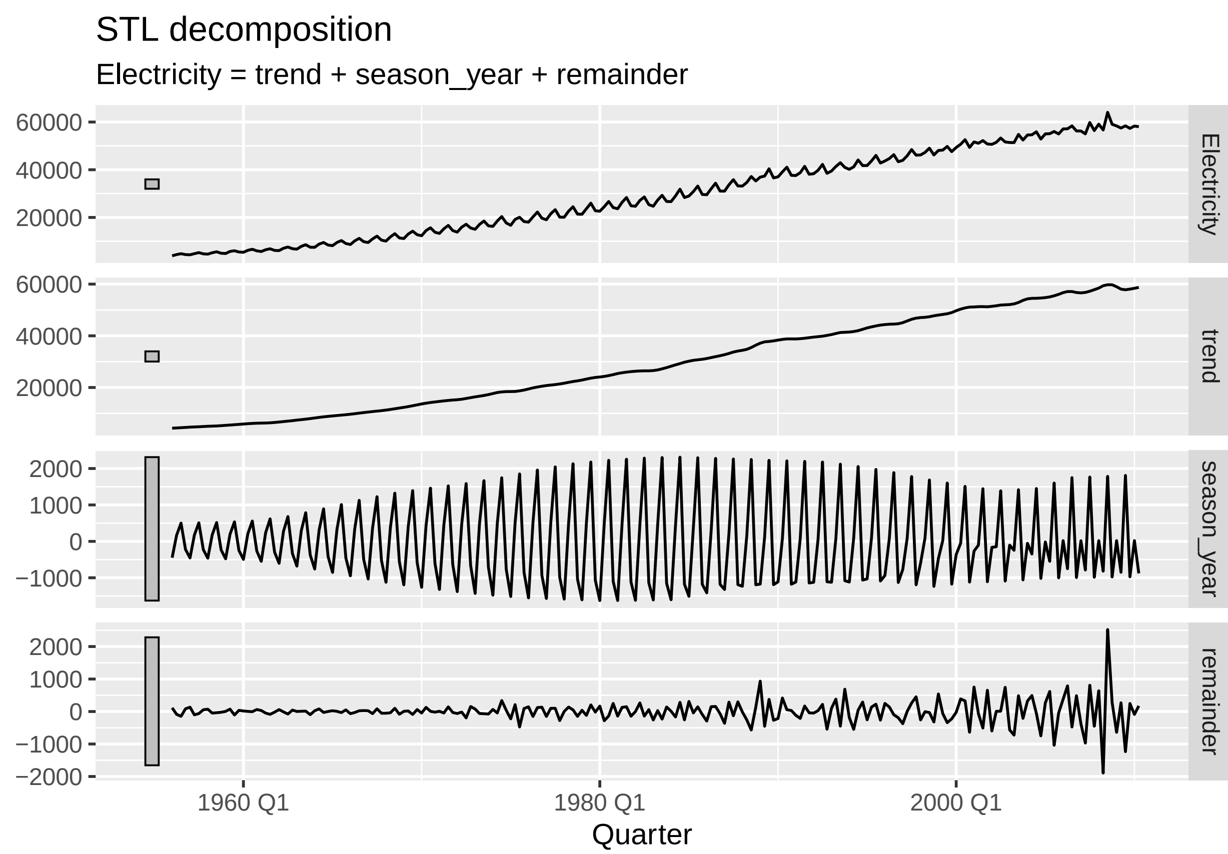
9.11 exercise 12

```
dcmp <- ex11 %>%
  model(stl = STL(Electricity))

components(dcmp)
```

```
## # A dable: 218 x 7 [1Q]
## # Key:     .model [1]
## # :        Electricity = trend + season_year + remainder
##    .model Quarter Electricity trend season_year remainder season_adjust
##    <chr>    <qtr>       <dbl> <dbl>       <dbl>     <dbl>         <dbl>
##  1 stl    1956 Q1        3923 4258.       -448.     114.          4371.
##  2 stl    1956 Q2        4436 4353.        169.     -86.1         4267.
```

```
##    3 stl     1956 Q3         4806 4452.        502.     -147.          4304.
##    4 stl     1956 Q4         4418 4556.       -222.       84.0         4640.
##    5 stl     1957 Q1         4339 4661.       -455.      134.          4794.
##    6 stl     1957 Q2         4811 4739.        175.     -102.          4636.
##    7 stl     1957 Q3         5259 4812.        509.      -62.3         4750.
##    8 stl     1957 Q4         4735 4908.       -228.       54.5         4963.
##    9 stl     1958 Q1         4608 4999.       -462.       71.5         5070.
##   10 stl     1958 Q2         5196 5063.        180.      -47.6         5016.
## # ... with 208 more rows
```

```
components(dcmp) %>% autoplot()
```




```
fit <- components(dcmp) %>%
  select(Quarter,season_adjust)%>%
  model('arima010' = ARIMA(season_adjust ~ pdq(0,1,0)),
        'arima110' = ARIMA(season_adjust ~ pdq(1,1,0)),
        'arima210' = ARIMA(season_adjust ~ pdq(2,1,0)),
        'arima011' = ARIMA(season_adjust ~ pdq(0,1,1)),
        'arima012' = ARIMA(season_adjust ~ pdq(0,1,1)),
        'auto'     = ARIMA(season_adjust, stepwise=FALSE))
report(fit)
```

```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 6 x 8
##   .model     sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>       <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 arima010 358541.  -1695. 3399. 3399. 3412. <cpl [0]>  <cpl [8]>
## 2 arima110 340423.  -1689. 3388. 3388. 3404. <cpl [1]>  <cpl [8]>
## 3 arima210 341464.  -1689. 3389. 3390. 3410. <cpl [2]>  <cpl [8]>
## 4 arima011 343584.  -1690. 3389. 3390. 3406. <cpl [0]>  <cpl [9]>
## 5 arima012 343584.  -1690. 3389. 3390. 3406. <cpl [0]>  <cpl [9]>
## 6 auto     308189.  -1679. 3373. 3374. 3400. <cpl [10]> <cpl [8]>
```

We can see now that the best model is the auto model with an AIC of 3373.029 and an AICc of 3373.72, this are better metrics than before, meaning that the new method is better

