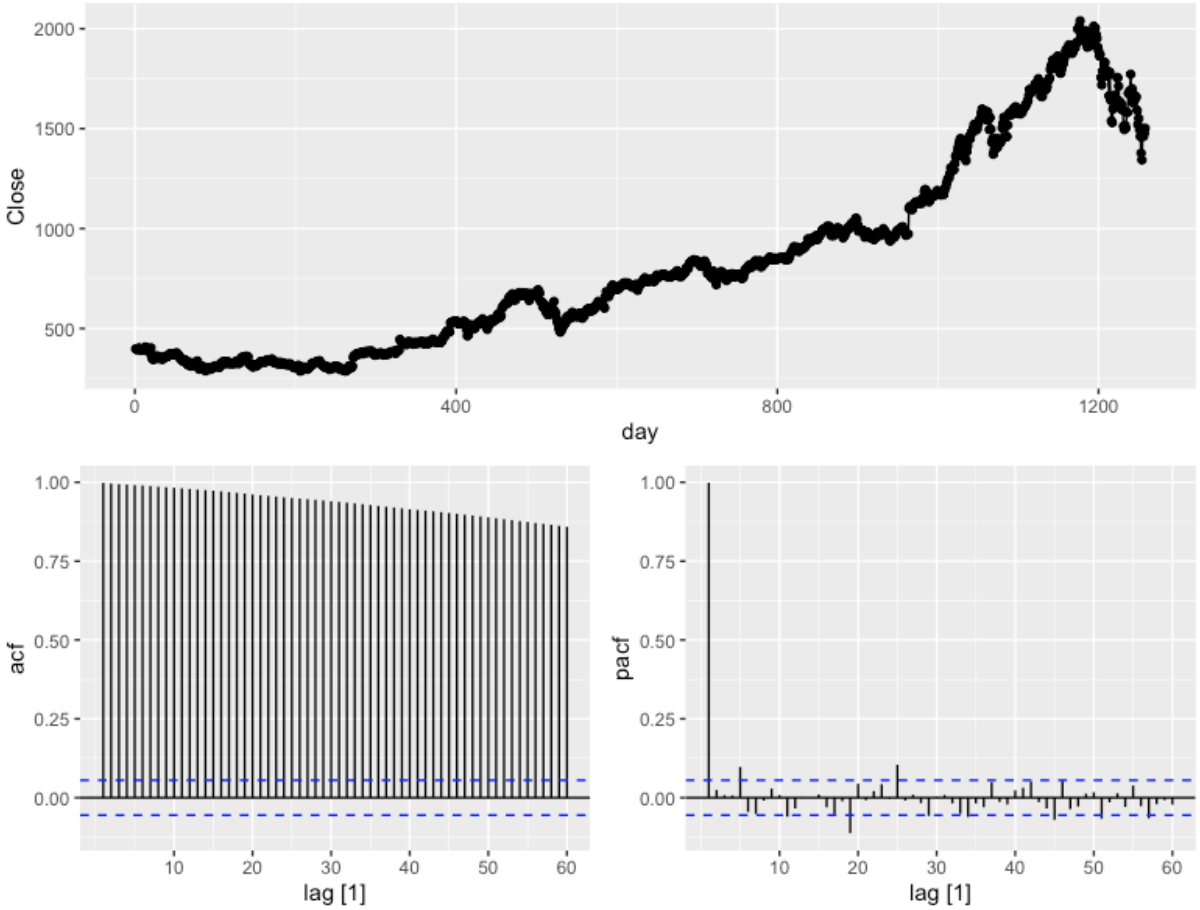
[1.a]
Since for all 3 series the lagged values are within the limits, all 3 are white noises.

[1.b]
Depending on the range T of all Time series, the limits are calculated as ±2/√T.
Therefore, the range for the 3 series are:
36 = ±0.3333333; 360 = ±0.1054093; 1000 = ±0.06324555

The autocorrelations are different for each series, since that is dependent on the data in the series.
Different white noise series can have different autocorrelations depending on their data.

[2]

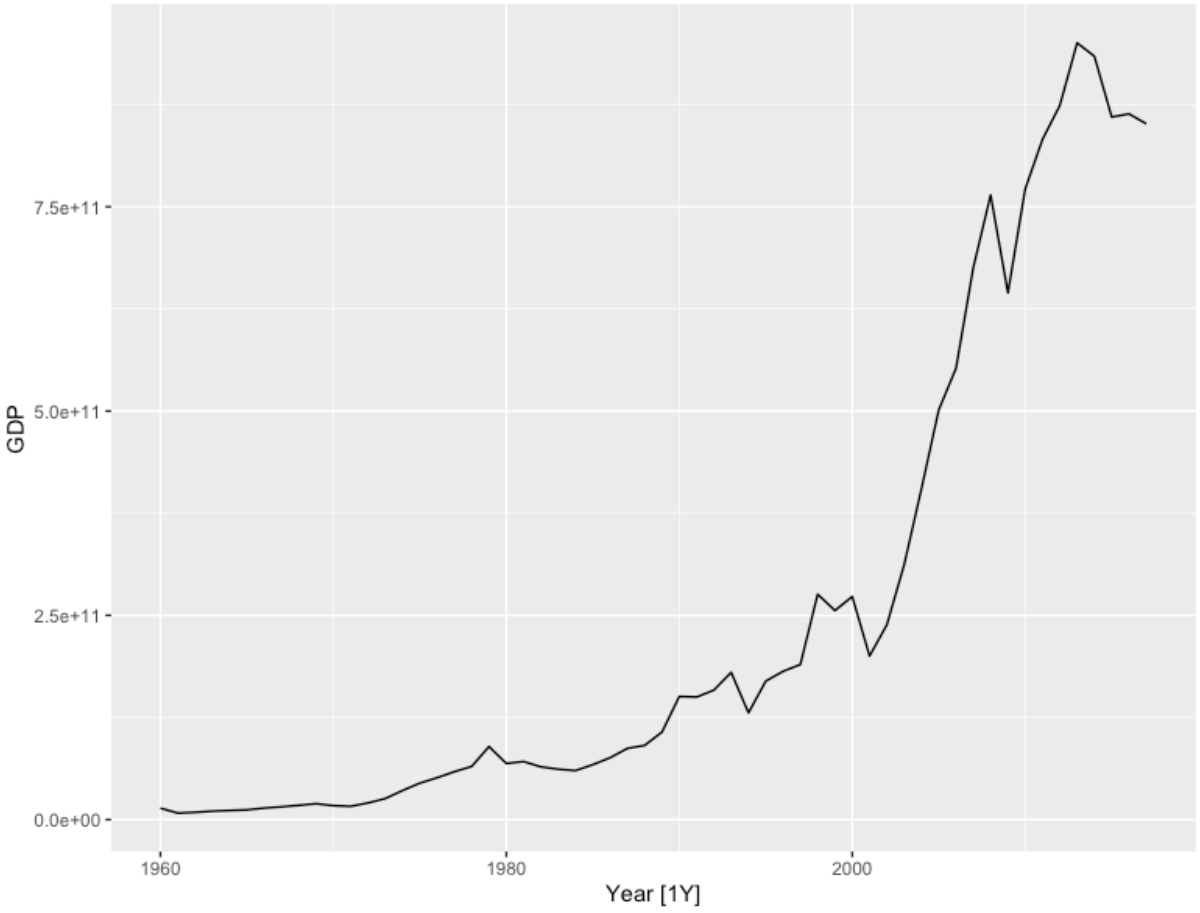
```
gafa_stock %>%
  filter(Symbol == "AMZN") %>%
  mutate(day = row_number()) %>%
  update_tsibble(index = day) %>%
  gg_tsdisplay(Close, plot_type = "partial", lag_max = 60)
```



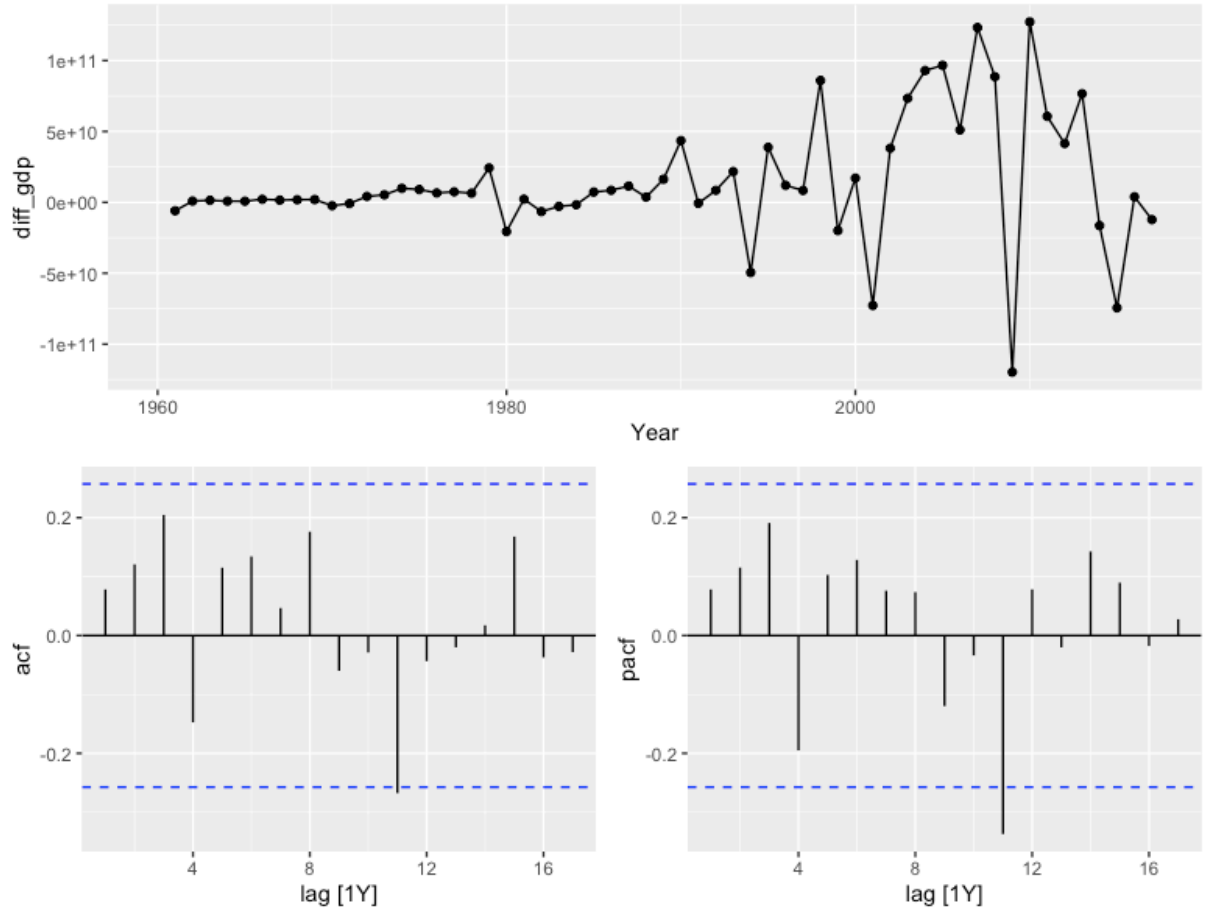
ACF plot shows that the first 60 lagged values have autocorrelations higher than the critical values, and are decreasing slowly. PACF plot shows a strong spike at lag 1 (near 1), meaning that stock price has strong correlation with the previous days price. There are also spikes at other lagged values, although these have no visible pattern. This shows that the series is non-stationary and should be differenced at least once, to remove effects of lagged values and stabilize the series.

