Homework 5

Aaron Banlao

```
library(ggplot2)
library(fpp3)
```

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

```
## v tibble      3.1.8          v tsibble     1.1.3
## v dplyr       1.0.10         v tsibbledata 0.4.1.9000
## v tidyr       1.2.1          v feasts      0.3.0
## v lubridate   1.9.0          v fable       0.3.2
```

```
## -- 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(seasonal)
```

```
##
## Attaching package: 'seasonal'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
library(tsibbledata)
library(fable)
```

Section 7.10 Exercises 1

Half-hourly electricity demand for Victoria, Australia is contained in vic_elec. Extract the January 2014 electricity demand, and aggregate this data to daily with daily total demands and maximum temperatures.

Plot the data and find the regression model for Demand with temperature as a predictor variable. Why is there a positive relationship?

Produce a residual plot. Is the model adequate? Are there any outliers or influential observations?

Use the model to forecast the electricity demand that you would expect for the next day if the maximum temperature was
15C and compare it with the forecast if the with maximum temperature was 35C. Do you believe these forecasts?

Give prediction intervals for your forecasts.

Plot Demand vs Temperature for all of the available data in vic_elec aggregated to daily total demand and maximum temperature. What does this say about your model?

Answer:

a) There is a positive relationship because the coefficient of Temperature is positive. As the temperature goes up, the demand for using the air conditioning goes up as well.

b) It seems that there is no apparent pattern with the residuals meaning that the model is adequate.

c) The forecast for 15 was off but the temperature for 35 seemed to be following the trend of previous data.

e) The plot suggest that the model is not linear but rather exponential

Code and Comments:

```r
jan14_vic_elec <- vic_elec |>
  filter(yearmonth(Time) == yearmonth("2014 Jan")) |>
  index_by(Date = as_date(Time)) |>
  summarise(
    Demand = sum(Demand),
    Temperature = max(Temperature)
  )

head(jan14_vic_elec)
```

```
## # A tsibble: 6 x 3 [1D]
##   Date        Demand Temperature
##   <date>       <dbl>       <dbl>
## 1 2014-01-01 175185.        26
## 2 2014-01-02 188351.        23
## 3 2014-01-03 189086.        22.2
## 4 2014-01-04 173798.        20.3
## 5 2014-01-05 169733.        26.1
## 6 2014-01-06 195241.        19.6
```

a)

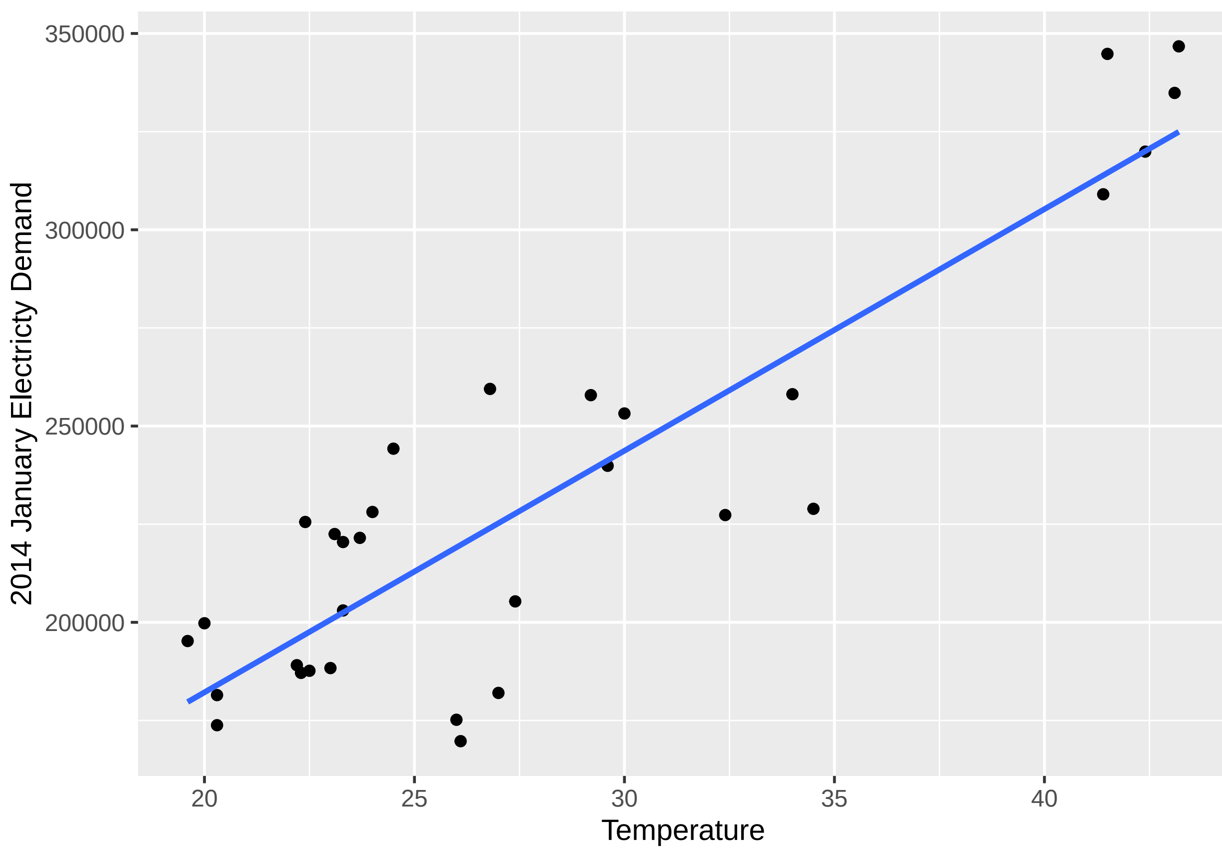
```r
fit <- jan14_vic_elec %>%
  model(tslm = TSLM(Demand ~ Temperature))

report(fit)
```

```
## Series: Demand
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49978.2 -10218.9   -121.3  18533.2  35440.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59083.9    17424.8   3.391  0.00203 **
## Temperature   6154.3      601.3  10.235 3.89e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24540 on 29 degrees of freedom
## Multiple R-squared: 0.7832,  Adjusted R-squared: 0.7757
## F-statistic: 104.7 on 1 and 29 DF, p-value: 3.8897e-11
```