10.7 exercise 3

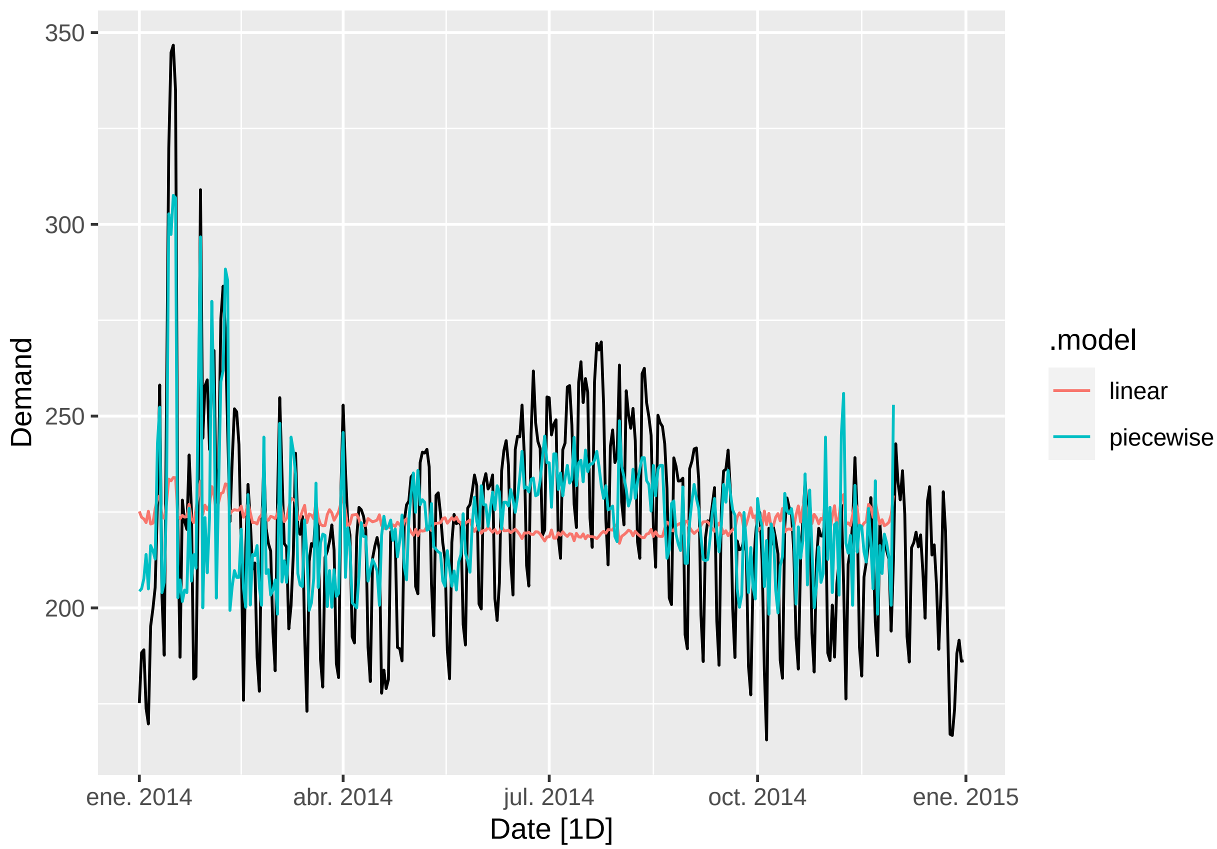
```
vic_elec_daily <- vic_elec %>%
  filter(year(Time) == 2014) %>%
  index_by(Date = date(Time)) %>%
  summarise(
    Demand = sum(Demand)/1e3,
    Temperature = max(Temperature),
    Holiday = any(Holiday)) %>%
  mutate(
    Temp2 = I(pmax(Temperature-25,0)),
    Day_Type = case_when(
      Holiday ~ "Holiday",
      wday(Date) %in% 2:6 ~ "Weekday",
      TRUE ~ "Weekend"))
vic_elec_daily
```

```
## # A tsibble: 365 x 6 [1D]
##    Date       Demand Temperature Holiday    Temp2 Day_Type
##    <date>      <dbl>       <dbl> <lgl>   <I<dbl>> <chr>
##  1 2014-01-01   175.        26   TRUE        1    Holiday
##  2 2014-01-02   188.        23   FALSE       0    Weekday
##  3 2014-01-03   189.        22.2 FALSE       0    Weekday
##  4 2014-01-04   174.        20.3 FALSE       0    Weekend
##  5 2014-01-05   170.        26.1 FALSE       1.10 Weekend
##  6 2014-01-06   195.        19.6 FALSE       0    Weekday
##  7 2014-01-07   200.        20   FALSE       0    Weekday
##  8 2014-01-08   205.        27.4 FALSE       2.4  Weekday
##  9 2014-01-09   227.        32.4 FALSE       7.4  Weekday
## 10 2014-01-10   258.        34   FALSE       9    Weekday
## # ... with 355 more rows
```

```
fit_trends <- vic_elec_daily %>%
  filter(Date < '2014-12-1  ') %>%
  model(
    linear = TSLM(Demand ~ Temperature ),
    piecewise = TSLM(Demand ~ Temperature + Temp2)
```

```
  )

fc_trends <- fit_trends %>% forecast(vic_elec_daily)


vic_elec_daily %>%
  autoplot(Demand,
    level = NULL) +
  geom_line(data = fitted(fit_trends),
            aes(y = .fitted, colour = .model))
```

```
## Warning: Ignoring unknown parameters: level
```



10.7 exercise 4

```
aus_accommodation
```

```
## # A tsibble: 592 x 5 [1Q]
## # Key:       State [8]
##        Date State                          Takings Occupancy   CPI
##       <qtr> <chr>                            <dbl>     <dbl> <dbl>
```