[3.a]

```
global_economy %>%
  filter(Code == "TUR") %>%
  autoplot(GDP)
```

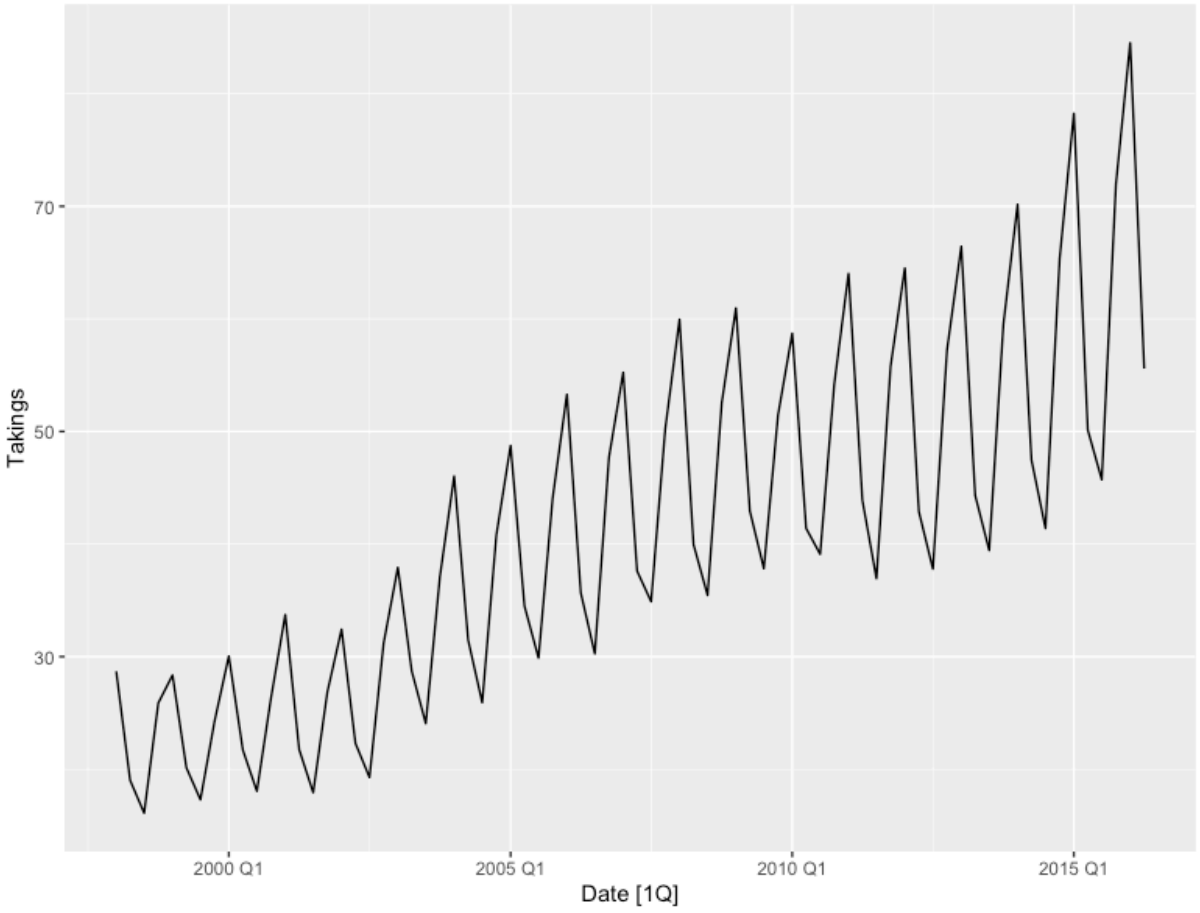


```
global_economy %>%
  filter(Code == "TUR") %>%
  mutate(diff_gdp = difference(GDP)) %>%
  gg_tsdisplay(diff_gdp, plot_type = “partial”)
```

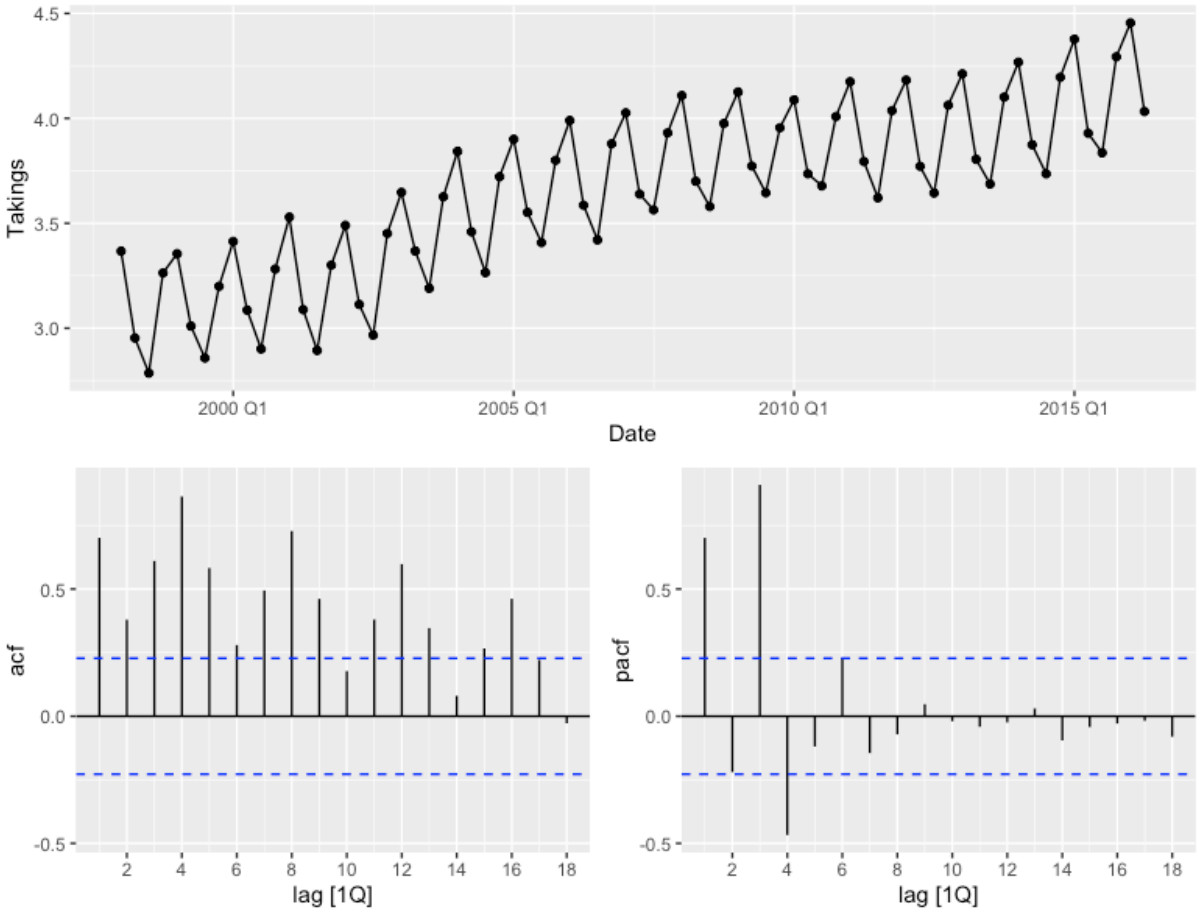


[3.b]

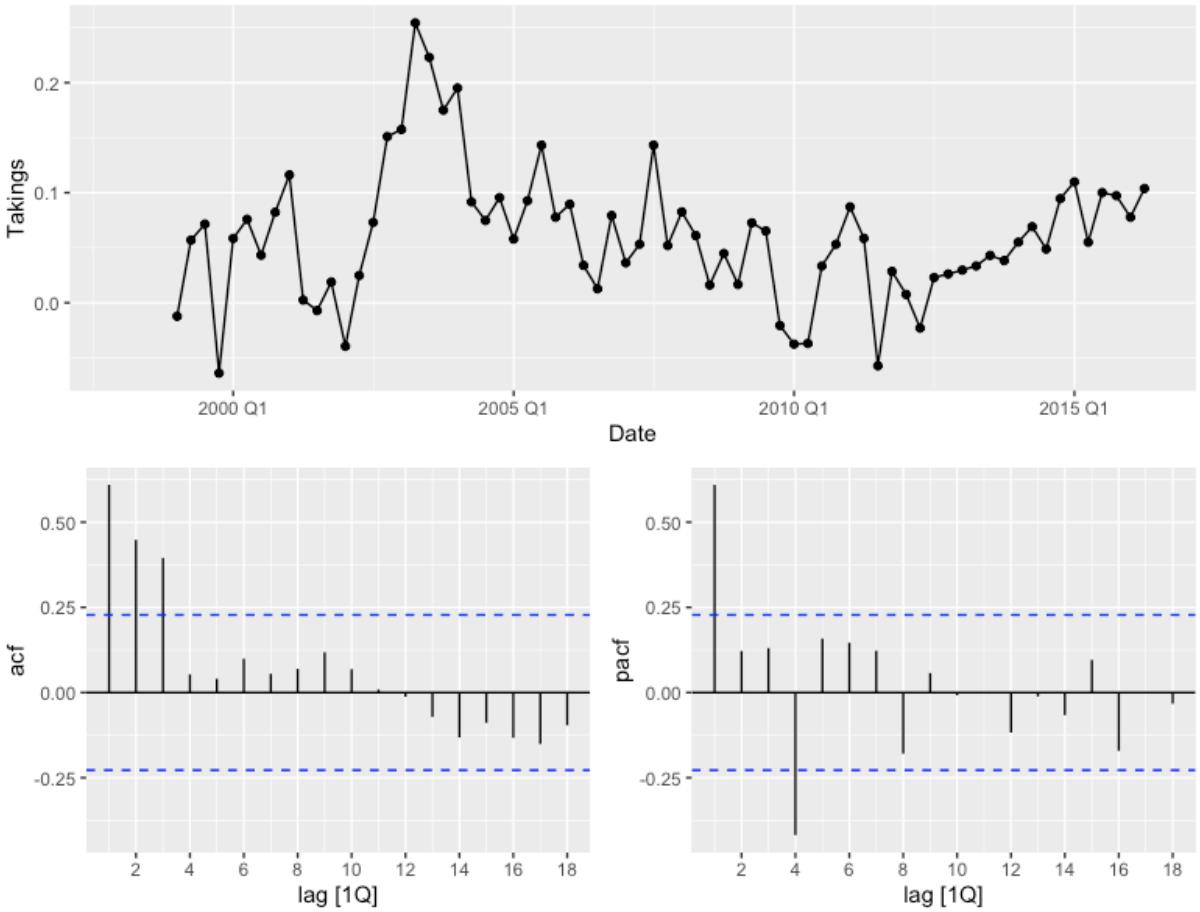
```
aus_accommodation %>%
  filter(State == "Tasmania") %>%
  autoplot(Takings)
```




```
lambda <- aus_accommodation %>%
  filter(State == "Tasmania") %>%
  features(Takings, guerrero) %>%
  pull(lambda_guerrero)

aus_accommodation %>%
  filter(State == "Tasmania") %>%
  transmute(Takings = box_cox(Takings, lambda)) %>%
  gg_tsdisplay(Takings, plot_type = “partial")
```