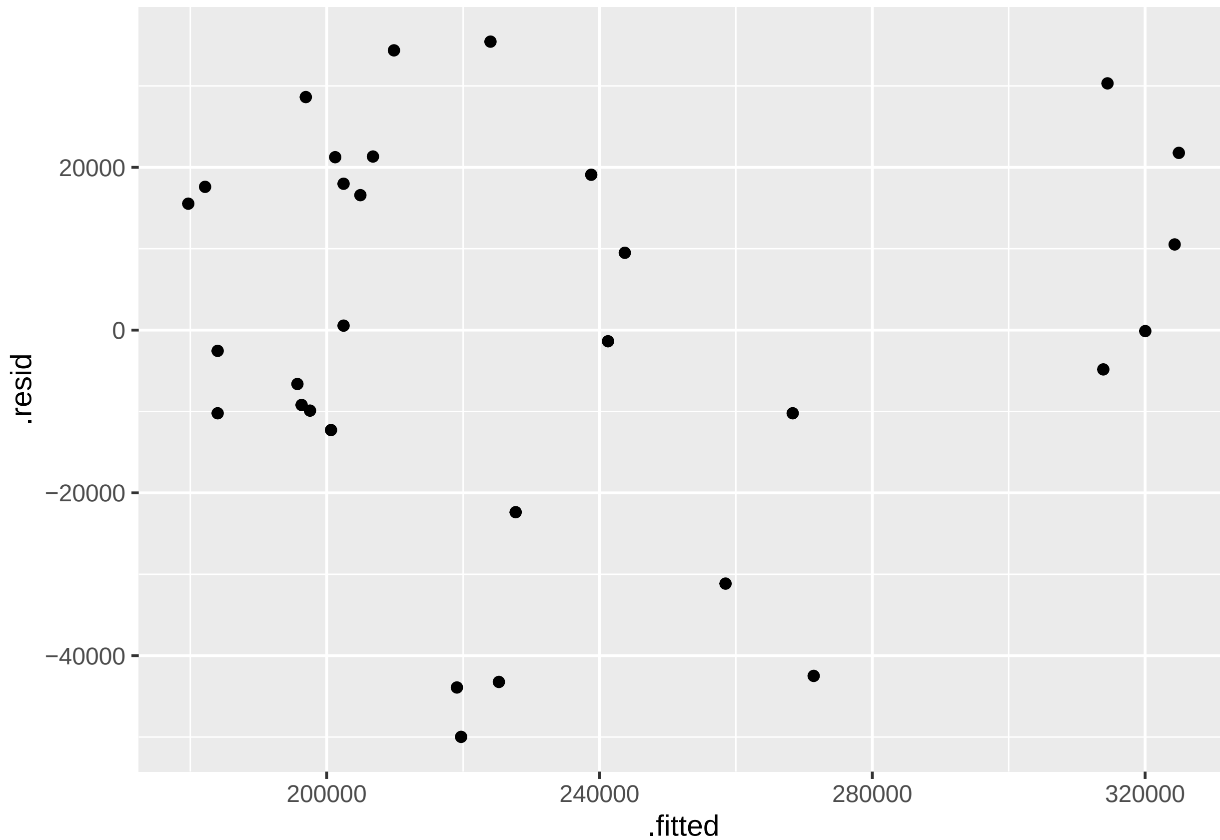
```
jan14_vic_elec %>%
  ggplot(aes(x = Temperature, y = Demand)) +
  labs(y = "2014 January Electricty Demand") +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



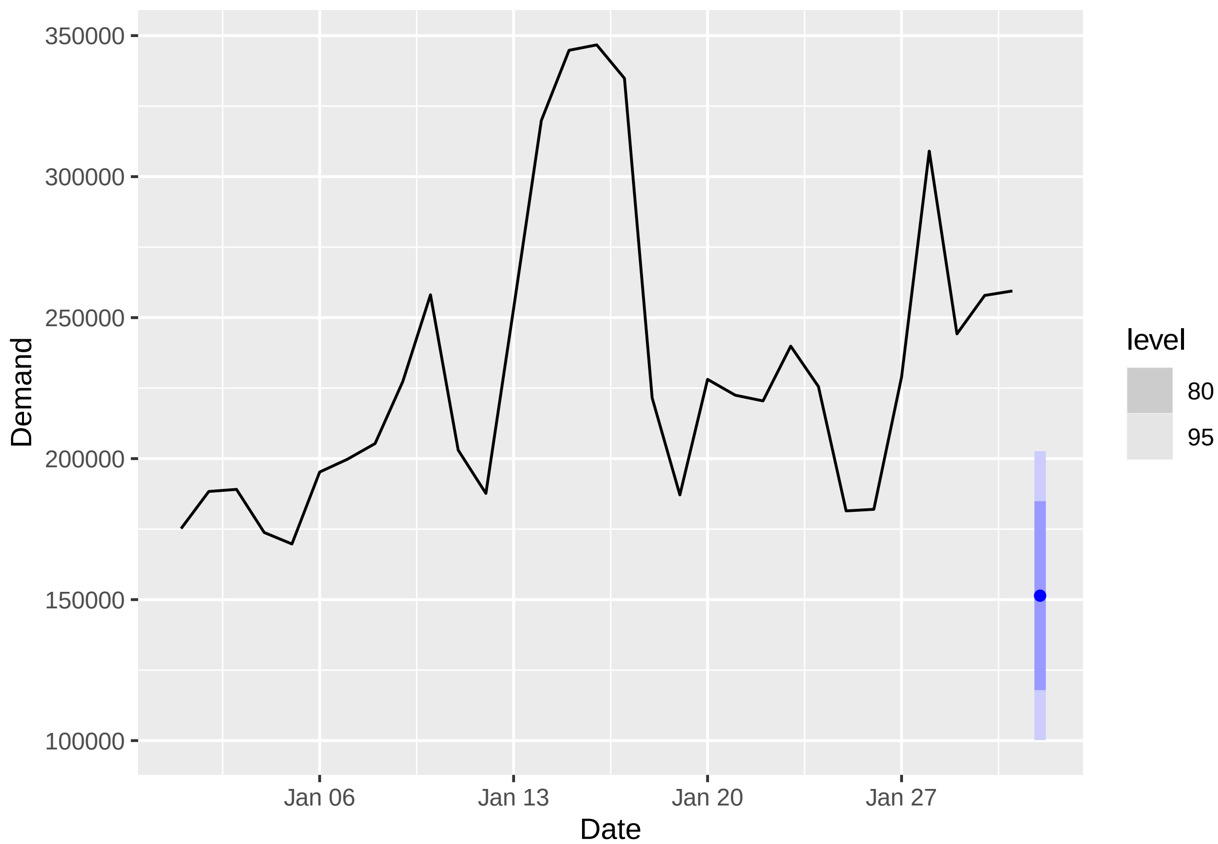
b)