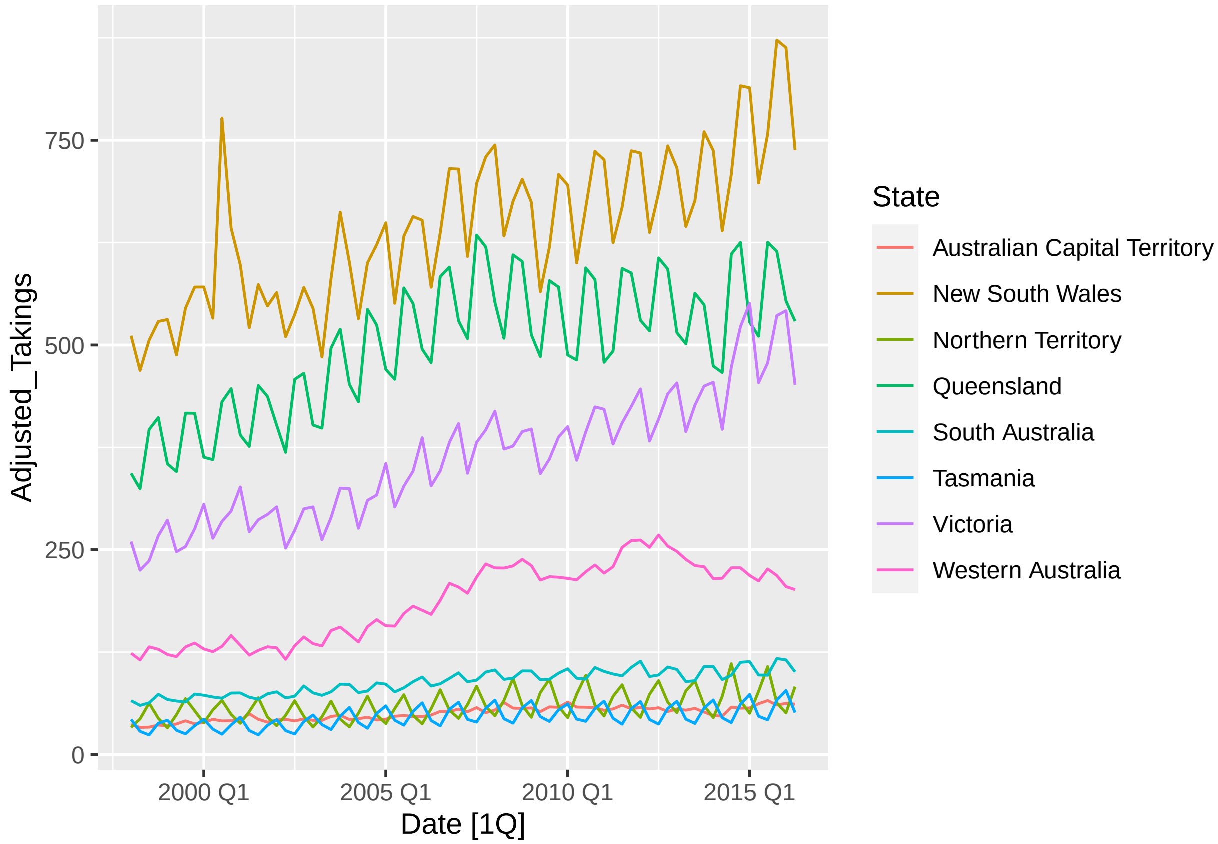
```
##   1 1998 Q1 Australian Capital Territory    24.3      65    67
##   2 1998 Q2 Australian Capital Territory    22.3      59    67.4
##   3 1998 Q3 Australian Capital Territory    22.5      58    67.5
##   4 1998 Q4 Australian Capital Territory    24.4      59    67.8
##   5 1999 Q1 Australian Capital Territory    23.7      58    67.8
##   6 1999 Q2 Australian Capital Territory    25.4      61    68.1
##   7 1999 Q3 Australian Capital Territory    28.2      66    68.7
##   8 1999 Q4 Australian Capital Territory    25.8      60    69.1
##   9 2000 Q1 Australian Capital Territory    27.3      60.9  69.7
## 10 2000 Q2 Australian Capital Territory    30.1      64.7  70.2
## # ... with 582 more rows
```

a)

```
aus_accommodation %>%
  mutate(Adjusted_Takings = Takings / CPI * 100) -> aus_accommodation_new

aus_accommodation_new
```

```
## # A tsibble: 592 x 6 [1Q]
## # Key:       State [8]
##       Date State                        Takings Occupancy   CPI Adjusted_Takings
##      <qtr> <chr>                          <dbl>     <dbl> <dbl>            <dbl>
##   1 1998 Q1 Australian Capital Territory    24.3      65    67               36.2
##   2 1998 Q2 Australian Capital Territory    22.3      59    67.4             33.1
##   3 1998 Q3 Australian Capital Territory    22.5      58    67.5             33.4
##   4 1998 Q4 Australian Capital Territory    24.4      59    67.8             36.0
##   5 1999 Q1 Australian Capital Territory    23.7      58    67.8             35.0
##   6 1999 Q2 Australian Capital Territory    25.4      61    68.1             37.3
##   7 1999 Q3 Australian Capital Territory    28.2      66    68.7             41.1
##   8 1999 Q4 Australian Capital Territory    25.8      60    69.1             37.4
##   9 2000 Q1 Australian Capital Territory    27.3      60.9  69.7             39.2
## 10 2000 Q2 Australian Capital Territory    30.1      64.7  70.2             42.9
## # ... with 582 more rows
```

```
aus_accommodation_new %>%
  autoplot(Adjusted_Takings)
```



b)

```
fit <- aus_accommodation_new %>%
  model(
    ARIMA(Adjusted_Takings ~ fourier(K=2) +
           trend(c(2008, 2009)) + pdq(1,1,0))
  )
```

```
## Warning: Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009)
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
## Provided exogenous regressors are rank deficient, removing regressors: `trend(c(2008, 2009))trend_18
```

```
report(fit)
```

```
## Warning in report.mdl_df(fit): Model reporting is only supported for individual
## models, so a glance will be shown. To see the report for a specific model, use
## `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 8 x 9
##   State                  .model sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
```

```
##   <chr>                  <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
## 1 Australian Capital ~ "ARIM~ 1.15e1   -190.  393.  394.  406. <cpl>    <cpl>
## 2 New South Wales      "ARIM~ 1.54e3   -369.  751.  753.  767. <cpl>    <cpl>
## 3 Northern Territory   "ARIM~ 2.49e1   -219.  455.  457.  473. <cpl>    <cpl>
## 4 Queensland           "ARIM~ 2.68e2   -305.  623.  625.  639. <cpl>    <cpl>
## 5 South Australia      "ARIM~ 1.19e1   -191.  398.  401.  417. <cpl>    <cpl>
## 6 Tasmania             "ARIM~ 5.12e0   -162.  338.  339.  354. <cpl>    <cpl>
## 7 Victoria             "ARIM~ 2.00e2   -294.  602.  604.  618. <cpl>    <cpl>
## 8 Western Australia    "ARIM~ 4.43e1   -239.  491.  492.  504. <cpl>    <cpl>
```