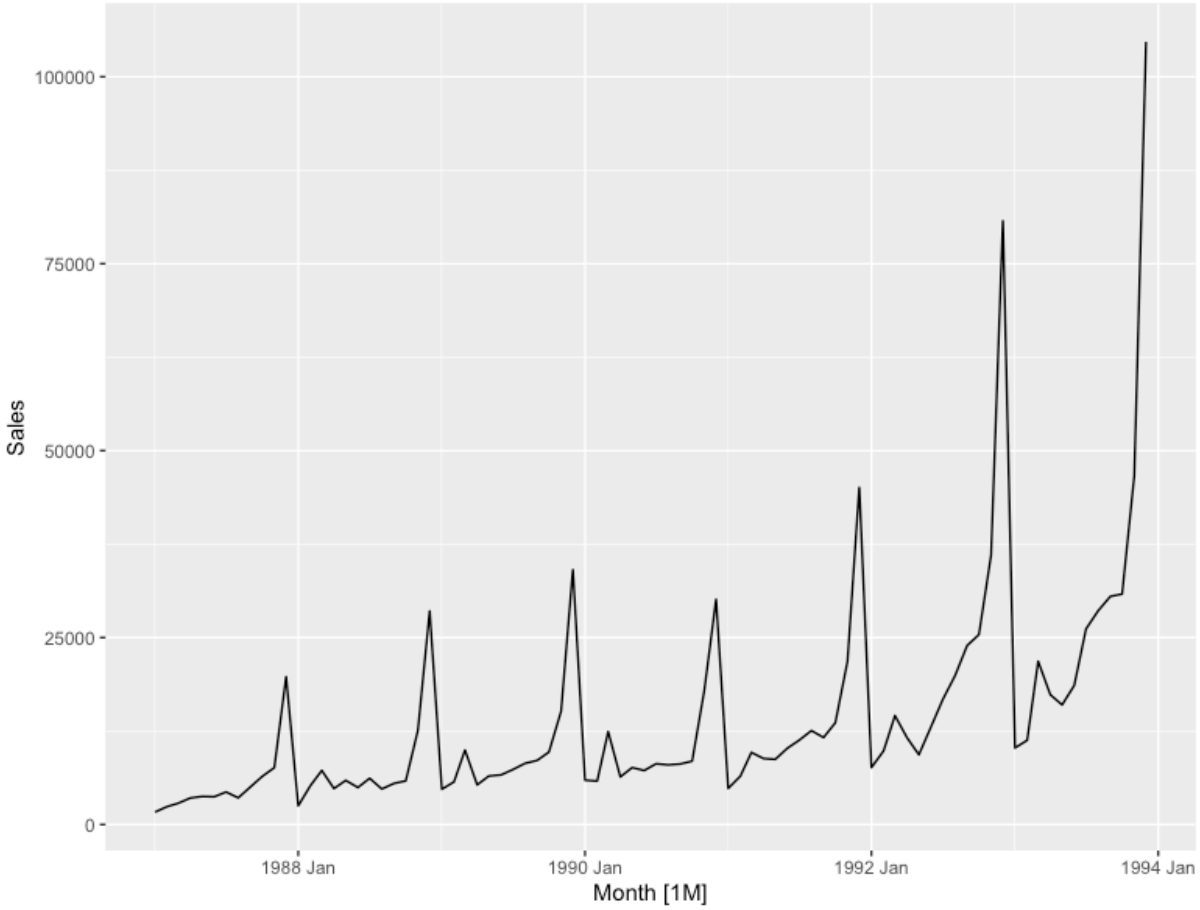

```
aus_accommodation %>%
  filter(State == "Tasmania") %>%
  transmute(Takings = box_cox(Takings, lambda)) %>%
  features(Takings, unitroot_nsdiffs)

aus_accommodation %>%
  filter(State == "Tasmania") %>%
  transmute(Takings = box_cox(Takings, lambda) %>% difference(4)) %>%
  features(Takings, unitroot_ndiffs)

aus_accommodation %>%
  filter(State == "Tasmania") %>%
  transmute(Takings = box_cox(Takings, lambda) %>% difference(4)) %>%
  gg_tsdisplay(Takings, plot_type = "partial")
```



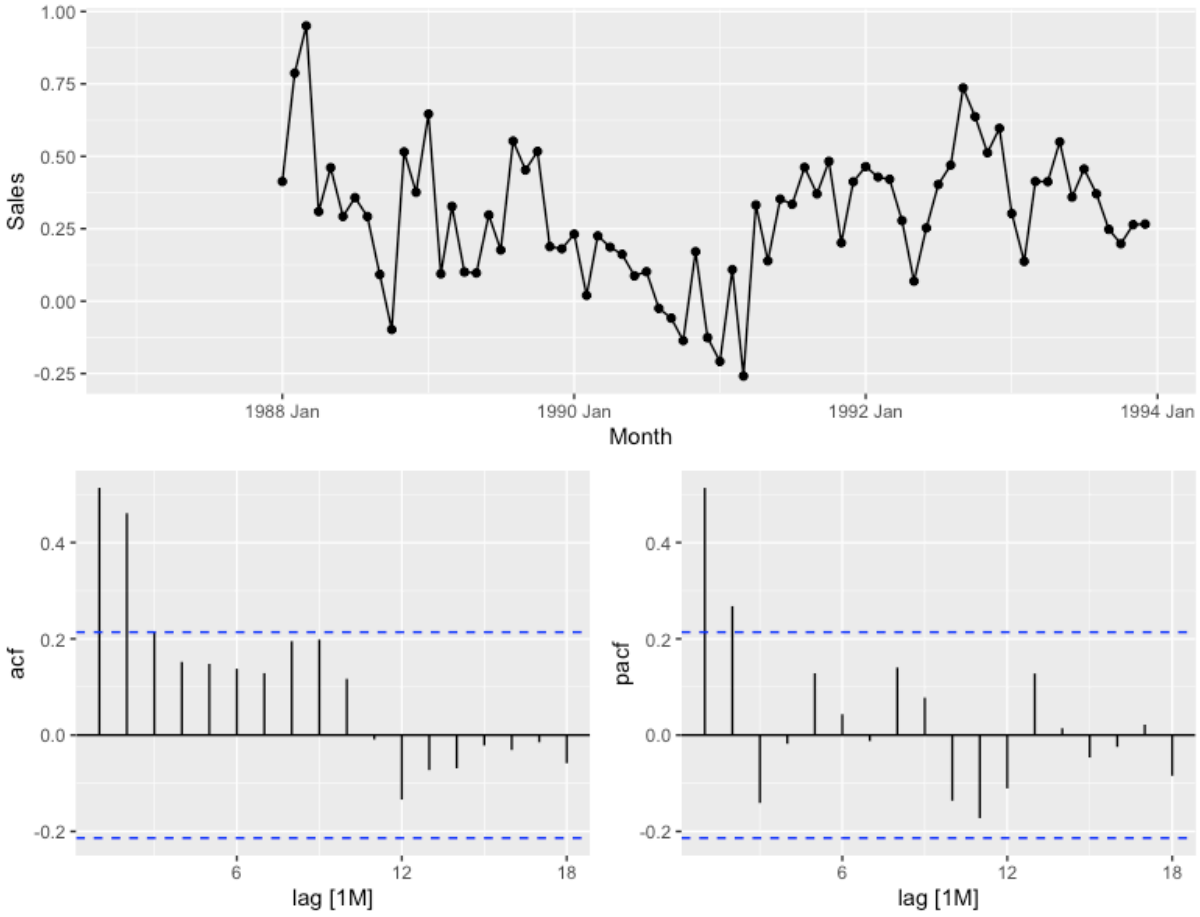

```
aus_accommodation %>%
  filter(State == "Tasmania") %>%
  transmute(Takings = box_cox(Takings, lambda) %>% difference(4)) %>%
  features(Takings, ljung_box)
```

[3.c]

```
souvenirs %>% autoplot(Sales)
lambda <- souvenirs %>% features(Sales, guerrero) %>% pull(lambda_guerrero)
```




```
souvenirs %>%
  transmute(Sales = box_cox(Sales, lambda)) %>%
  features(Sales, unitroot_nsdiffs)

souvenirs %>%
  transmute(Sales = box_cox(Sales, lambda)) %>%
  features(Sales, unitroot_ndiffs)

souvenirs %>%
  transmute(Sales = box_cox(Sales, lambda) %>% difference(12)) %>%
  gg_tsdisplay(Sales, plot_type = "partial")
```



[4]
Data needs 1 seasonal difference after Box-Cox transformation (lambda = 0.00213)

Model equation is:

w[t] = (sign(y[t])|y[t]|^0.00213 - 1)/0.00213
= (y[t]^0.00213 - 1)/0.00213 # since sales values are positive we can drop sign and ||

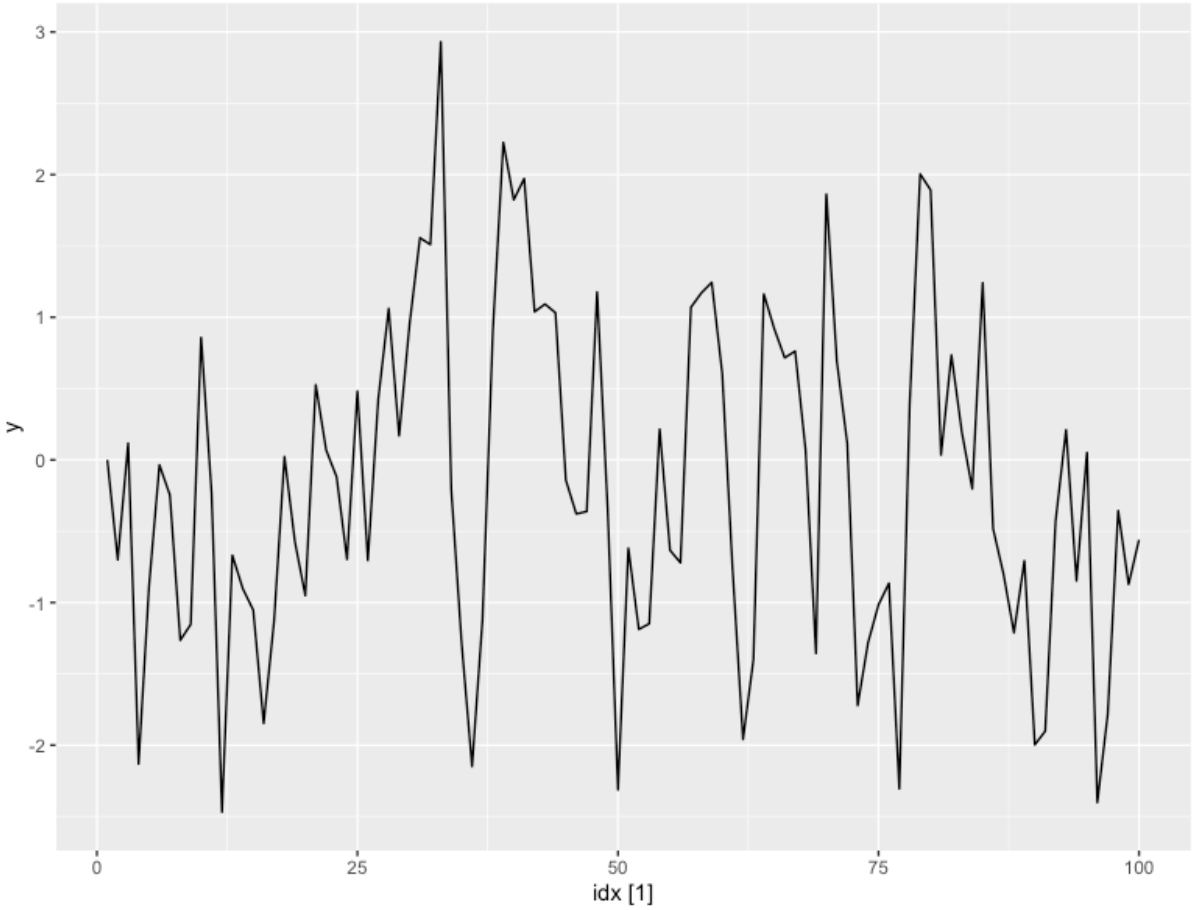
y'[t] = w[t] - w[t-4]
= w[t] - B^4w[t]
= (1 - B^4)w[t]

[6.a]

```
y <- numeric(100)
e <- rnorm(100)

for (i in 2:100) {
  y[i] <- 0.6*y[i-1] + e[i]
}

sim <- tsibble(idx = seq_len(100), y = y, index = idx)
```

[6.b]

```
autoplot(sim, y)
```




```
ar1generator <- function(phi1 = 0.6){
  y <- numeric(100)
  e <- rnorm(100)

  for (i in 2:100) {
    y[i] <- phi1*y[i-1] + e[i]
  }

  sim <- tsibble(idx = seq_len(100), y = y, phi = phi1, index = idx, key = phi)

  return(sim)
}

bind_rows(
  ar1generator(),
  ar1generator(0.3),
  ar1generator(0.9)
) %>%
  autoplot(y)
```

As phi increases, the variation in y increases.