```
augment(fit) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point()
```

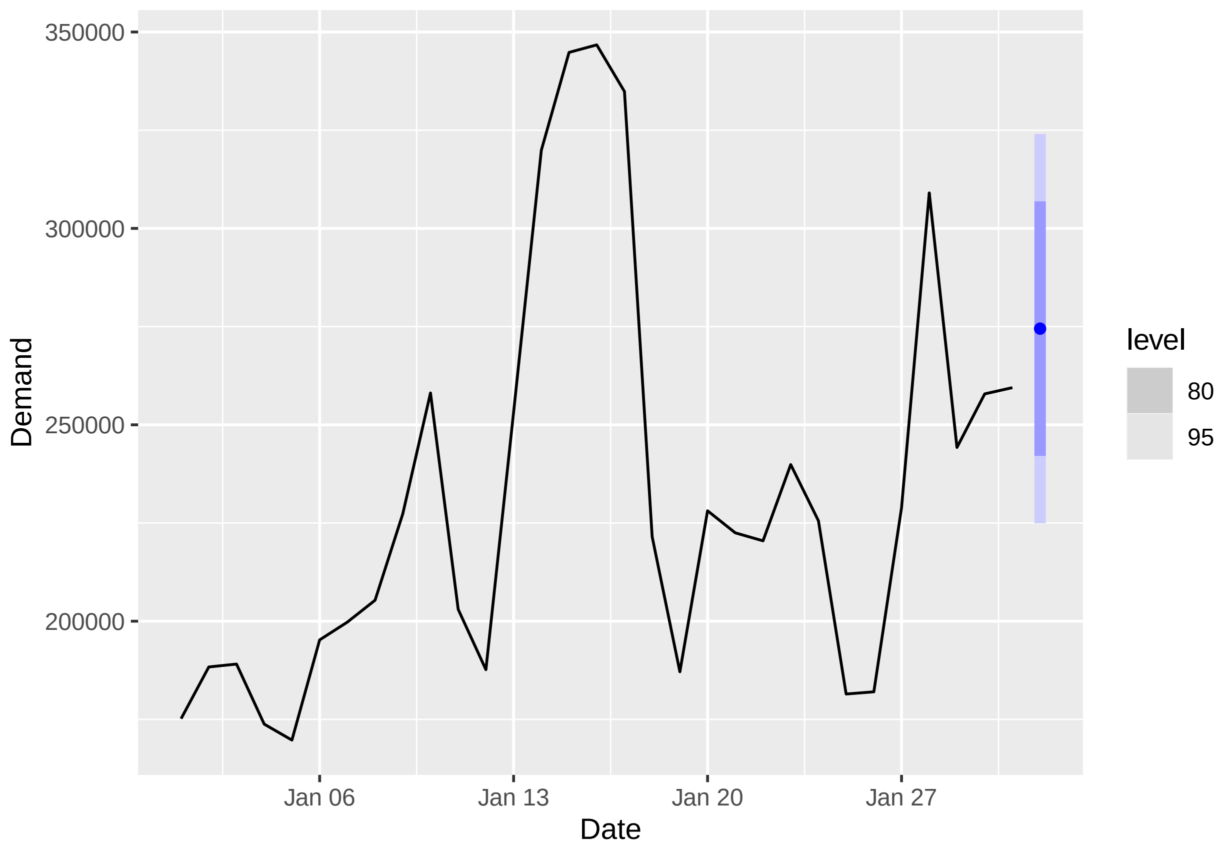


c)

```
jan14_vic_elec |>
  model(TSLM(Demand ~ Temperature)) |>
  forecast(
    new_data(jan14_vic_elec, 1) |>
      mutate(Temperature = 15)
  ) |>
  autoplot(jan14_vic_elec)
```




```
jan14_vic_elec |>
  model(TSLM(Demand ~ Temperature)) |>
  forecast(
    new_data(jan14_vic_elec, 1) |>
      mutate(Temperature = 35)
  ) |>
  autoplot(jan14_vic_elec)
```

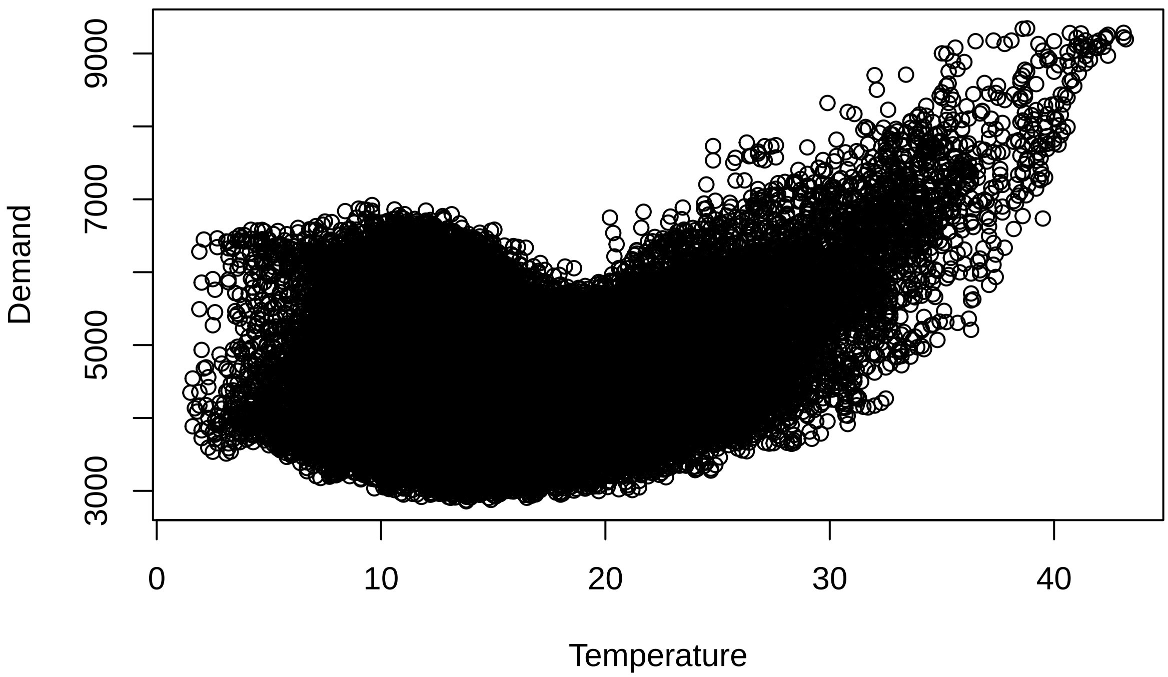



```
fit <- jan14_vic_elec %>%
  model(TSLM(Demand ~ Temperature))

pred <- scenarios(
  "15" = new_data(jan14_vic_elec, 2) %>%
    mutate(Temperature = 15),
  "35" = new_data(jan14_vic_elec, 2) %>%
    mutate(Temperature = 35)
)
```

d)

```
plot(Demand~Temperature, data = vic_elec)
```



e)

Section 7.10 Exercises 2

Data set olympic_running contains the winning times (in seconds) in each Olympic Games sprint, middle-distance and long-distance track events from 1896 to 2016.

Plot the winning time against the year for each event. Describe the main features of the plot. Fit a regression line to the data for each event. Obviously the winning times have been decreasing, but at what average rate per year? Plot the residuals against the year. What does this indicate about the suitability of the fitted lines? Predict the winning time for each race in the 2020 Olympics. Give a prediction interval for your forecasts. What assumptions have you made in these calculations?

Answer:

b) For the men's 100, the rate is decreasing at about 0.01 seconds. For the men's 200, the rate is decreasing at about 0.02 seconds. For the men's 400, the rate is decreasing at about 0.06 seconds. For the men's 800, the rate is decreasing at about 0.15 seconds. For the men's 1500, the rate is decreasing at about 0.31 seconds. For the men's 5000, the rate is decreasing at about 1.02 seconds. For the men's 10000, the rate is decreasing at about 1.03 seconds.

c) The residual plot shows that there is no apparent pattern and the model is fitted well.

Code and Comments:

```
data("olympic_running")
olympic_running <- olympic_running %>%
  filter(Sex == "men")
olympic_running %>%
  distinct(Length)
```