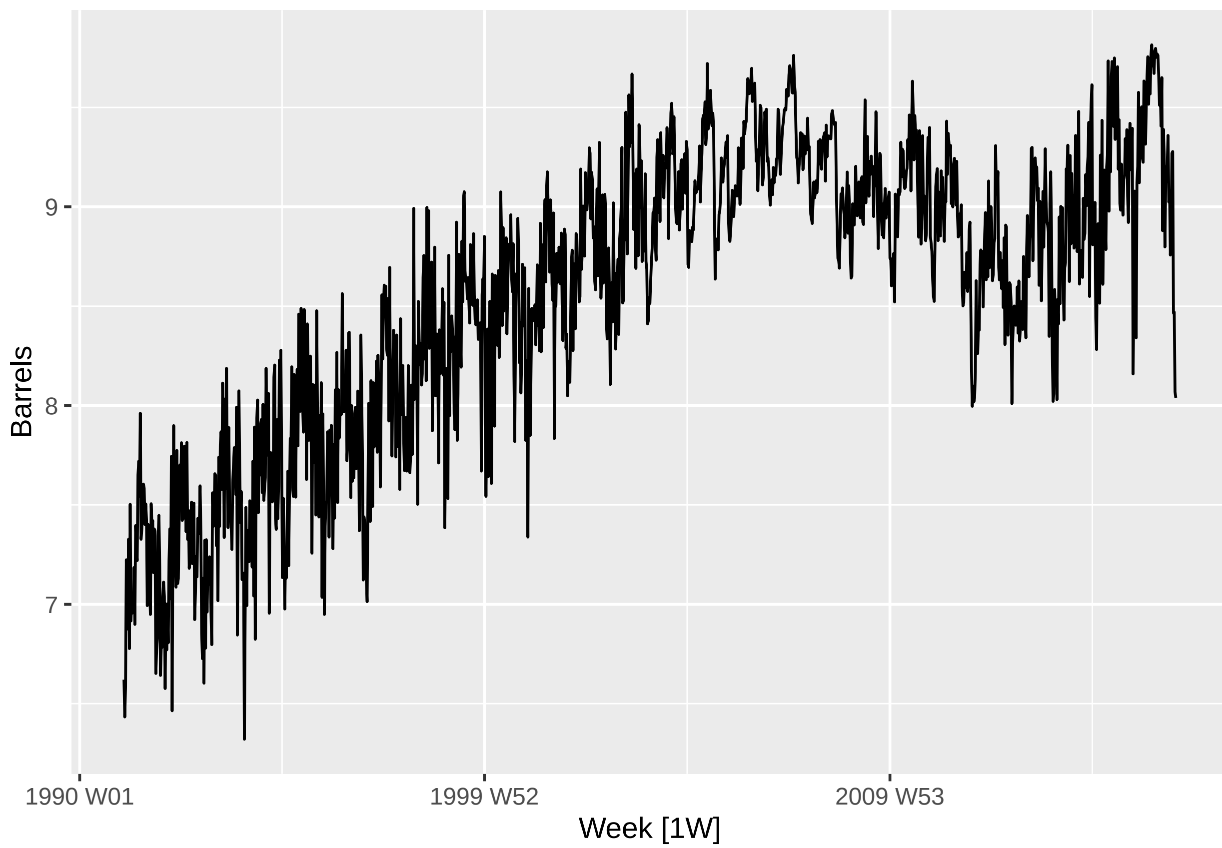
I don't know what error that is, or how to fix it. So, no idea what to do or how to continue

10.7 exercise 5

```
us_gasoline
```

```
## # A tsibble: 1,355 x 2 [1W]
##        Week Barrels
##      <week>   <dbl>
##  1 1991 W06    6.62
##  2 1991 W07    6.43
##  3 1991 W08    6.58
##  4 1991 W09    7.22
##  5 1991 W10    6.88
##  6 1991 W11    6.95
##  7 1991 W12    7.33
##  8 1991 W13    6.78
##  9 1991 W14    7.50
## 10 1991 W15    6.92
## # ... with 1,345 more rows
```

```
us_gasoline %>%
  autoplot(Barrels)
```



a)

```
fit <- us_gasoline %>%
  model(
    'model1_6-11' = TSLM(Barrels ~ fourier(K=1) + trend(c(2006, 2011)) ),
    'model1_7-12' = TSLM(Barrels ~ fourier(K=1) + trend(c(2007, 2012)) ),
    'model1_8-13' = TSLM(Barrels ~ fourier(K=1) + trend(c(2008, 2013)) ),
    'model2_6-11' = TSLM(Barrels ~ fourier(K=2) + trend(c(2006, 2011)) ),
    'model2_7-12' = TSLM(Barrels ~ fourier(K=2) + trend(c(2007, 2012)) ),
    'model2_8-13' = TSLM(Barrels ~ fourier(K=2) + trend(c(2008, 2013)) ),
    'model3_6-11' = TSLM(Barrels ~ fourier(K=3) + trend(c(2006, 2011)) ),
    'model3_7-12' = TSLM(Barrels ~ fourier(K=3) + trend(c(2007, 2012)) ),
    'model3_8-13' = TSLM(Barrels ~ fourier(K=3) + trend(c(2008, 2013)) )
  )

glance(fit)
```

```
## # A tibble: 9 x 15
##   .model    r_squared adj_r_squared sigma2 statistic p_value    df log_lik    AIC
##   <chr>         <dbl>         <dbl>  <dbl>     <dbl>   <dbl> <int>   <dbl>  <dbl>
## 1 model1_~      0.843         0.842 0.0843     1447.       0     6   -244. -3344.
## 2 model1_~      0.843         0.842 0.0844     1445.       0     6   -245. -3342.
## 3 model1_~      0.842         0.842 0.0845     1443.       0     6   -246. -3340.
## 4 model2_~      0.845         0.844 0.0832     1050.       0     8   -234. -3360.
## 5 model2_~      0.845         0.844 0.0833     1048.       0     8   -235. -3357.
## 6 model2_~      0.845         0.844 0.0834     1047.       0     8   -236. -3355.
```

```
## 7 model3_~     0.853         0.852 0.0790      869.       0    10   -198. -3428.
## 8 model3_~     0.853         0.852 0.0792      866.       0    10   -199. -3425.
## 9 model3_~     0.853         0.852 0.0793      864.       0    10   -201. -3422.
## # ... with 6 more variables: AICc <dbl>, BIC <dbl>, CV <dbl>, deviance <dbl>,
## #   df.residual <int>, rank <int>
```

We can see that the model with the lowest AIC and AICc is the model2_6-11 with 2 term Fourier and the knots between 2006-2011.

b)