[6.c]

```
ma1generator <- function(theta1 = 0.6){
  y <- numeric(100)
  e <- rnorm(100)

  for (i in 2:100) {
    y[i] <- e[i] + theta1*e[i-1]
  }

  sim <- tsibble(idx = seq_len(100), y = y, theta1 = theta1, index = idx, key = theta1)

  return(sim)
}
```

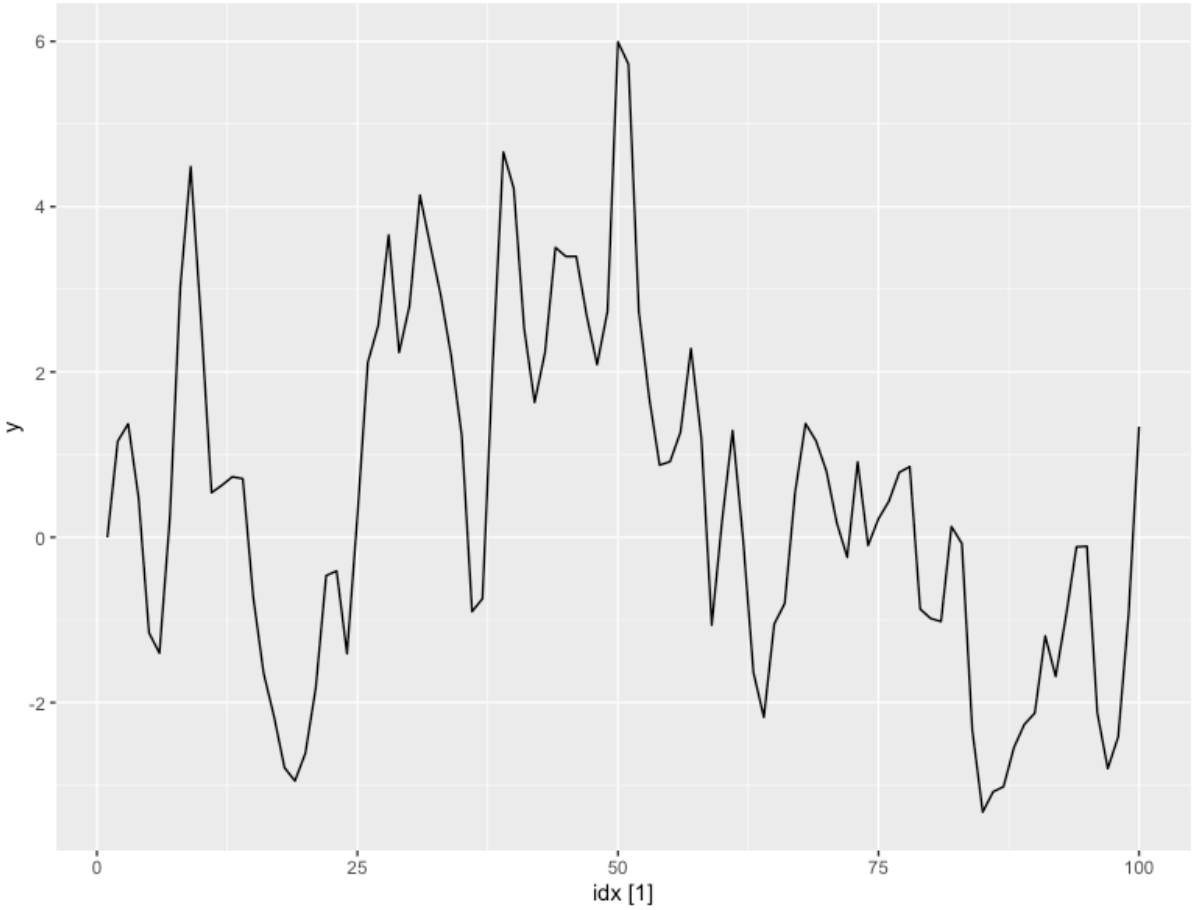
[6.d]

```
bind_rows(
  ma1generator(),
  ma1generator(0.3),
  ma1generator(0.9)
) %>%
  autoplot(y)
```

As theta increases, the variation in y increases.

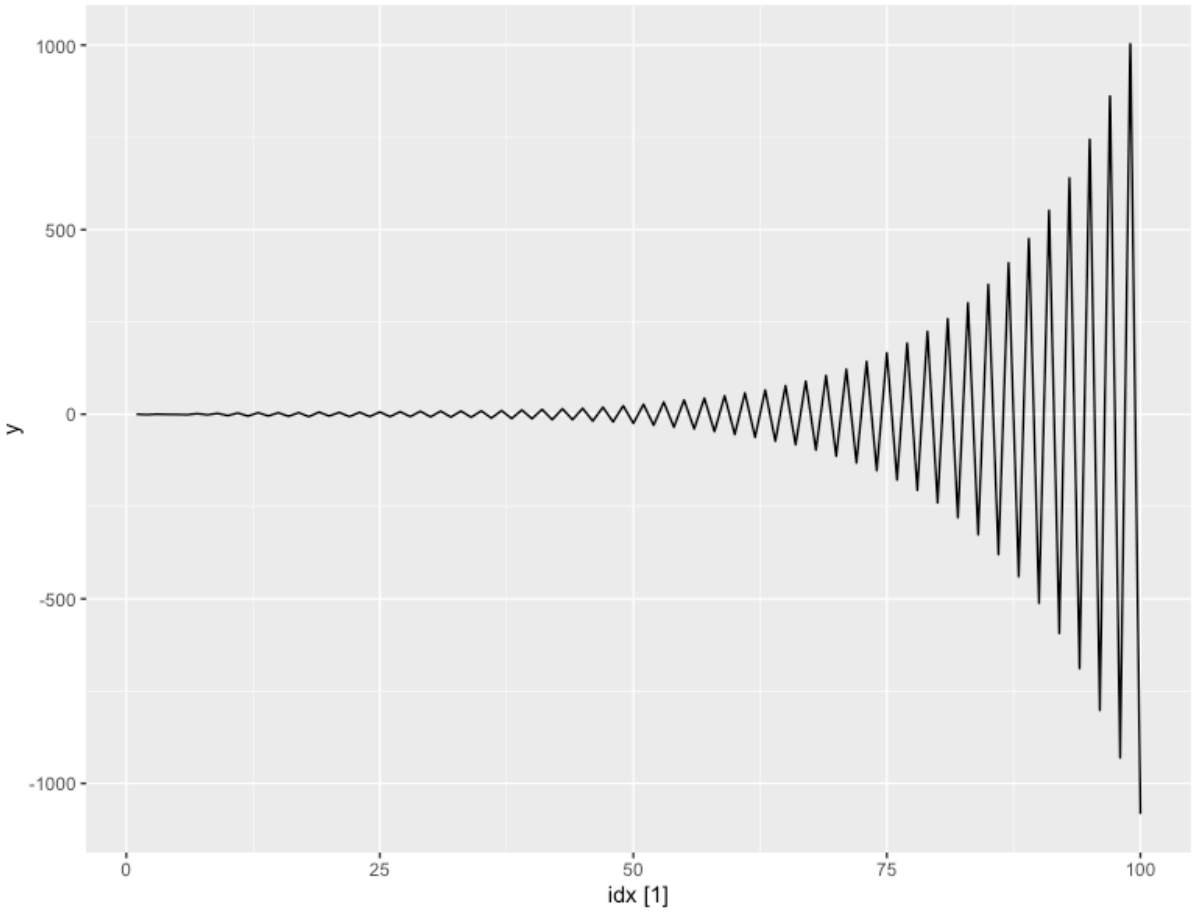
[6.e]

```
arma11generator <- function(phi1 = 0.6, theta1 = 0.6){
  y <- numeric(100)
  e <- rnorm(100)

  for (i in 2:100) {
    y[i] <- phi1*y[i-1] + e[i] + theta1*e[i-1]
  }

  sim <- tsibble(idx = seq_len(100), y = y, phi1 = phi1, theta1 = theta1, index = idx, key = c(phi1,
theta1))

  return(sim)
}

autoplot(arma11generator(), y)
```



[6.f]

```
ar2generator <- function(phi1 = -0.8, phi2 = 0.3){
  y <- numeric(100)
  e <- rnorm(100)

  y[2] <- phi1*y[i] + e[2]

  for (i in 3:100) {
    y[i] <- phi1*y[i-1] + phi2*y[i-2] + e[i]
  }

  sim <- tsibble(idx = seq_len(100), y = y, phi1 = phi1, phi2 = phi2, index = idx, key = c(phi1, phi2))

  return(sim)
}

autoplot(ar2generator(), y)
```

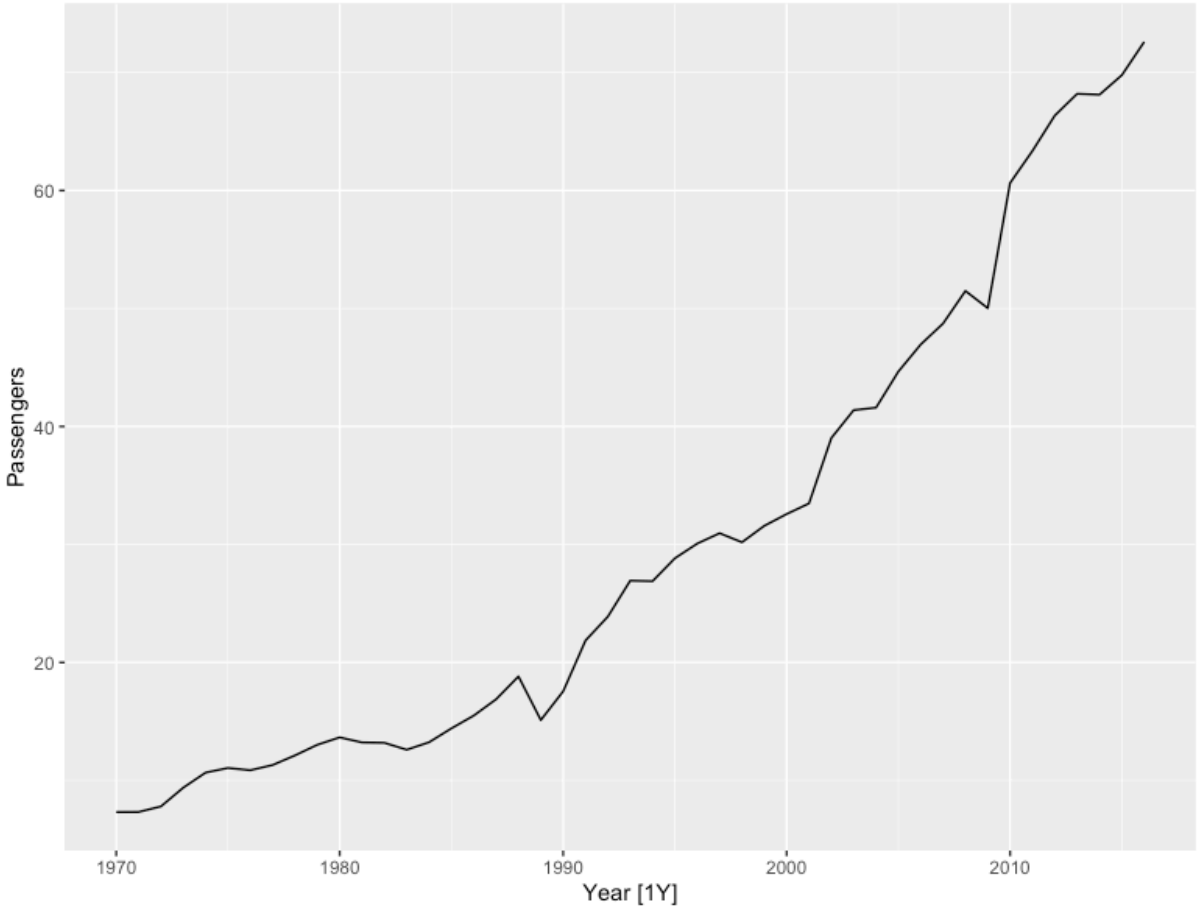


[6.g]
p1 <- autoplot(arma11generator(), y)
p2 <- autoplot(ar2generator(), y)