```
## # A tibble: 7 x 1
##   Length
##    <int>
## 1    100
## 2    200
## 3    400
## 4    800
## 5   1500
## 6   5000
## 7  10000
```

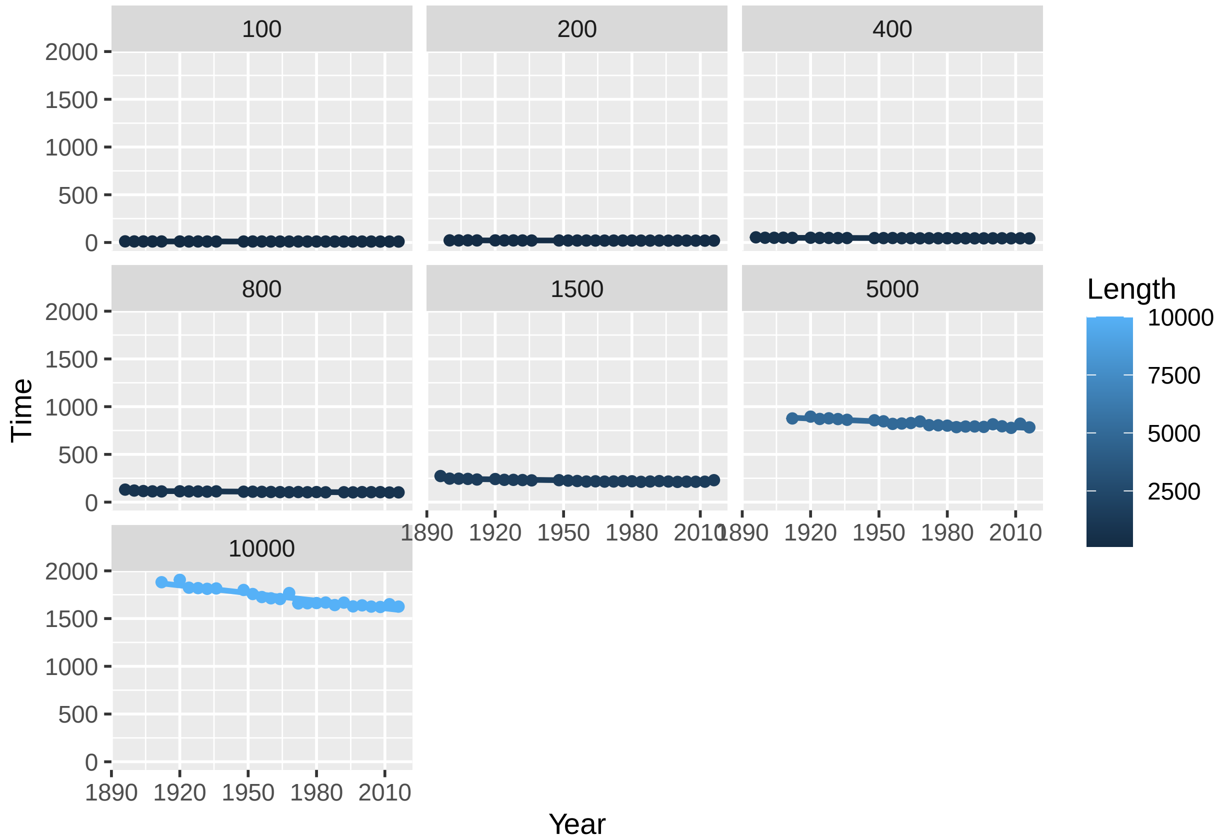
```
olympic_running %>%
  ggplot(aes(x = Year, y = Time, color = Length)) +
  geom_point() +
  facet_wrap(~Length) +
  geom_smooth(method = "lm", se = F)
```

a)

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 22 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 22 rows containing missing values (`geom_point()`).
```




```
one <- olympic_running %>%
  filter(Length == 100) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

b)

```
## Series: Time
## Model: TSLM
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.3443995 -0.1081360  0.0007715  0.0750701  0.9015537
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.011581   2.347437   14.91 2.94e-14 ***
## Year        -0.012612   0.001198  -10.52 7.24e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2321 on 26 degrees of freedom
## Multiple R-squared: 0.8099,   Adjusted R-squared: 0.8026
```

```
## F-statistic: 110.8 on 1 and 26 DF, p-value: 7.2403e-11
```

```r
two <- olympic_running %>%
  filter(Length == 200) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6112 -0.1872 -0.1046  0.1935  0.6959
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 69.376498   3.553448   19.52  < 2e-16 ***
## Year        -0.024881   0.001812  -13.73  3.8e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3315 on 25 degrees of freedom
## Multiple R-squared: 0.8829,  Adjusted R-squared: 0.8782
## F-statistic: 188.5 on 1 and 25 DF, p-value: 3.7956e-13
```

```r
four <- olympic_running %>%
  filter(Length == 400) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6001 -0.5747 -0.2858  0.5751  4.1505
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 172.481477  11.487522   15.02 2.52e-14 ***
## Year         -0.064574   0.005865  -11.01 2.75e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.136 on 26 degrees of freedom
## Multiple R-squared: 0.8234,  Adjusted R-squared: 0.8166
## F-statistic: 121.2 on 1 and 26 DF, p-value: 2.7524e-11
```

```r
eight <- olympic_running %>%
  filter(Length == 800) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7306 -1.8734 -0.8449  0.6851 12.9408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 405.8457    34.6506  11.712 1.21e-11 ***
## Year         -0.1518     0.0177  -8.576 6.47e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.387 on 25 degrees of freedom
## Multiple R-squared: 0.7463,  Adjusted R-squared: 0.7361
## F-statistic: 73.54 on 1 and 25 DF, p-value: 6.4705e-09
```

```
fift <- olympic_running %>%
  filter(Length == 1500) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.302  -4.585  -1.215   1.925  27.133
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 843.43666   81.81773  10.309 1.12e-10 ***
## Year         -0.31507    0.04177  -7.543 5.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.09 on 26 degrees of freedom
## Multiple R-squared: 0.6864,  Adjusted R-squared: 0.6743
## F-statistic:  56.9 on 1 and 26 DF, p-value: 5.2345e-08
```