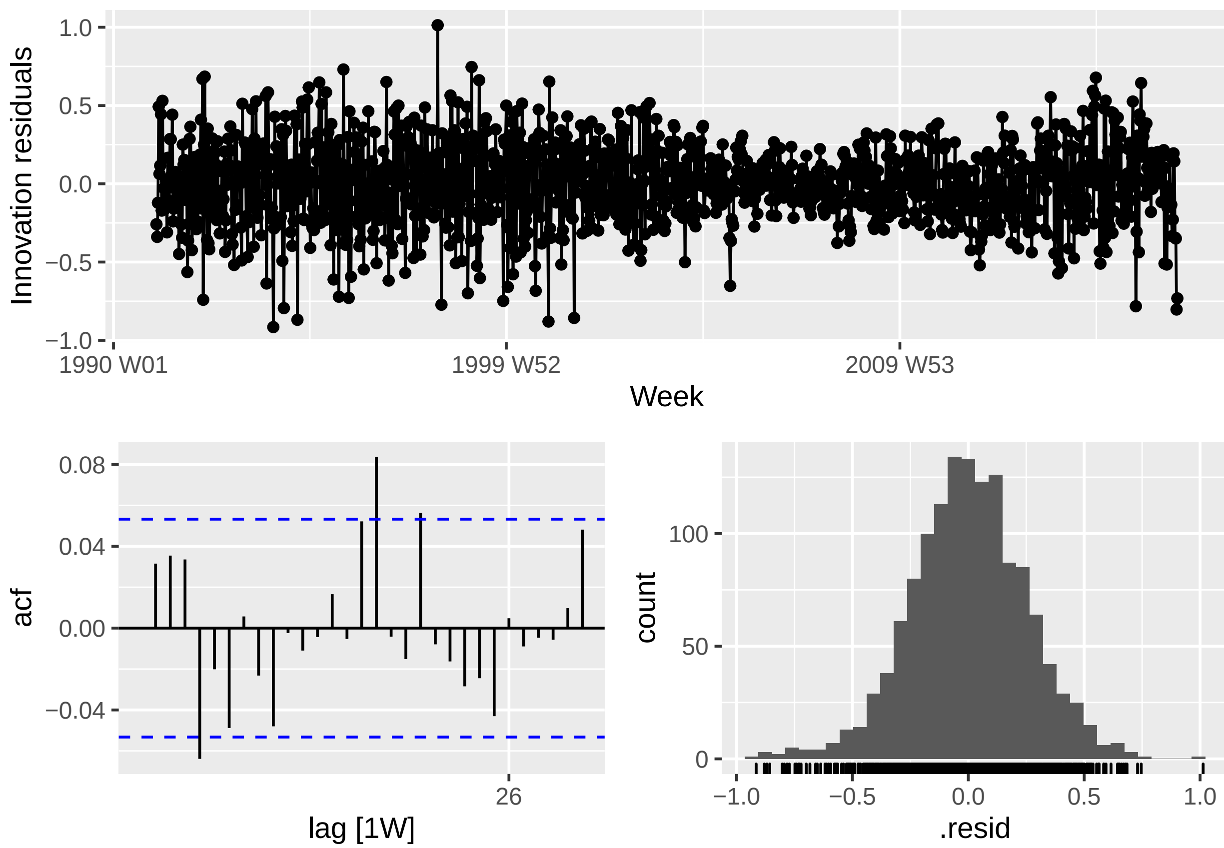
```
fit <- us_gasoline %>%
  model(
    'model2_6-11' = ARIMA(Barrels ~ fourier(K=2) + trend(c(2006, 2011)) ) # + pdq(1,1,0)
  )

report(fit)
```

```
## Series: Barrels
## Model: LM w/ ARIMA(1,0,2)(2,0,0)[52] errors
##
## Coefficients:
##          ar1      ma1      ma2    sar1    sar2  fourier(K = 2)C1_52
##       0.9802  -0.8589  -0.0169  0.1781  0.1284              -0.1127
## s.e.  0.0086   0.0289   0.0294  0.0280  0.0296               0.0193
##       fourier(K = 2)S1_52  fourier(K = 2)C2_52  fourier(K = 2)S2_52
##                   -0.2362               0.0413               0.0307
## s.e.               0.0192               0.0150               0.0150
##       trend(c(2006, 2011))trend  trend(c(2006, 2011))trend_906
##                          0.0025                        -0.0542
## s.e.                     0.0002                         0.0256
##       trend(c(2006, 2011))trend_911  intercept
##                              0.0514     7.1879
## s.e.                         0.0258     0.1260
##
## sigma^2 estimated as 0.06452:  log likelihood=-61.75
## AIC=151.49   AICc=151.8   BIC=224.45
```

We can see that the ARIMA model chosen was (1,0,2)(2,0,0) with an AIC of 151.48 and AICc of 151.8.

c)

```
fit %>% gg_tsresiduals()
```



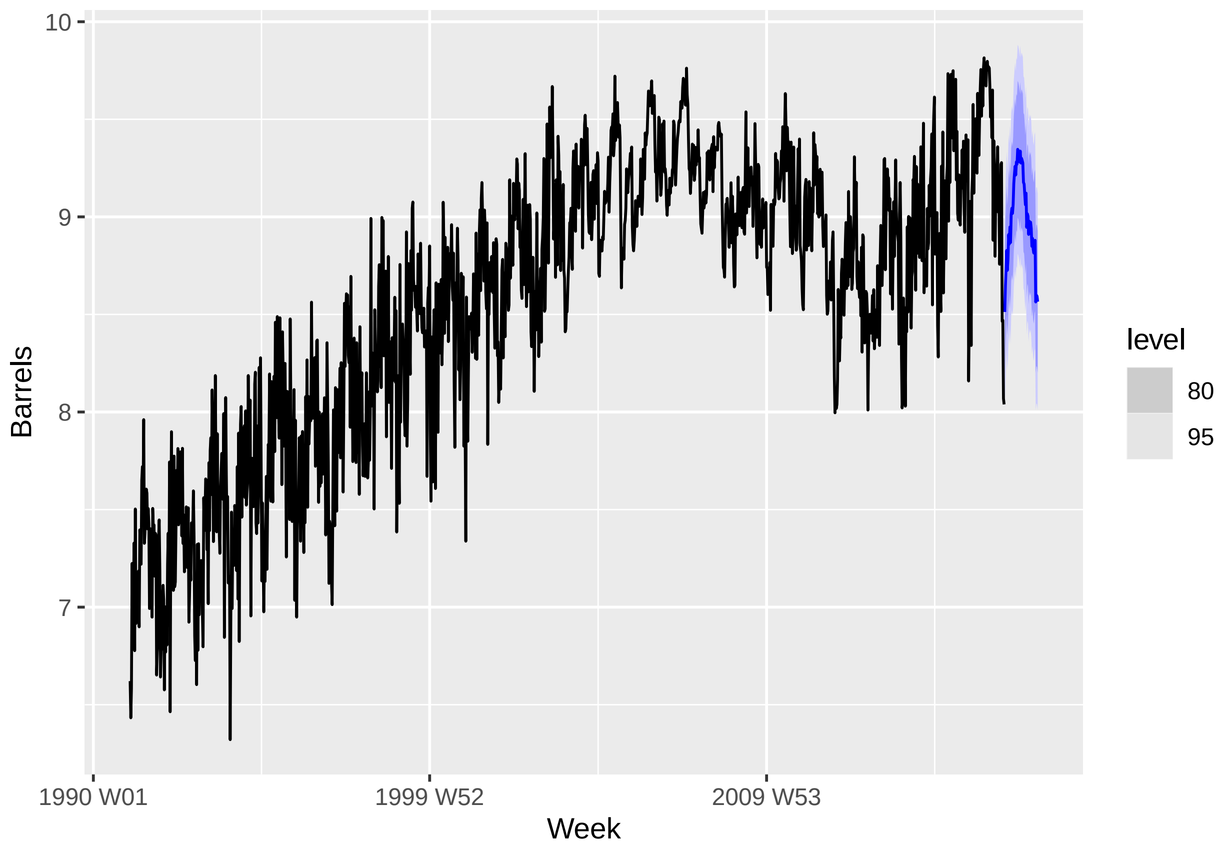

```
augment(fit) %>%
  features(.innov, ljung_box, lag=12, dof=4)
```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>         <dbl>     <dbl>
## 1 model2_6-11    18.1    0.0208
```

The model looks like it resembles white noise.

d)