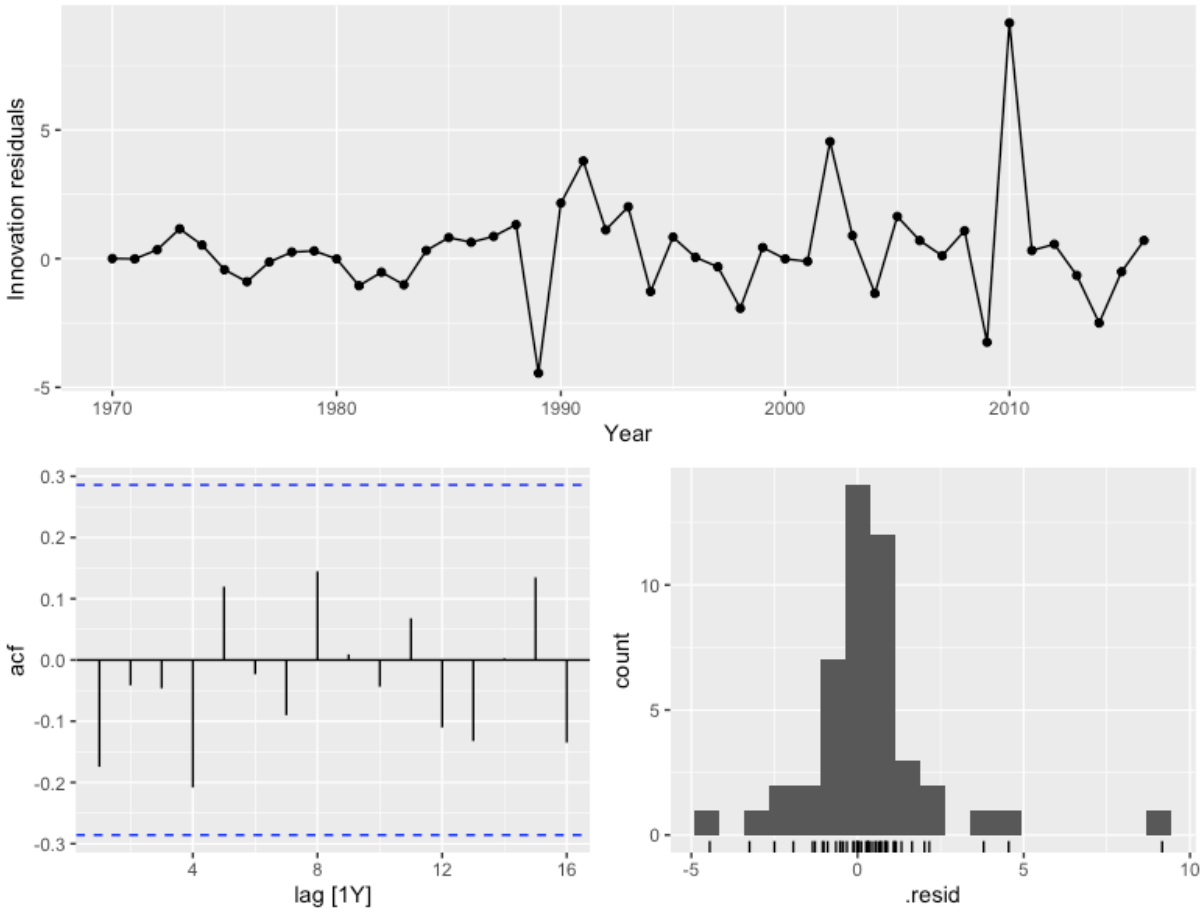
p1 / p2

data from AR(2) increased with oscelations, and is non-stationary.
AR(1) model is stationary.

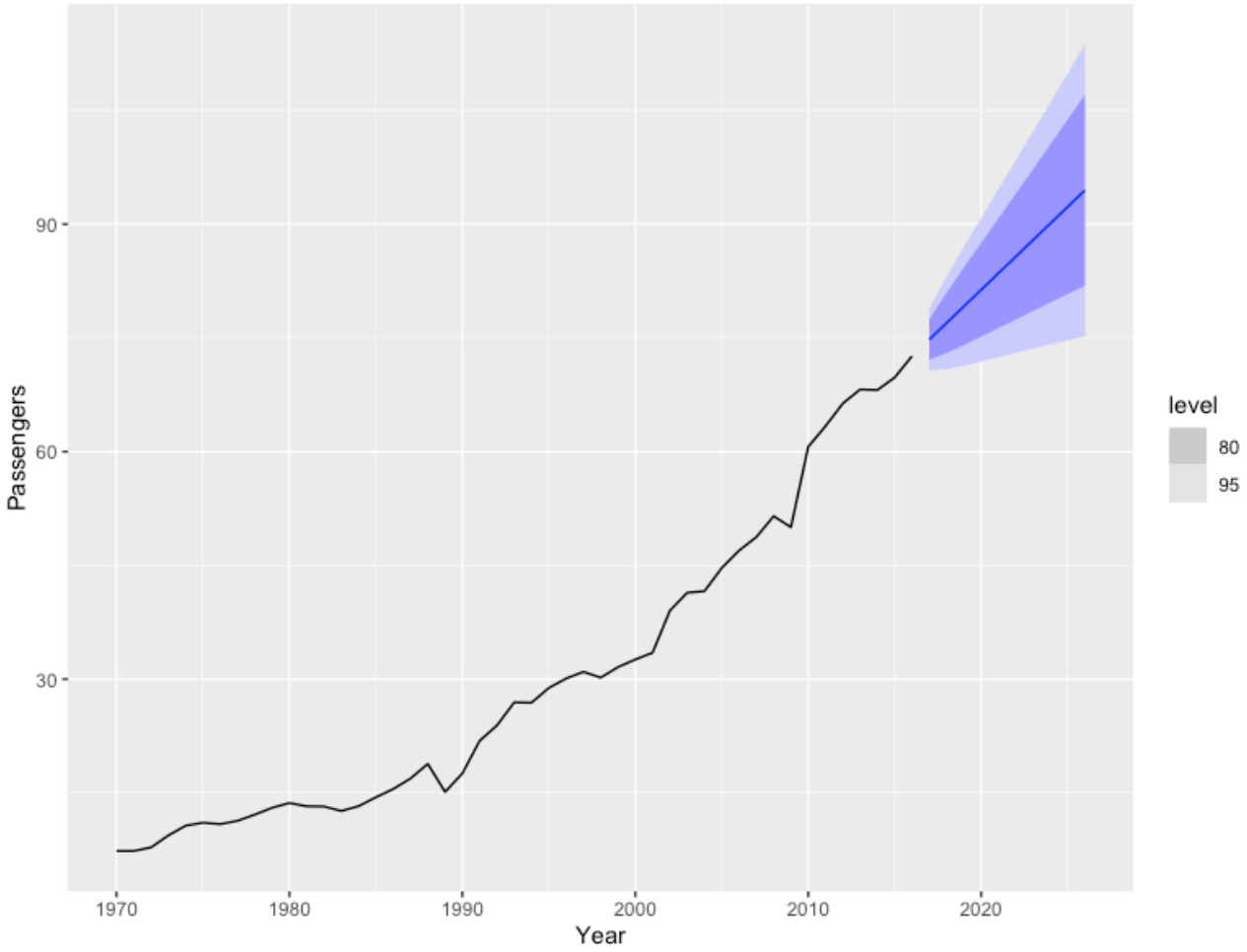
[7.a]
aus_airpassengers %>% autoplot(Passengers)




```
fit <- aus_airpassengers %>%
  model(ARIMA(Passengers))

fit %>% gg_tsresiduals()
```




```
fit %>%
  forecast(h = 10) %>%
  autoplot(aus_airpassengers)
```

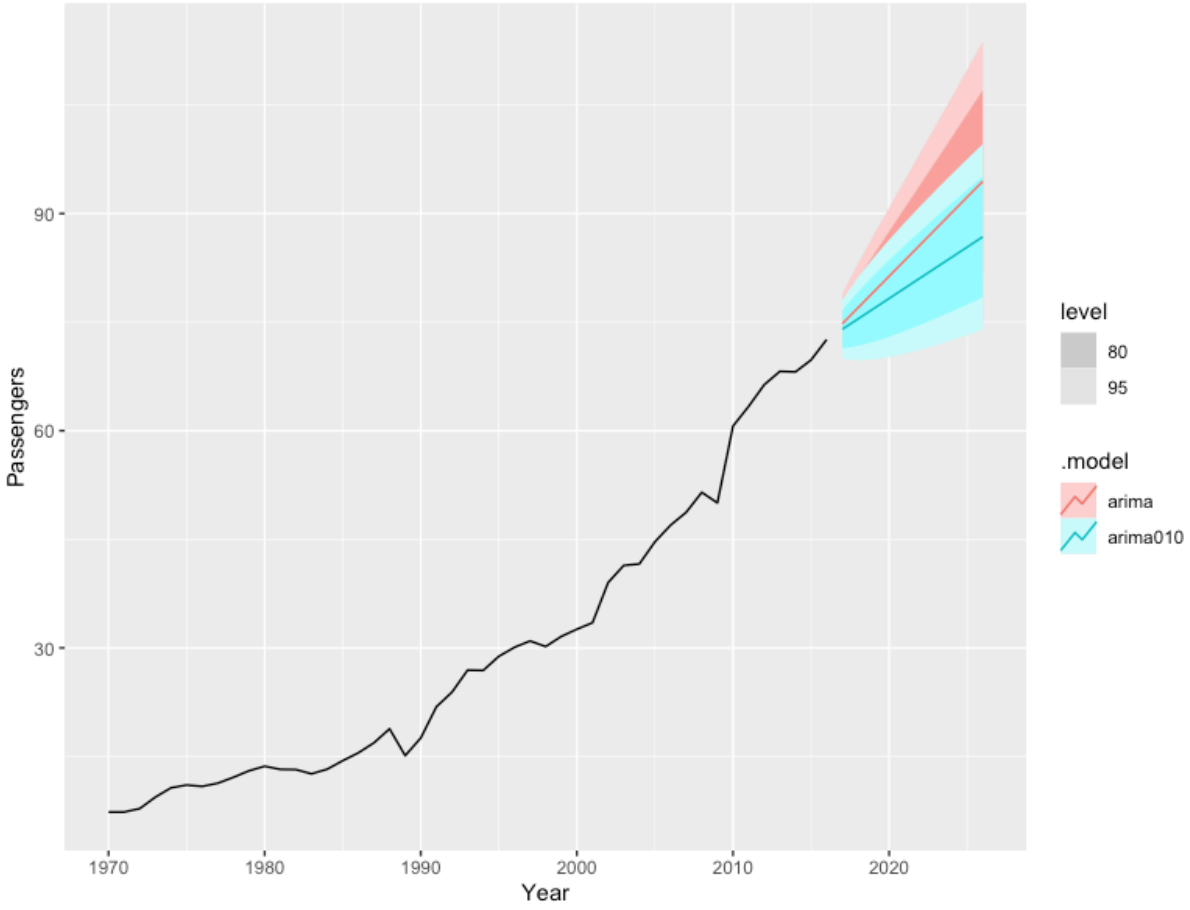


[7.b]

$$y''[t] = c + \varepsilon[t] + \theta[1]\varepsilon[t-1] + \theta[2]\varepsilon[t-2]$$
$$= c + \varepsilon[t] + \theta[1]B\varepsilon[t] + \theta[2]B[^2]\varepsilon[t]$$
$$= c + \varepsilon[t] (1 + \theta[1]B + \theta[2]B[^2])$$