```
five <- olympic_running %>%
  filter(Length == 5000) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.311 -11.668  -1.096   7.515  40.596
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2853.1995   205.1246  13.910 2.22e-12 ***
## Year          -1.0299     0.1042  -9.881 1.50e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.66 on 22 degrees of freedom
## Multiple R-squared: 0.8161,  Adjusted R-squared: 0.8078
## F-statistic: 97.64 on 1 and 22 DF, p-value: 1.4995e-09
```

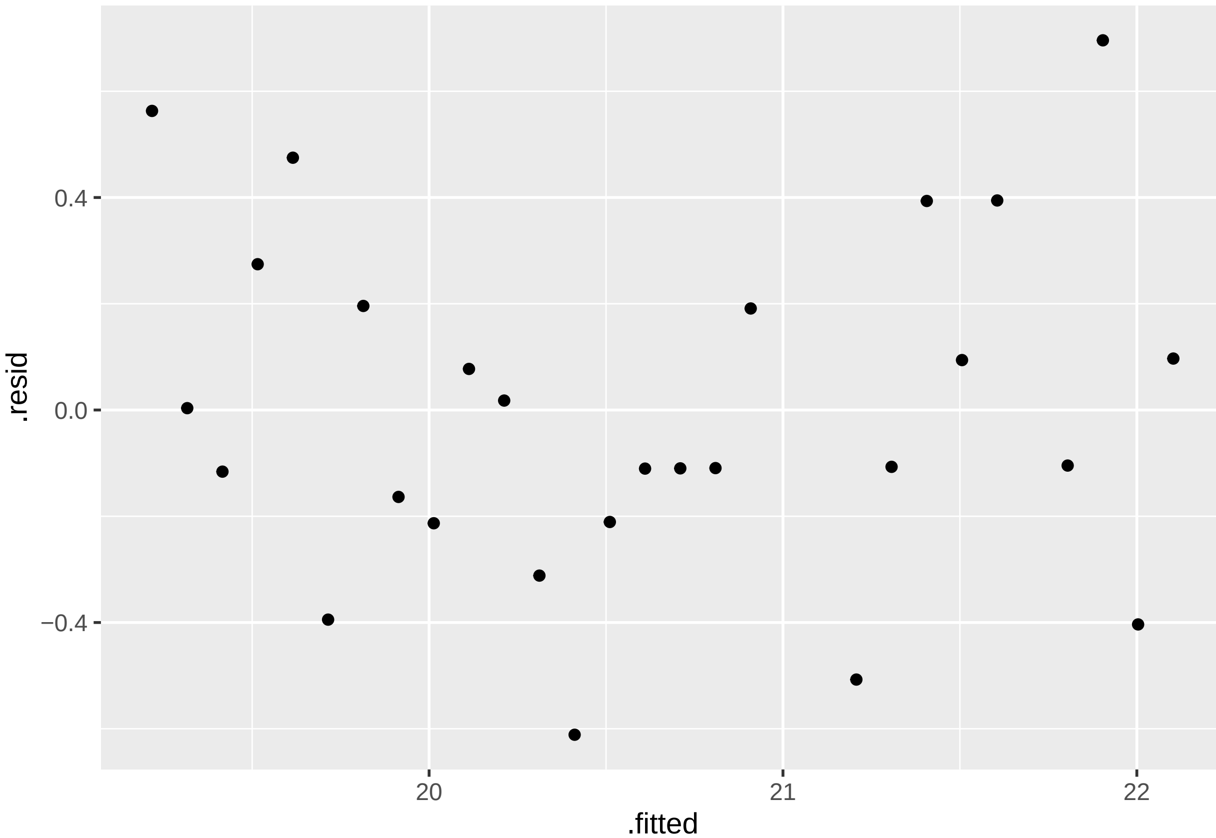
```
ten <- olympic_running %>%
  filter(Length == 10000) %>%
  model(TSLM(Time ~ Year)) %>%
  report()
```

```
## Series: Time
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.964 -24.222  -4.091  11.911  58.859
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6965.4594   378.4635   18.41 7.52e-15 ***
## Year          -2.6659     0.1923  -13.86 2.37e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.89 on 22 degrees of freedom
## Multiple R-squared: 0.8973,  Adjusted R-squared: 0.8926
## F-statistic: 192.2 on 1 and 22 DF, p-value: 2.3727e-12
```

```
augment(two) %>%
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point()
```

c)

```
## Warning: Removed 3 rows containing missing values (`geom_point()`).
```



Section 8.8 Exercises 5

Data set global_economy contains the annual Exports from many countries. Select one country to analyse.

Plot the Exports series and discuss the main features of the data. Use an ETS(A,N,N) model to forecast the series, and plot the forecasts. Compute the RMSE values for the training data. 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 data set. Compare the forecasts from both methods. Which do you think is best? Calculate a 95% prediction interval for the first forecast for each model, using the RMSE values and assuming normal errors. Compare your intervals with those produced using R.

Answer:

a) We see an upward overall trend with the United States exports. There are major dips, for example, in 1980, 2001, and 2008 when there were recessions that would affect the output of goods.

c) The RMSE value for the model is 0.6319877

d/e) The holt's method is a better method to use because the RSME value is smaller. With holt's we are able to caputure trending forecasts. SES is not the suitable method because there is a clear trend apparent with the plot.

Code and Comments:

```
head(global_economy)
```

```
## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country     Code   Year        GDP Growth   CPI Imports Exports Population
##   <fct>       <fct> <dbl>      <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02    4.13    8996351
## 2 Afghanistan AFG    1961  548888896.     NA    NA    8.10    4.45    9166764
## 3 Afghanistan AFG    1962  546666678.     NA    NA    9.35    4.88    9345868
## 4 Afghanistan AFG    1963  751111191.     NA    NA   16.9     9.17    9533954
## 5 Afghanistan AFG    1964  800000044.     NA    NA   18.1     8.89    9731361
## 6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4    11.3     9938414
```

```
global_economy %>%
  distinct(Country)
```

```
## # A tibble: 263 x 1
##    Country
##    <fct>
##  1 Afghanistan
##  2 Albania
##  3 Algeria
##  4 American Samoa
##  5 Andorra
##  6 Angola
##  7 Antigua and Barbuda
##  8 Arab World
##  9 Argentina
## 10 Armenia
## # ... with 253 more rows
```

```
us <- global_economy %>%
  filter(Country == "United States")
head(us)
```

```
## # A tsibble: 6 x 9 [1Y]
## # Key:       Country [1]
##   Country       Code   Year          GDP Growth   CPI Imports Exports Population
##   <fct>         <fct> <dbl>        <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
## 1 United States USA    1960 543300000000  NA     13.6    4.20    4.97  180671000
## 2 United States USA    1961 563300000000   2.30  13.7    4.03    4.90  183691000
## 3 United States USA    1962 605100000000   6.10  13.9    4.13    4.81  186538000
## 4 United States USA    1963 638600000000   4.40  14.0    4.09    4.87  189242000
## 5 United States USA    1964 685800000000   5.80  14.2    4.10    5.10  191889000
## 6 United States USA    1965 743700000000   6.40  14.4    4.24    4.99  194303000
```

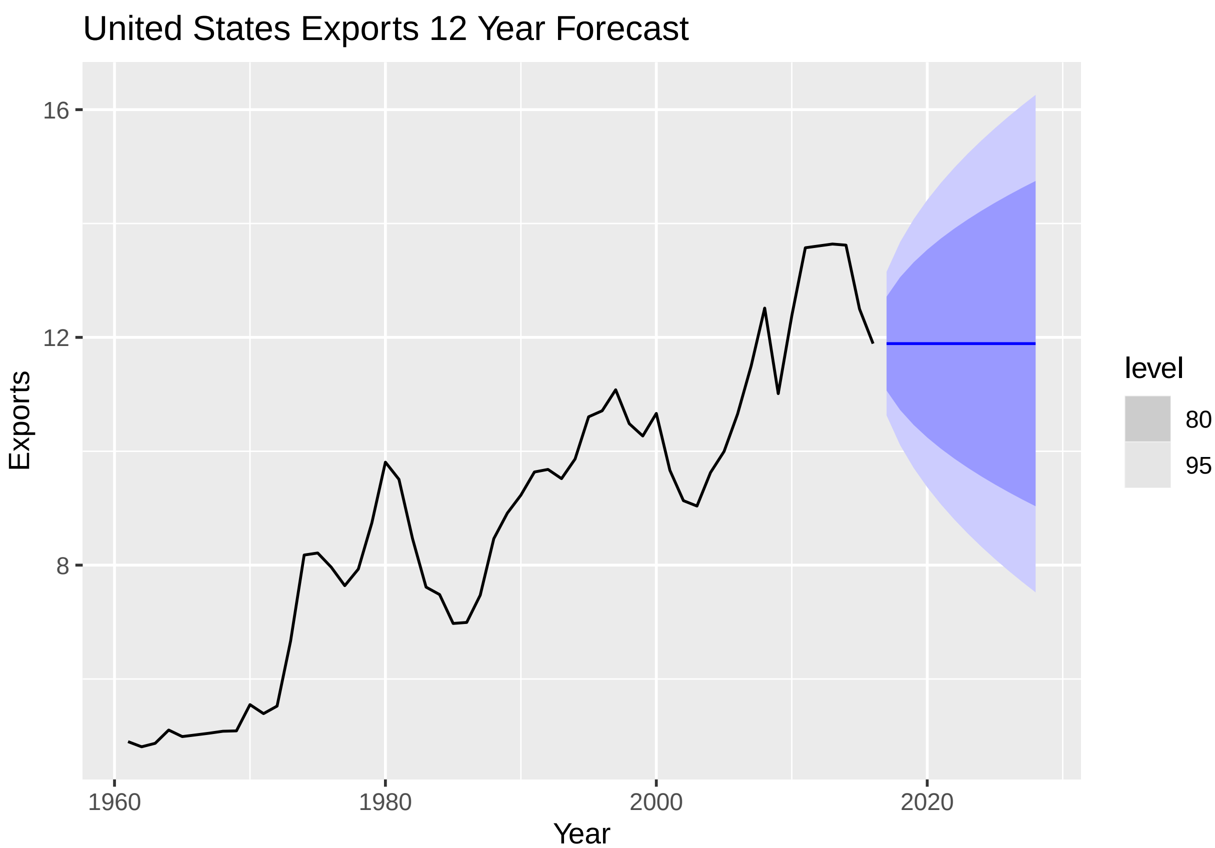
```
us <- us %>%
  drop_na()
```