```
fc <- fit %>%
  forecast(h=52)

fc %>%
  autoplot(us_gasoline)
```



10.7 exercise 6

a) The model is: (0,1,1)(2,1,0)

b) These two parameters' reflets the increase in the monthly total kilowatt-hour of electricity used when the monthly total heating degrees increase by 1 (B1) and the monthly total cooling degrees increase by 1(B2). So, the exact values that the fitted model reflets that: for each increase in 1 by the total heating degrees the total kilowatt-hour of electricity used increase by 0.0077; for each increase in 1 by monthly total cooling degrees the total kilowatt-hour of electricity used increase by 0.0208.

11.7 exercise 2

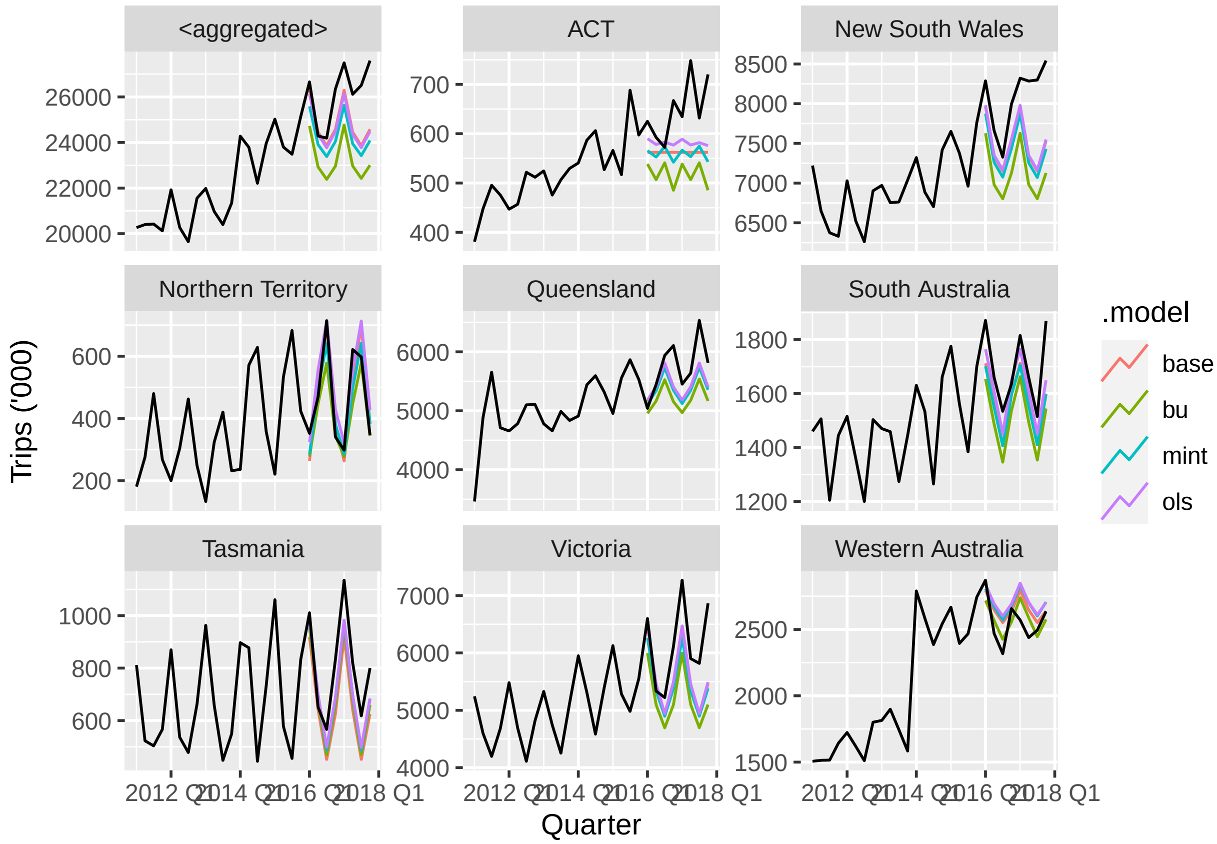
```
tourism_full <- tourism %>%
  aggregate_key((State/Region) * Purpose, Trips = sum(Trips))
tourism_full
```

```
## # A tsibble: 34,000 x 5 [1Q]
## # Key:       State, Purpose, Region [425]
##    Quarter State        Purpose      Region        Trips
##      <qtr> <chr*>       <chr*>       <chr*>        <dbl>
##  1 1998 Q1 <aggregated> <aggregated> <aggregated> 23182.
##  2 1998 Q2 <aggregated> <aggregated> <aggregated> 20323.
```

```
##  3 1998 Q3 <aggregated> <aggregated> <aggregated> 19827.
##  4 1998 Q4 <aggregated> <aggregated> <aggregated> 20830.
##  5 1999 Q1 <aggregated> <aggregated> <aggregated> 22087.
##  6 1999 Q2 <aggregated> <aggregated> <aggregated> 21458.
##  7 1999 Q3 <aggregated> <aggregated> <aggregated> 19914.
##  8 1999 Q4 <aggregated> <aggregated> <aggregated> 20028.
##  9 2000 Q1 <aggregated> <aggregated> <aggregated> 22339.
## 10 2000 Q2 <aggregated> <aggregated> <aggregated> 19941.
## # ... with 33,990 more rows
```

```
fit <- tourism_full %>%
  filter(year(Quarter) <= 2015) %>%
  model(base = ETS(Trips)) %>%
  reconcile(
    bu = bottom_up(base),
    ols = min_trace(base, method = "ols"),
    mint = min_trace(base, method = "mint_shrink"),
  )
```

```
fc <- fit %>% forecast(h = "2 years")
fc %>%
  filter(is_aggregated(Region), is_aggregated(Purpose)) %>%
  autoplot(
    tourism_full %>% filter(year(Quarter) >= 2011),
    level = NULL
  ) +
  labs(y = "Trips ('000)") +
  facet_wrap(vars(State), scales = "free_y")
```




```
fc %>%
  filter(is_aggregated(State), !is_aggregated(Purpose)) %>%
  autoplot(
    tourism_full %>% filter(year(Quarter) >= 2011),
    level = NULL
  ) +
  labs(y = "Trips ('000)") +
  facet_wrap(vars(Purpose), scales = "free_y")
```


```r
fc %>%
  filter(is_aggregated(State), is_aggregated(Purpose)) %>%
  accuracy(
    data = tourism_full,
    measures = list(rmse = RMSE, mase = MASE)
  ) %>%
  group_by(.model) %>%
  summarise(rmse = mean(rmse), mase = mean(mase))
```

```
## # A tibble: 4 x 3
##   .model  rmse  mase
##   <chr>  <dbl> <dbl>
## 1 base   1721.  1.53
## 2 bu     3070.  3.16
## 3 mint   2158.  2.09
## 4 ols    1804.  1.63
```

```r
fc %>%
  filter(is_aggregated(State), is_aggregated(Purpose)) %>%
  accuracy(tourism_full, list(skill = skill_score(CRPS))) %>%
  arrange(desc(skill))
```

```
## # A tibble: 4 x 6
##   .model State        Purpose      Region       .type   skill
##   <chr>  <chr*>       <chr*>       <chr*>       <chr>   <dbl>
```