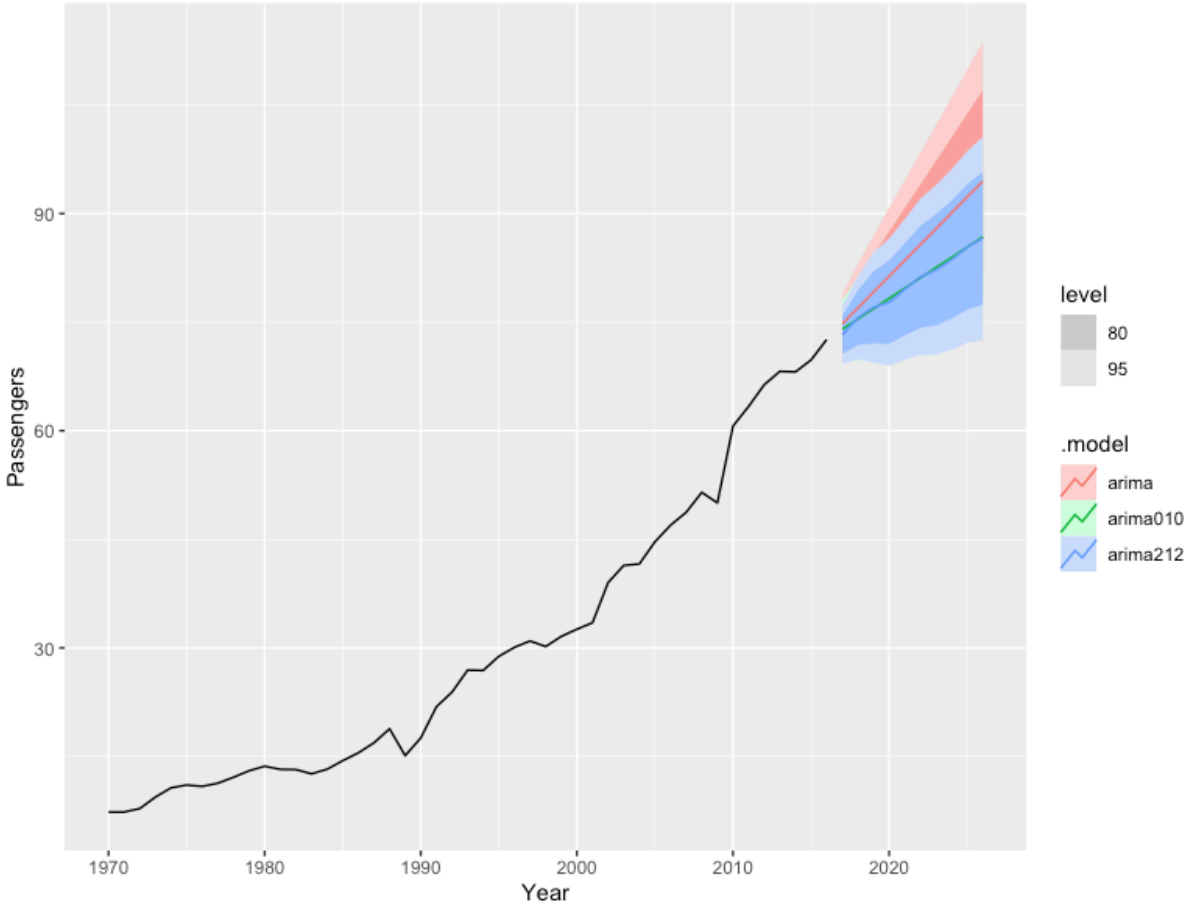
[7.c]

```
aus_airpassengers %>%
  model(
    arima = ARIMA(Passengers),
    arima010 = ARIMA(Passengers ~ 1 + pdq(0,1,0))
  ) %>%
  forecast(h = 10) %>%
  autoplot(aus_airpassengers)
```

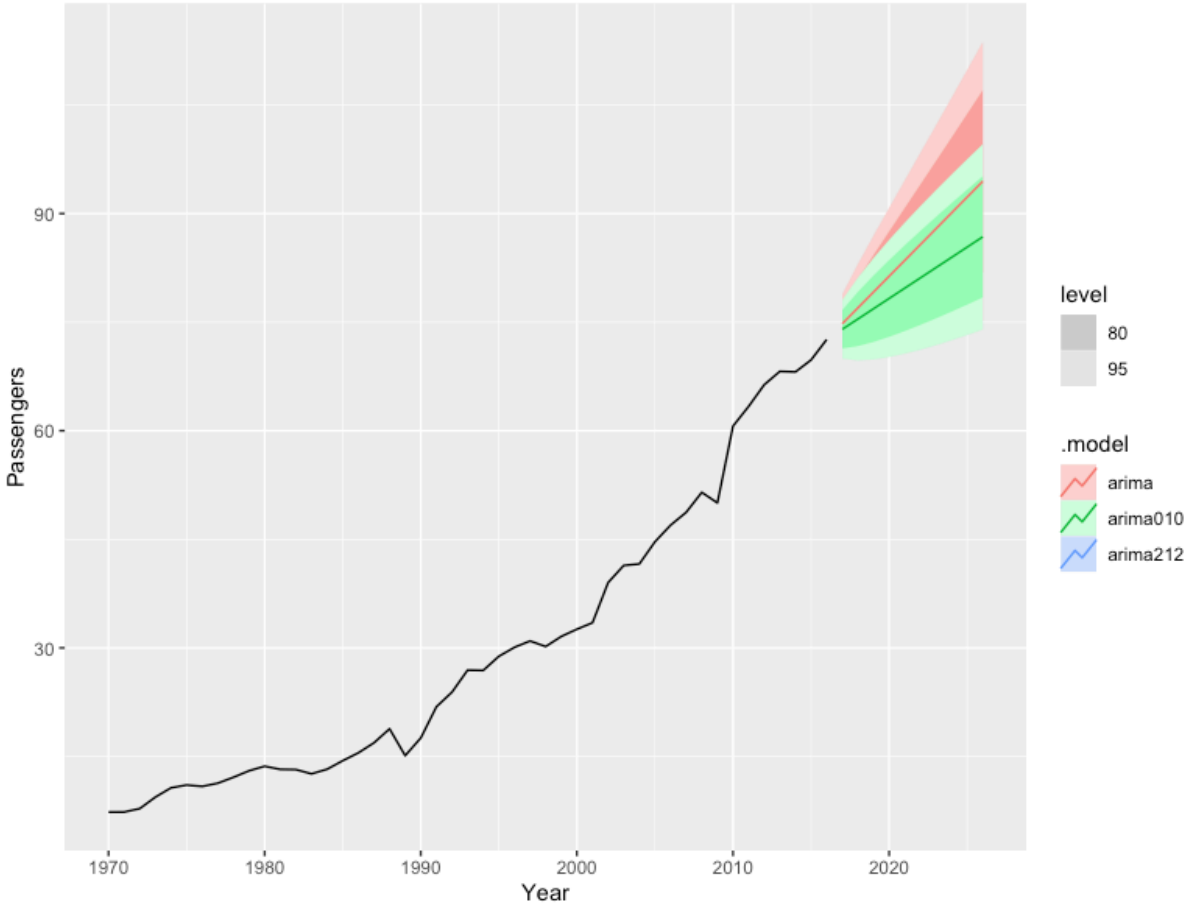


[7.d]

```
aus_airpassengers %>%
  model(
    arima = ARIMA(Passengers),
    arima010 = ARIMA(Passengers ~ 1 + pdq(0,1,0)),
    arima212 = ARIMA(Passengers ~ 1 + pdq(2,1,2))
  ) %>%
  forecast(h = 10) %>%
  autoplot(aus_airpassengers)
```



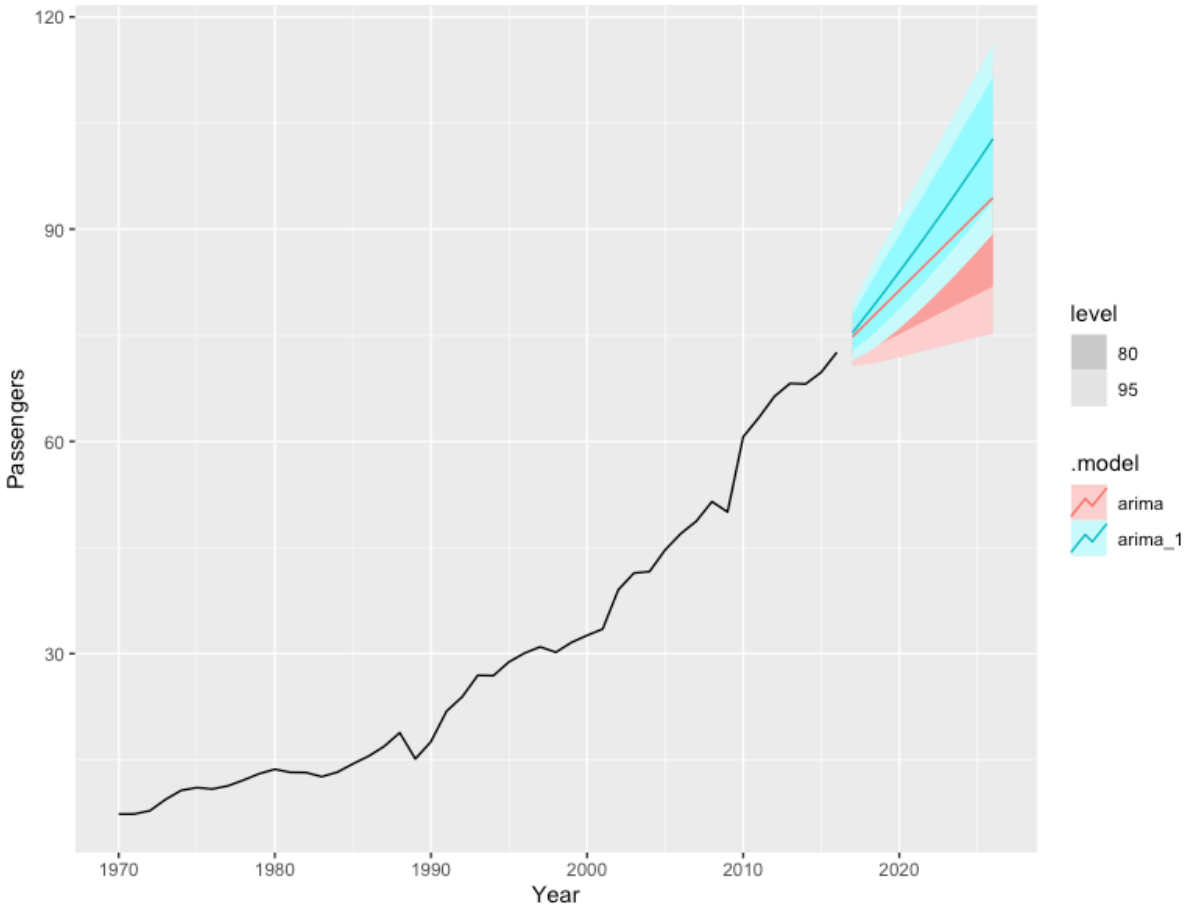

```
aus_airpassengers %>%
  model(
    arima = ARIMA(Passengers),
    arima010 = ARIMA(Passengers ~ 1 + pdq(0,1,0)),
    arima212 = ARIMA(Passengers ~ 0 + pdq(2,1,2))
  ) %>%
  forecast(h = 10) %>%
  autoplot(aus_airpassengers)
```



Removing the constant gives an error as the AR becomes non-stationary.

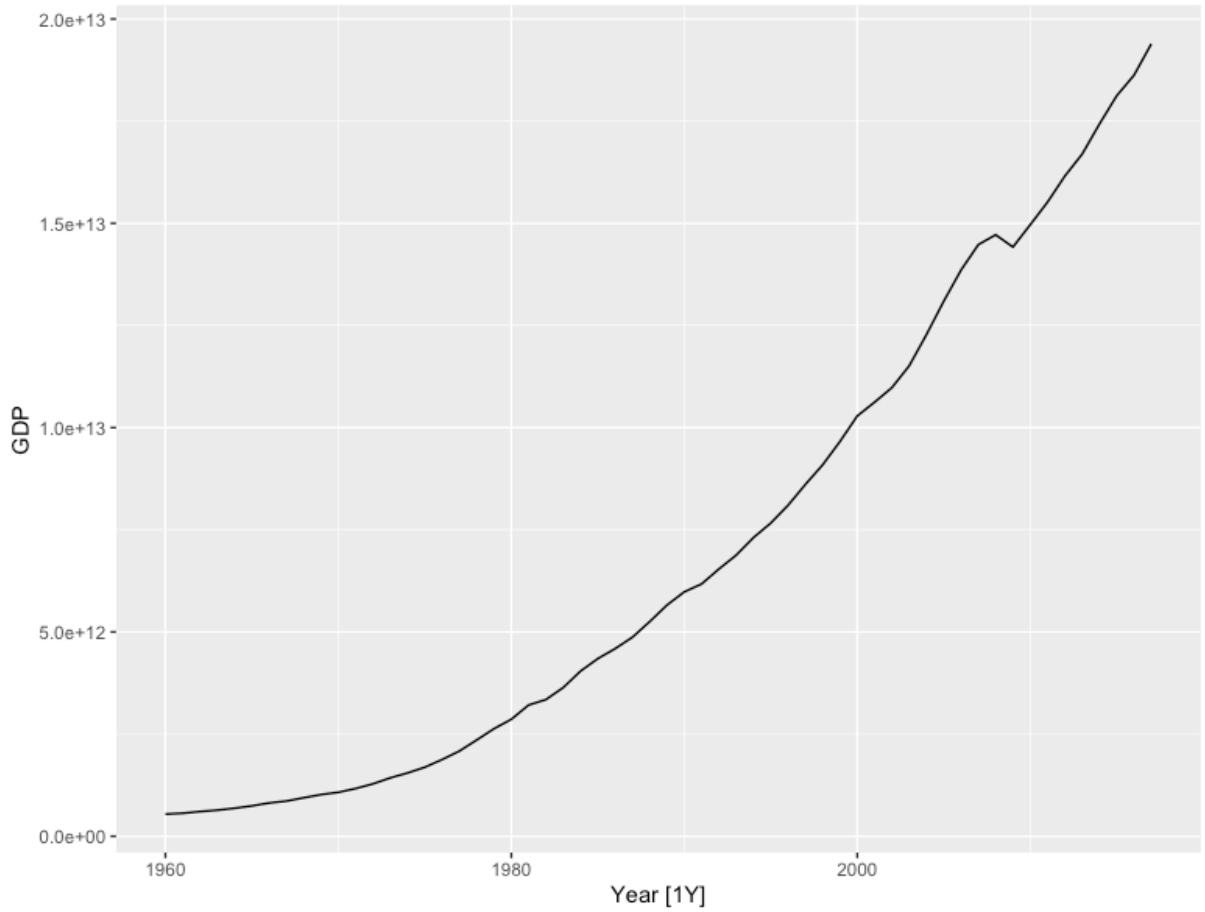
[7.e]

```
aus_airpassengers %>%
  model(
    arima = ARIMA(Passengers),
    arima_1 = ARIMA(Passengers ~ 1 + pdq(0,2,1))
  ) %>%
  forecast(h = 10) %>%
  autoplot(aus_airpassengers)
```