```r
us %>%
  autoplot(Exports) +
  labs(title = "United States Exports")
```

a)

```r
fit1 <- us %>%
  model(ses = ETS(Exports ~ error("A") + trend("N") + season("N")))

fit1 %>%
  forecast(h = 12) %>%
  autoplot(us) +
  labs(title = "United States Exports 12 Year Forecast")
```



b)

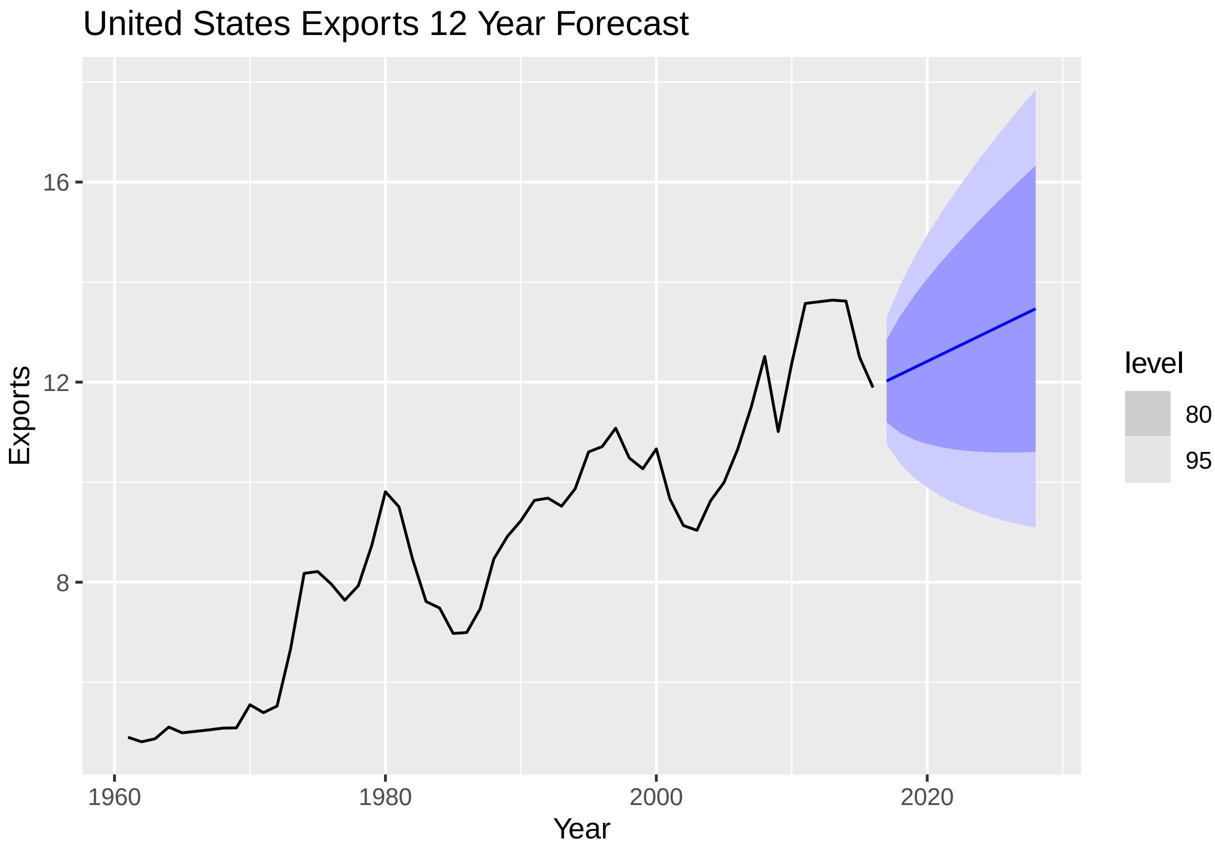
c)

```
accuracy(fit1)
```

```
## # A tibble: 1 x 11
##   Country       .model .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <fct>         <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 United States ses    Training 0.125 0.632 0.468  1.35  5.09 0.982 0.991 0.239
```

```
fit2 <- us %>%
  model(holts = ETS(Exports ~ error("A") + trend("A") + season("N")))

fit2 %>%
  forecast(h = 12) %>%
  autoplot(us) +
  labs(title = "United States Exports 12 Year Forecast")
```



d)

```
accuracy(fit2)
```

```
## # A tibble: 1 x 11
##   Country       .model .type        ME  RMSE   MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <fct>         <chr>  <chr>     <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 United States holts  Train~ 0.00170 0.621 0.476 -0.123  5.30 0.997 0.974 0.233
```

```
rmse <- accuracy(fit1) %>%
  pull(RMSE)
yhat <- forecast(fit1, h = 1) %>%
  pull(.mean)
```

```
yhat[1] + c(-1, 1) * qnorm(0.975) * rmse[1]
```

f)

```
## [1] 10.65201 13.12936
```

```r
rmse <- accuracy(fit2) %>%
  pull(RMSE)
yhat <- forecast(fit2, h = 1) %>%
  pull(.mean)
```

```r
yhat[1] + c(-1, 1) * qnorm(0.975) * rmse[1]
```

```
## [1] 10.80505 13.23907
```

```r
fit1 %>%
  forecast(h = 12) %>%
  mutate(interval = hilo(Exports, level = 95)) %>%
  unpack_hilo(interval)
```