```
## 1 base   <aggregated> <aggregated> <aggregated> Test   0.189
## 2 ols    <aggregated> <aggregated> <aggregated> Test   0.0779
## 3 mint   <aggregated> <aggregated> <aggregated> Test  -0.329
## 4 bu     <aggregated> <aggregated> <aggregated> Test  -1.24
```

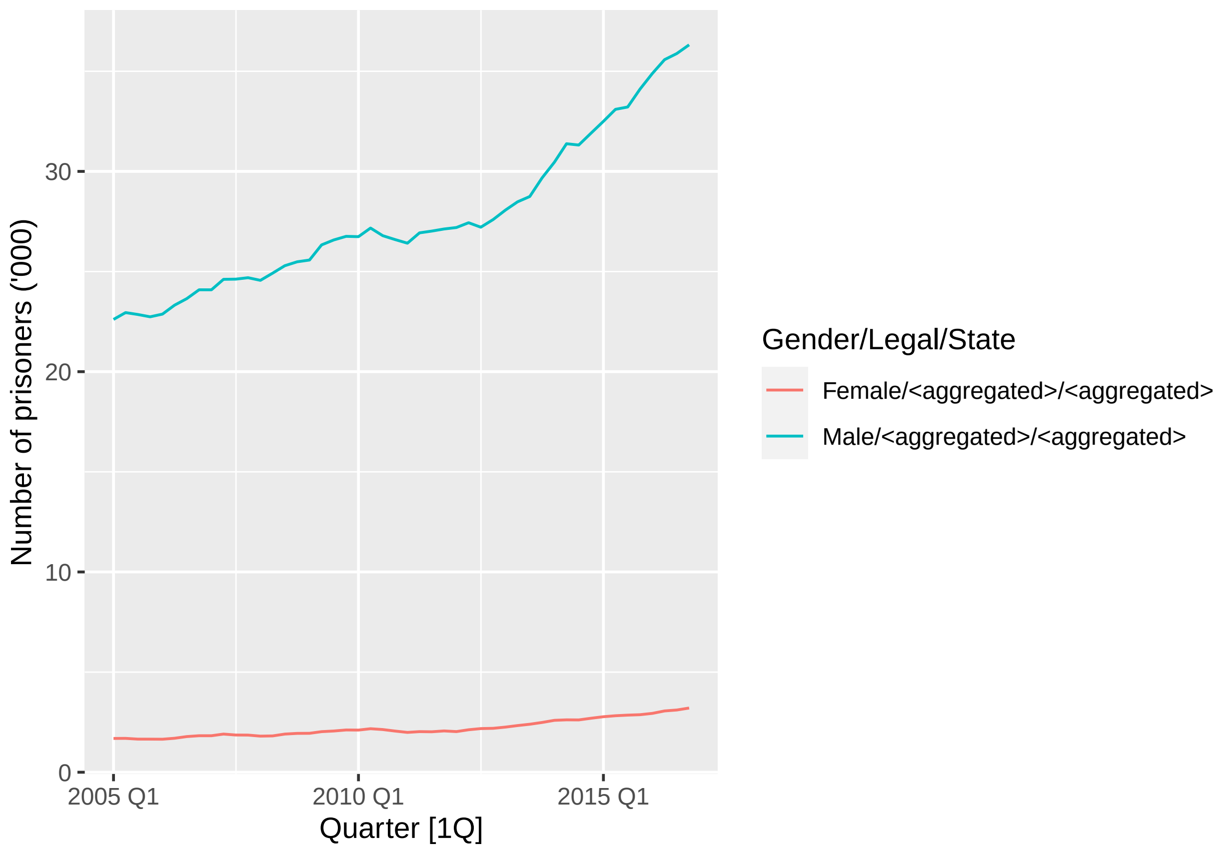
We can see that the base model achieve the best performance using this method, with a skill score of 0.1886.

11.7 exercise 3

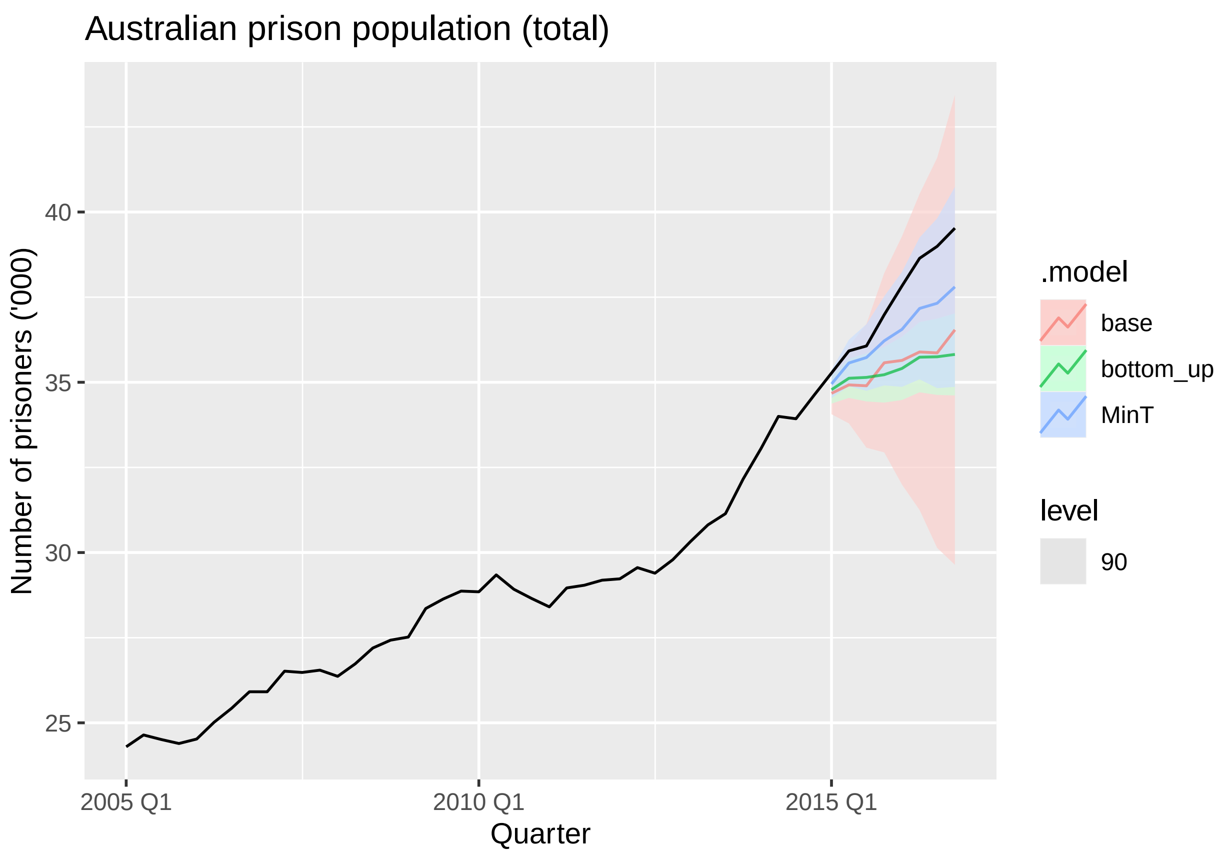
```
prison <- readr::read_csv("https://OTexts.com/fpp3/extrafiles/prison_population.csv") %>%
  mutate(Quarter = yearquarter(Date)) %>%
  select(-Date)  %>%
  as_tsibble(key = c(Gender, Legal, State, Indigenous),
             index = Quarter) %>%
  relocate(Quarter)
```