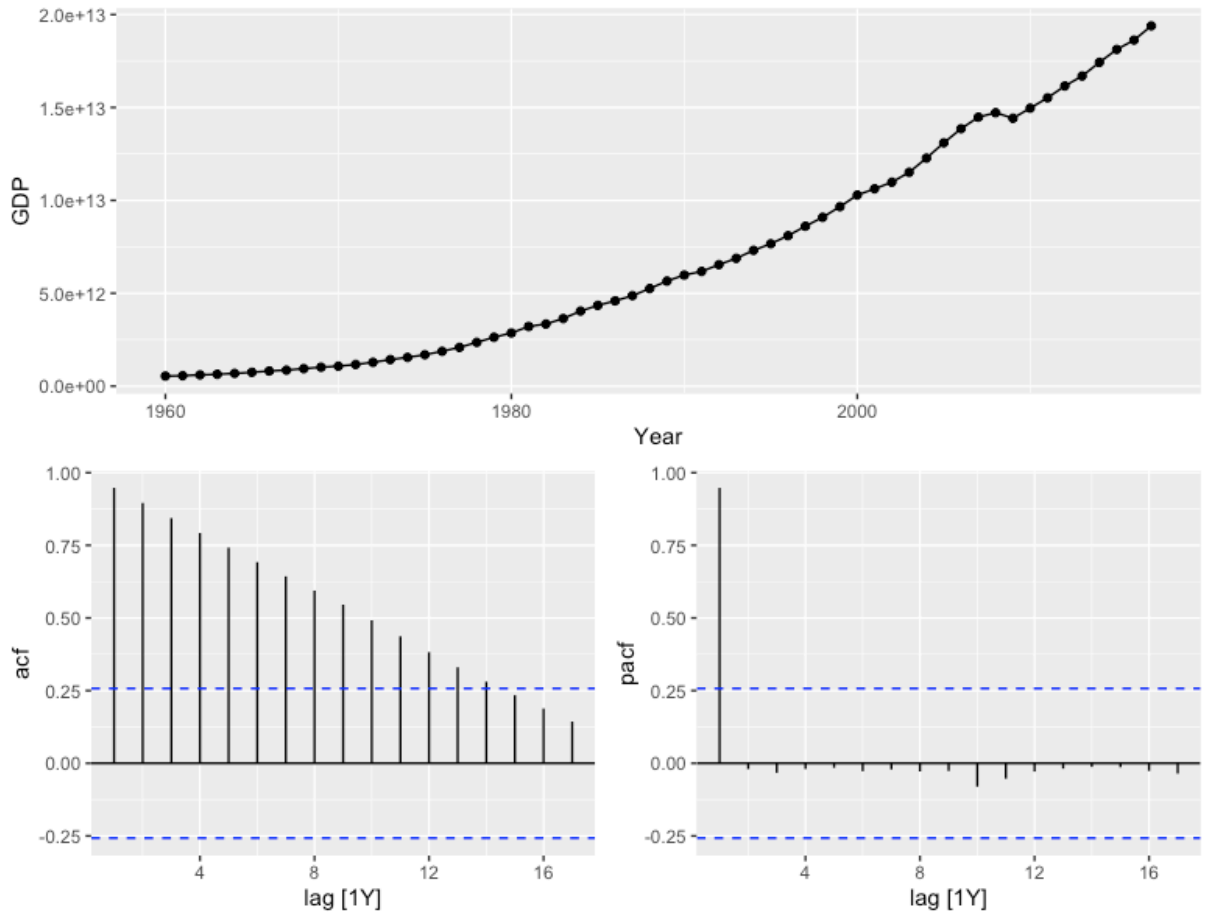
Introducing constant causes the model to give a quadratic or higher order polynomial trend. This is generally discouraged. The trend is higher than for without constant, but the confidence interval is narrower.

[8.a]
gdp <- global_economy %>% filter(Code == "USA") %>% select(GDP)
gdp %>% autoplot(GDP)

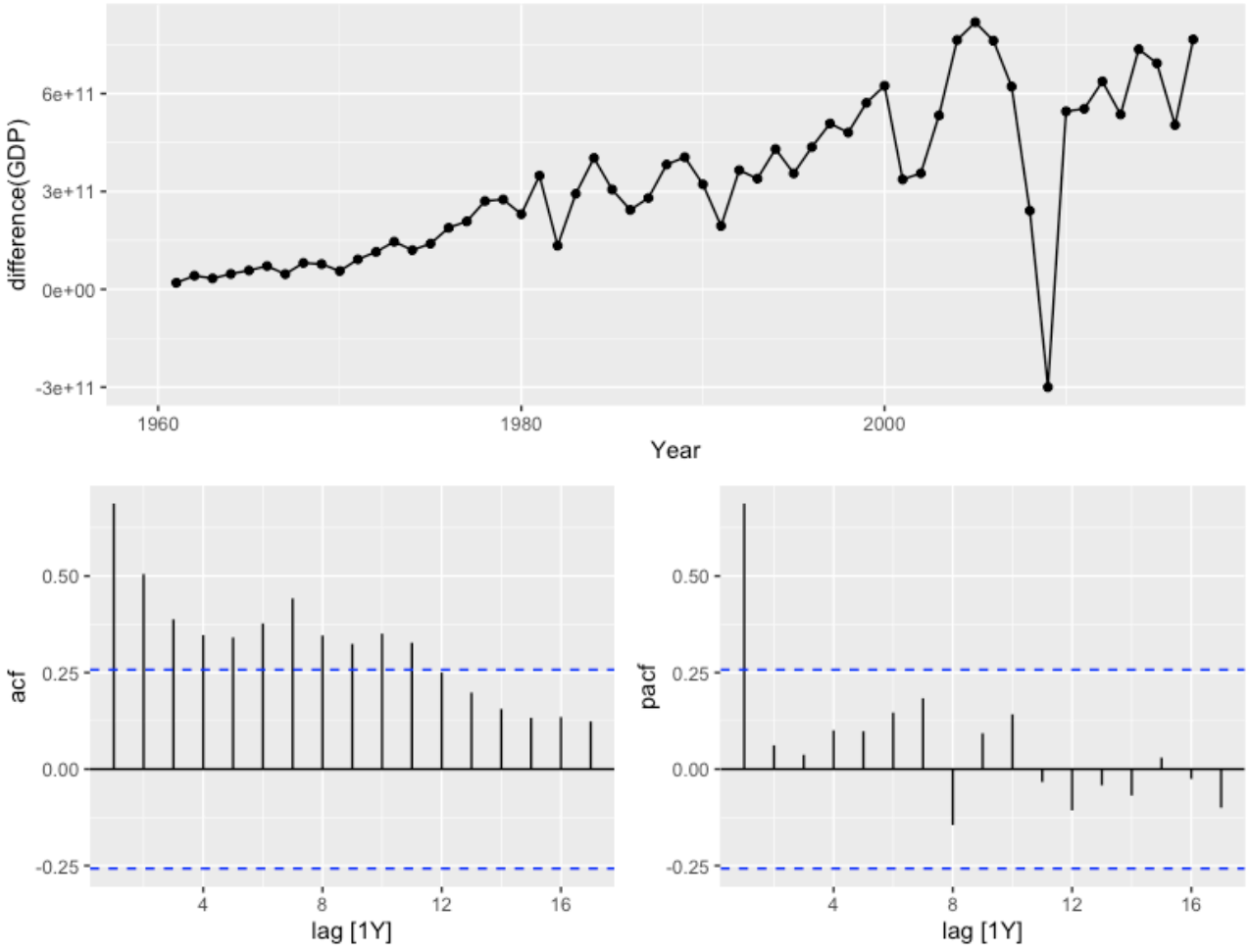


[8.b]
```
gdp %>%
  model(ARIMA(GDP))
```

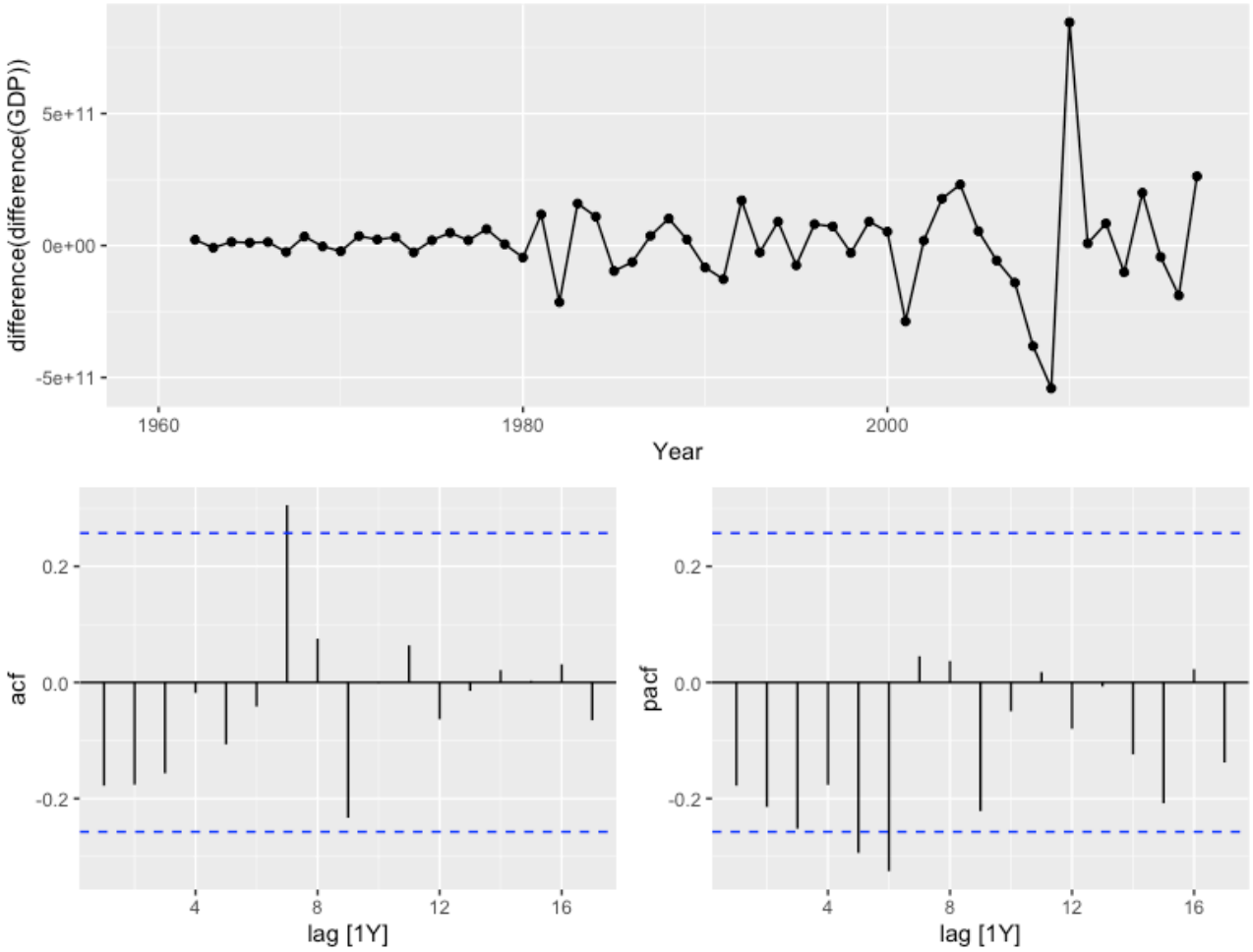
[8,c]
```
gdp %>%
  gg_tsdisplay(GDP, plot_type = "partial")
```