```
## # A fable: 12 x 7 [1Y]
## # Key:     Country, .model [1]
##    Country       .model  Year     Exports .mean interval_lower interval_upper
##    <fct>         <chr>  <dbl>      <dist> <dbl>          <dbl>          <dbl>
##  1 United States ses     2017 N(12, 0.41)  11.9          10.6            13.2
##  2 United States ses     2018 N(12, 0.83)  11.9          10.1            13.7
##  3 United States ses     2019  N(12, 1.2)  11.9           9.71           14.1
##  4 United States ses     2020  N(12, 1.7)  11.9           9.37           14.4
##  5 United States ses     2021  N(12, 2.1)  11.9           9.07           14.7
##  6 United States ses     2022  N(12, 2.5)  11.9           8.80           15.0
##  7 United States ses     2023  N(12, 2.9)  11.9           8.55           15.2
##  8 United States ses     2024  N(12, 3.3)  11.9           8.32           15.5
##  9 United States ses     2025  N(12, 3.7)  11.9           8.11           15.7
## 10 United States ses     2026  N(12, 4.1)  11.9           7.90           15.9
## 11 United States ses     2027  N(12, 4.6)  11.9           7.71           16.1
## 12 United States ses     2028    N(12, 5)  11.9           7.52           16.3
```

```r
fit2 %>%
  forecast(h = 12) %>%
  mutate(interval = hilo(Exports, level = 95)) %>%
  unpack_hilo(interval)
```

```
## # A fable: 12 x 7 [1Y]
## # Key:     Country, .model [1]
##    Country       .model  Year     Exports .mean interval_lower interval_upper
##    <fct>         <chr>  <dbl>      <dist> <dbl>          <dbl>          <dbl>
##  1 United States holts   2017 N(12, 0.42)  12.0          10.8            13.3
##  2 United States holts   2018 N(12, 0.83)  12.2          10.4            13.9
##  3 United States holts   2019  N(12, 1.2)  12.3          10.1            14.5
##  4 United States holts   2020  N(12, 1.7)  12.4           9.89           14.9
##  5 United States holts   2021  N(13, 2.1)  12.5           9.72           15.4
##  6 United States holts   2022  N(13, 2.5)  12.7           9.58           15.8
##  7 United States holts   2023  N(13, 2.9)  12.8           9.47           16.2
##  8 United States holts   2024  N(13, 3.3)  12.9           9.37           16.5
##  9 United States holts   2025  N(13, 3.7)  13.1           9.28           16.9
## 10 United States holts   2026  N(13, 4.2)  13.2           9.21           17.2
## 11 United States holts   2027  N(13, 4.6)  13.3           9.15           17.5
## 12 United States holts   2028    N(13, 5)  13.5           9.09           17.8
```

Section 8.8 Exercises 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?

Answer:

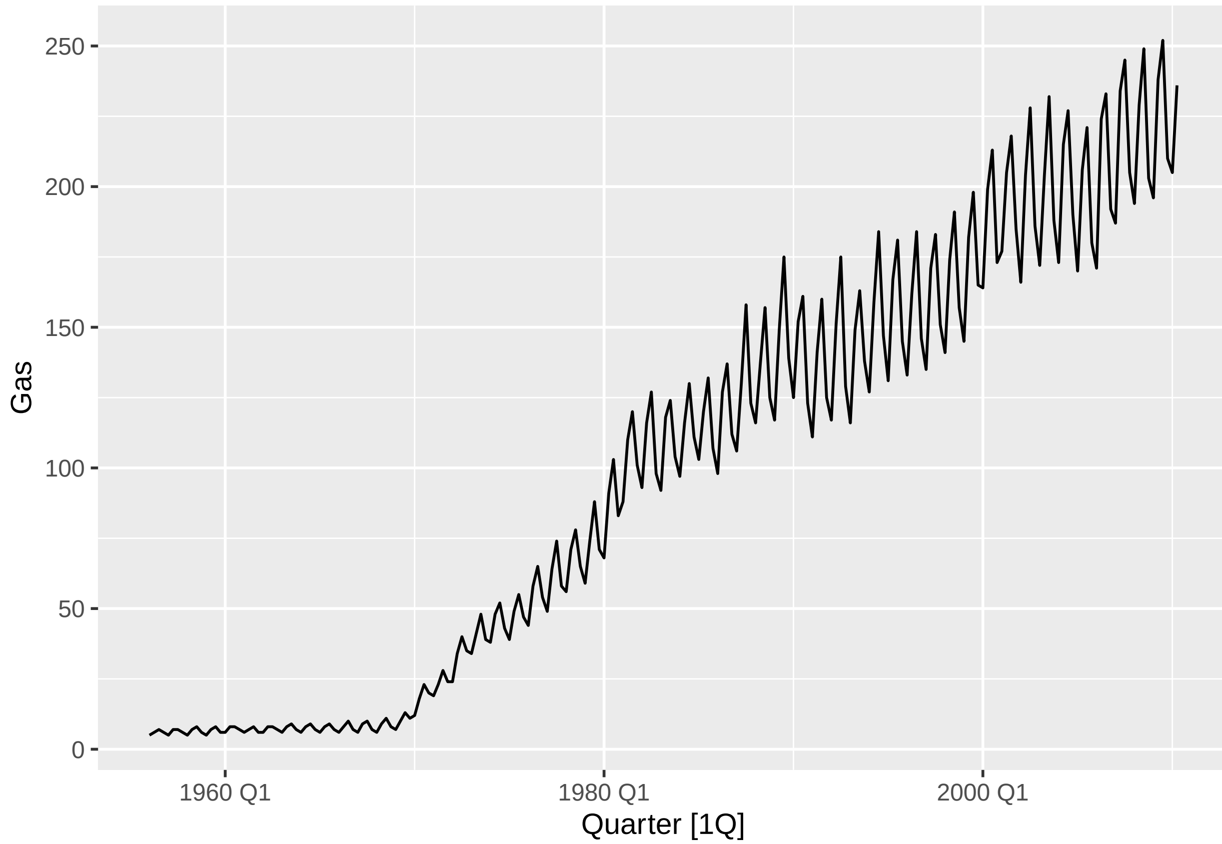
The multiplicative method is preferred because the seasonal variations are changing proportional to the level of the series. The damped model, compared to the multiplicative model, is a better method because the RMSE is lower.

Code and Comments:

```
head(aus_production)
```

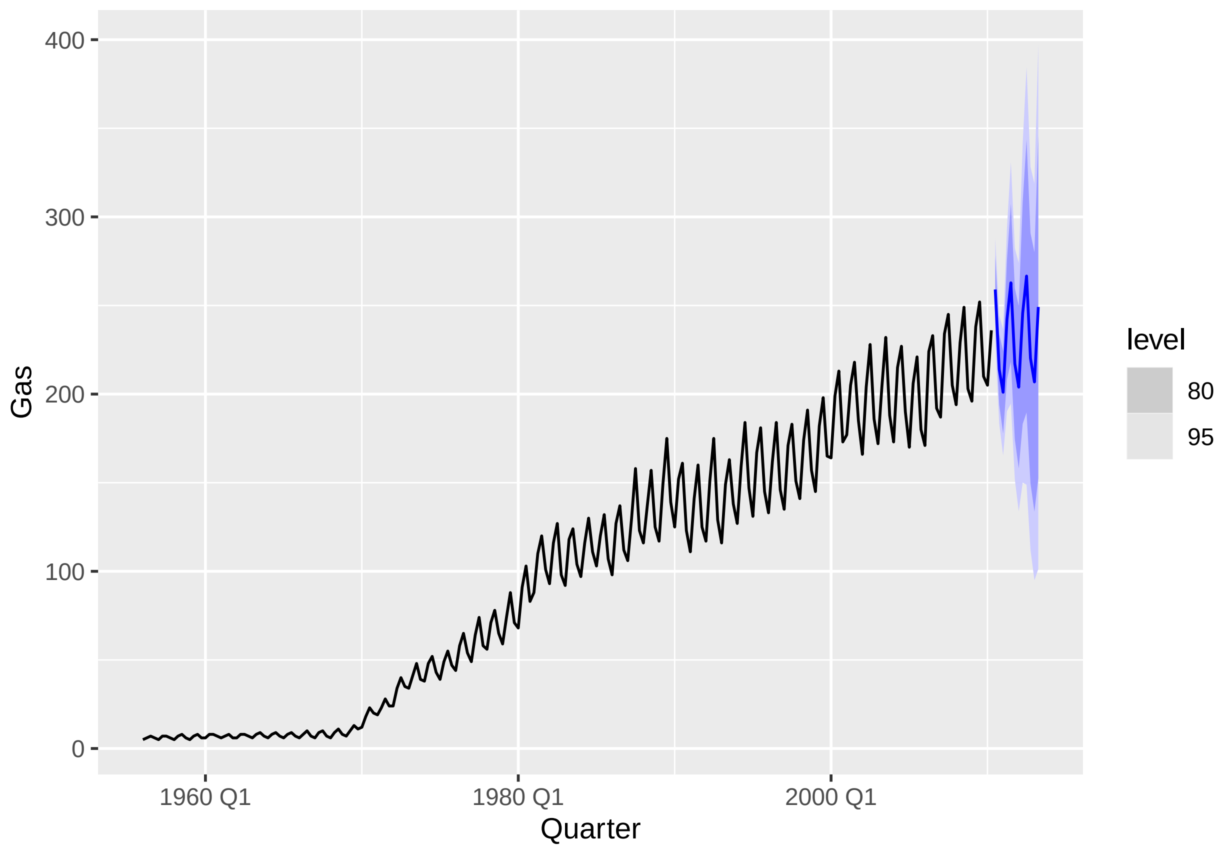
```
## # A tsibble: 6 x 7 [1Q]
##   Quarter  Beer Tobacco Bricks Cement Electricity   Gas
##     <qtr> <dbl>   <dbl>  <dbl>  <dbl>       <dbl> <dbl>
## 1 1956 Q1   284    5225    189    465        3923     5
## 2 1956 Q2   213    5178    204    532        4436     6
## 3 1956 Q3   227    5297    208    561        4806     7
## 4 1956 Q4   308    5681    197    570        4418     6
## 5 1957 Q1   262    5577    187    529        4339     5
## 6 1957 Q2   228    5651    214    604        4811     7
```

```
aus_production %>%
  autoplot(Gas)
```



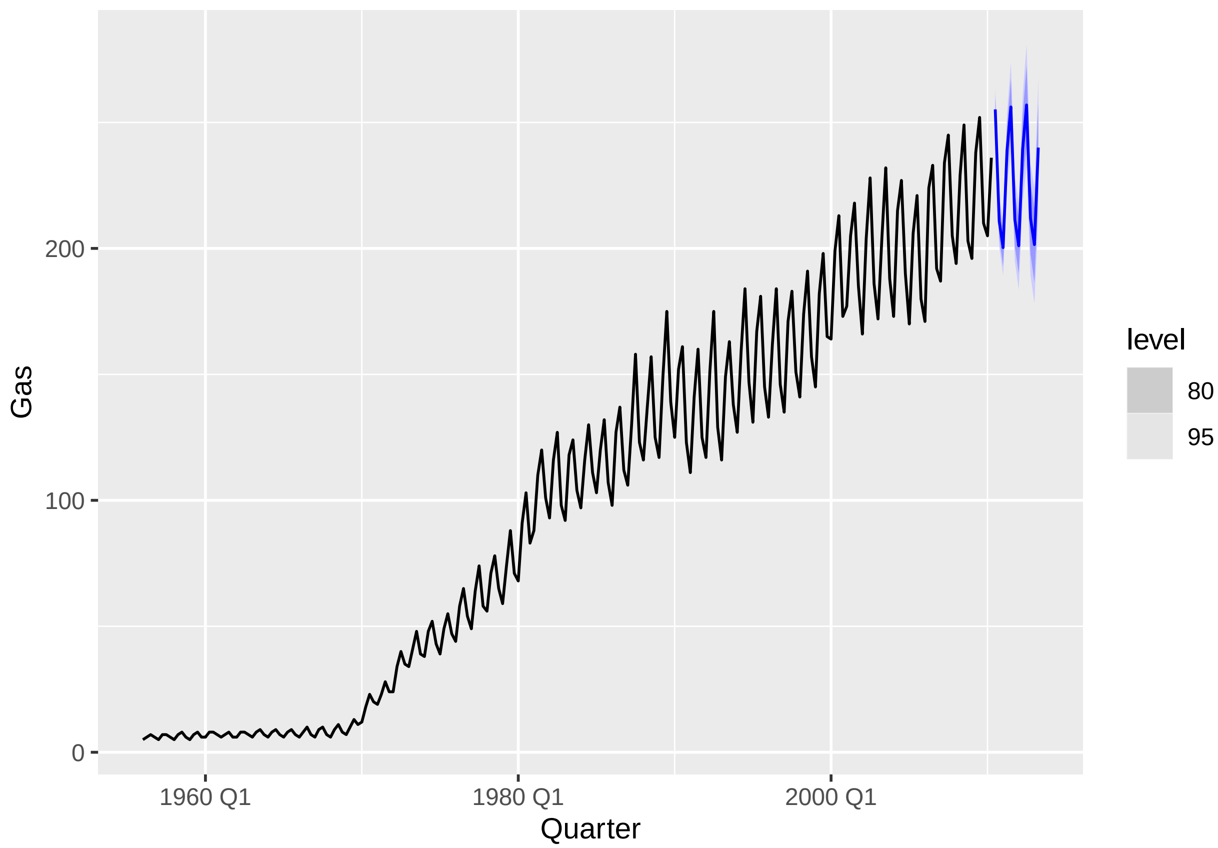

```
fit <- aus_production %>%
  model(multiplicative = ETS(Gas ~ error("M") + trend("A") + season("M")))

fit %>%
  forecast(h = 12) %>%
  autoplot(aus_production)
```




```
fit2 <- aus_production %>%
  model(damped = ETS(Gas ~ error("A") + trend("Ad", phi = 0.9) + season("M")))

fit2 %>%
  forecast(h = 12) %>%
  autoplot(aus_production)
```




```
fit %>%
  accuracy()
```

```
## # 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 multiplicative Training -0.115   4.60  3.02 0.199  4.08 0.542 0.606 -0.0131
```

```
fit2 %>%
  accuracy()
```

```
## # 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 Training 0.796  4.28  2.86  1.46  4.08 0.513 0.564 0.0134
```