```
## Rows: 3072 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): State, Gender, Legal, Indigenous
## dbl  (1): Count
## date (1): Date
##
## 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.
```

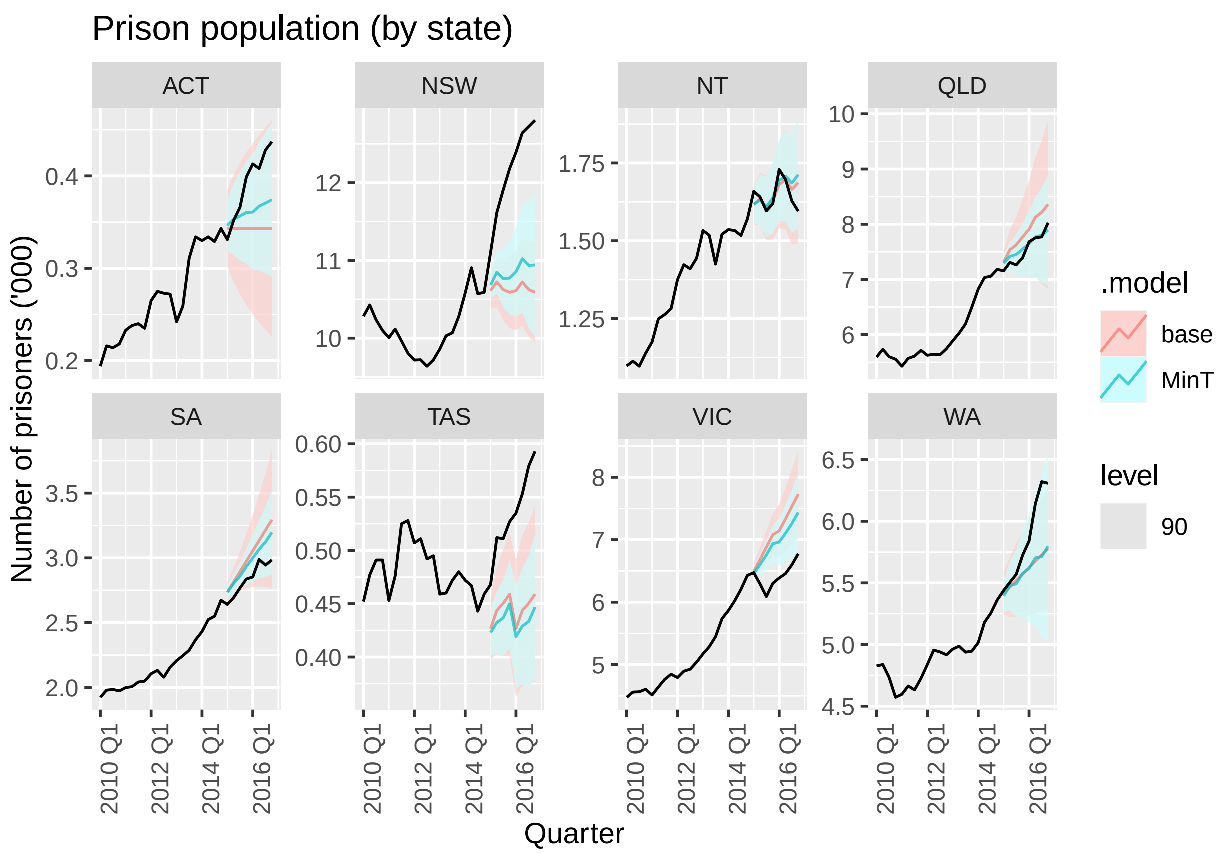
```
prison_gts <- prison %>%
  aggregate_key(Gender * Legal * State, Count = sum(Count)/1e3)

prison_gts %>%
  filter(!is_aggregated(Gender), is_aggregated(Legal),
         is_aggregated(State)) %>%
  autoplot(Count) +
  labs(y = "Number of prisoners ('000)")
```




```
fit <- prison_gts %>%
  filter(year(Quarter) <= 2014) %>%
  model(base = ETS(Count)) %>%
  reconcile(
    bottom_up = bottom_up(base),
    MinT = min_trace(base, method = "mint_shrink")
  )
fc <- fit %>% forecast(h = 8)
fc %>%
  filter(is_aggregated(State), is_aggregated(Gender),
         is_aggregated(Legal)) %>%
  autoplot(prison_gts, alpha = 0.7, level = 90) +
  labs(y = "Number of prisoners ('000)",
       title = "Australian prison population (total)")
```




```
fc %>%
  filter(
    .model %in% c("base", "MinT"),
    !is_aggregated(State), is_aggregated(Legal),
    is_aggregated(Gender)
  ) %>%
  autoplot(
    prison_gts %>% filter(year(Quarter) >= 2010),
    alpha = 0.7, level = 90
  ) +
  labs(title = "Prison population (by state)",
       y = "Number of prisoners ('000)") +
  facet_wrap(vars(State), scales = "free_y", ncol = 4) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
fc %>%
  filter(is_aggregated(State), is_aggregated(Gender),
         is_aggregated(Legal)) %>%
  accuracy(data = prison_gts,
           measures = list(mase = MASE,
                           ss = skill_score(CRPS)
                           )
           ) %>%
  group_by(.model) %>%
  summarise(mase = mean(mase), sspc = mean(ss) * 100)
```

```
## # A tibble: 3 x 3
##   .model     mase  sspc
##   <chr>     <dbl> <dbl>
## 1 base      1.72   55.9
## 2 bottom_up 1.84   33.5
## 3 MinT      0.895  76.8
```

```
fc <- fit %>% forecast(h = 8, bootstrap = TRUE)
fc %>%
  filter(is_aggregated(State), is_aggregated(Gender),
         is_aggregated(Legal)) %>%
  accuracy(data = prison_gts,
           measures = list(mase = MASE,
                           ss = skill_score(CRPS)
```

```
                           )
         ) %>%
  group_by(.model) %>%
  summarise(mase = mean(mase), sspc = mean(ss) * 100)
```

```
## # A tibble: 3 x 3
##   .model     mase  sspc
##   <chr>     <dbl> <dbl>
## 1 base      1.72   55.5
## 2 bottom_up 1.85   21.8
## 3 MinT      0.892  62.3
```

We can see that the new CRPS skill scores are higher for the base model, but lower on the other two model (bottom up and Mint)