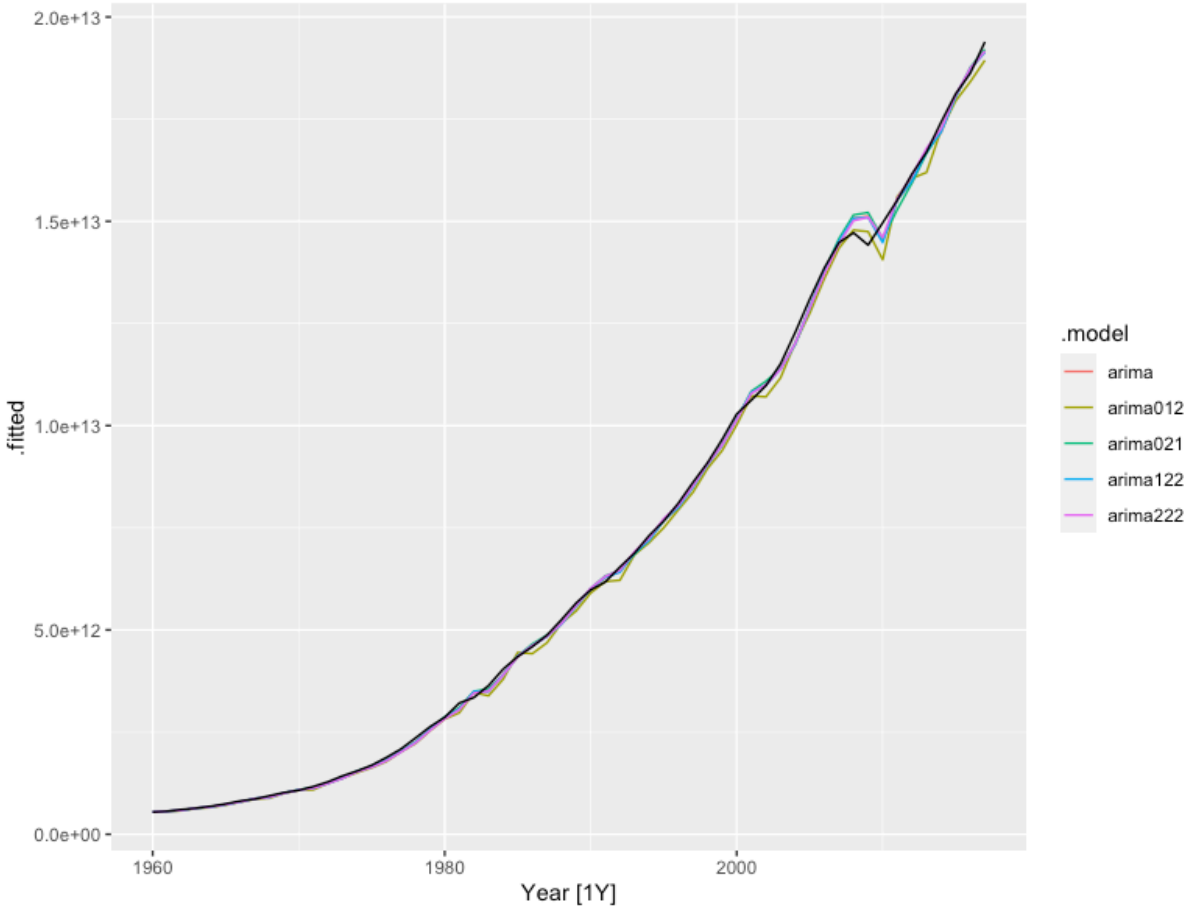

```
gdp %>%
  gg_tsdisplay(difference(GDP), plot_type = "partial")
```




```
gdp %>%
  gg_tsdisplay(difference(difference(GDP)), plot_type = "partial")
```

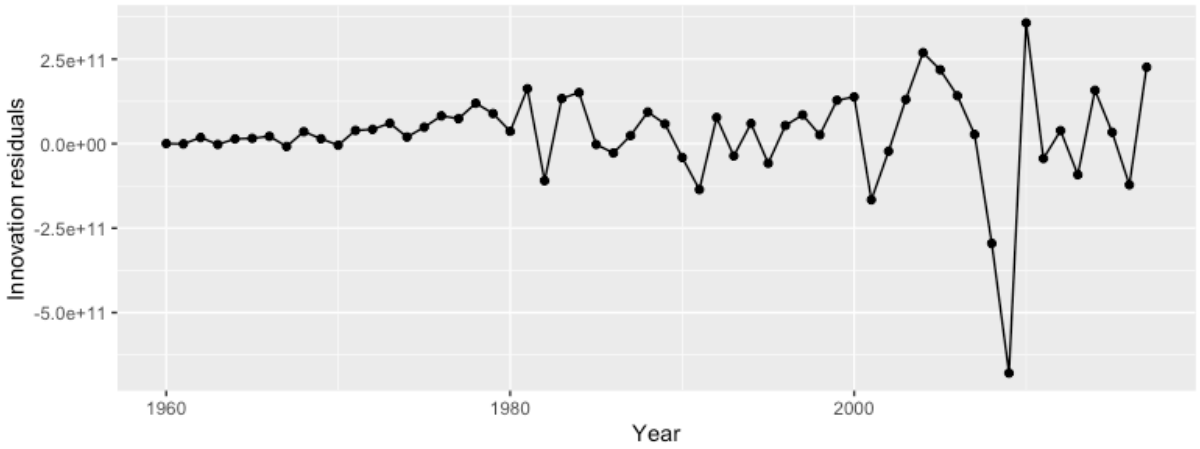


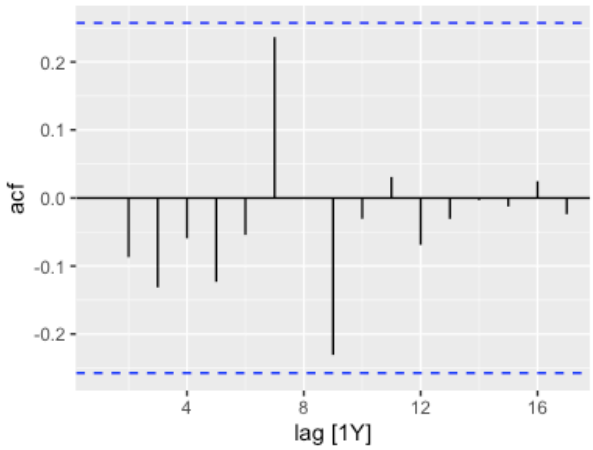

```
gdp %>%
  model(
    arima = ARIMA(GDP),
    arima122 = ARIMA(GDP ~ pdq(1,2,2)),
    arima021 = ARIMA(GDP ~ pdq(0,2,1)),
    arima012 = ARIMA(GDP ~ pdq(0,1,2)),
    arima222 = ARIMA(GDP ~ pdq(2,2,2))
  ) %>%
  # accuracy() %>% arrange(RMSE)
  augment() %>%
  autoplot(.fitted) +
  autolayer(gdp, GDP)
```

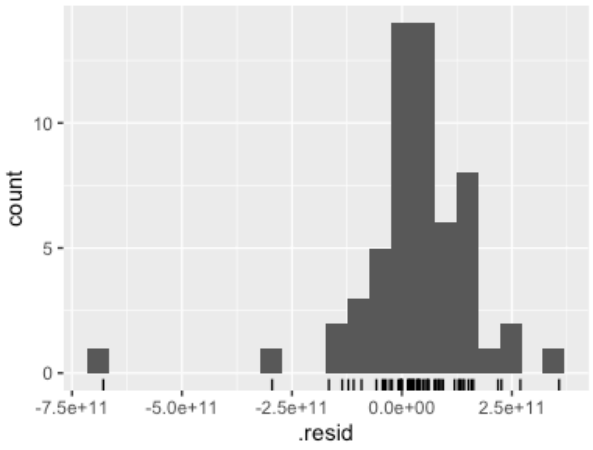


[8.d]

```
gdp %>%
  model(ARIMA(GDP ~ pdq(2,2,2))) %>%
  gg_tsresiduals()
```

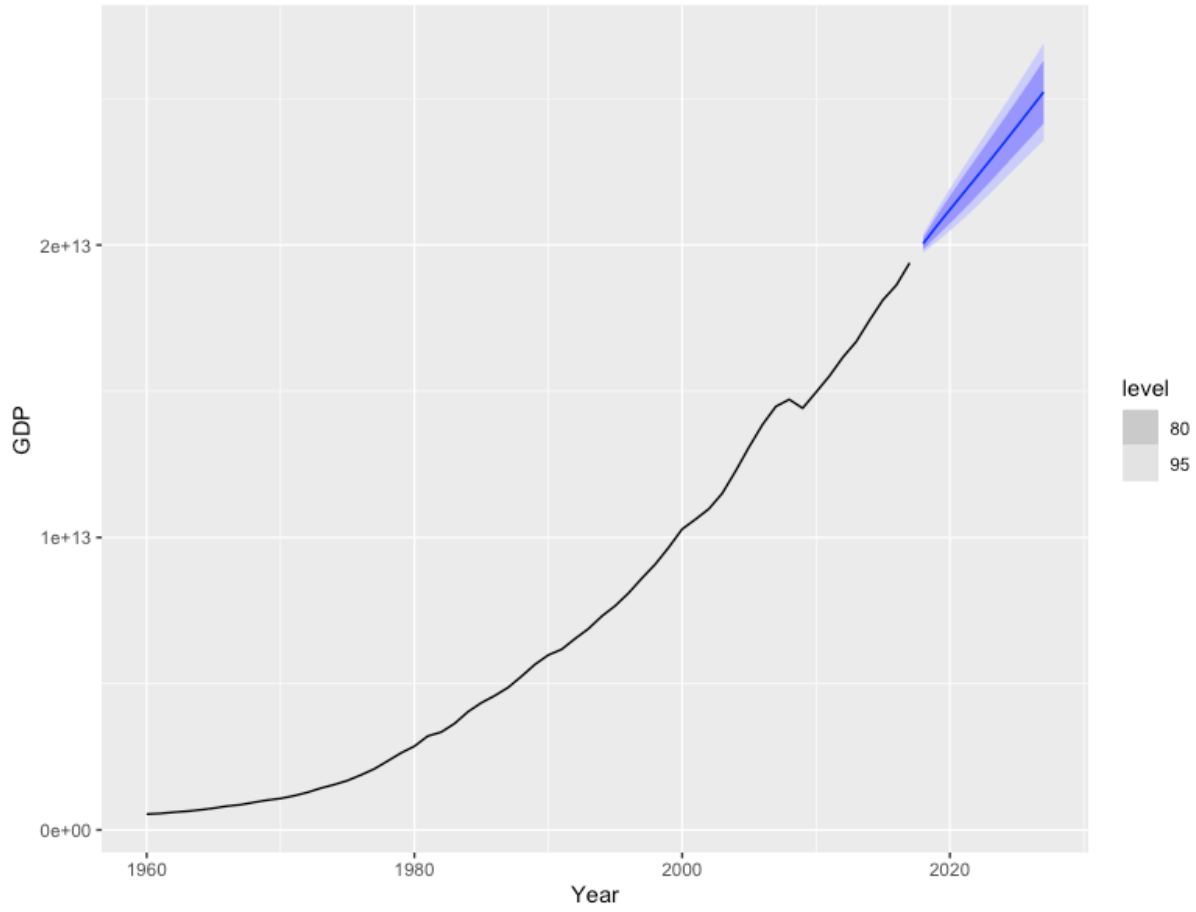






[8.e]

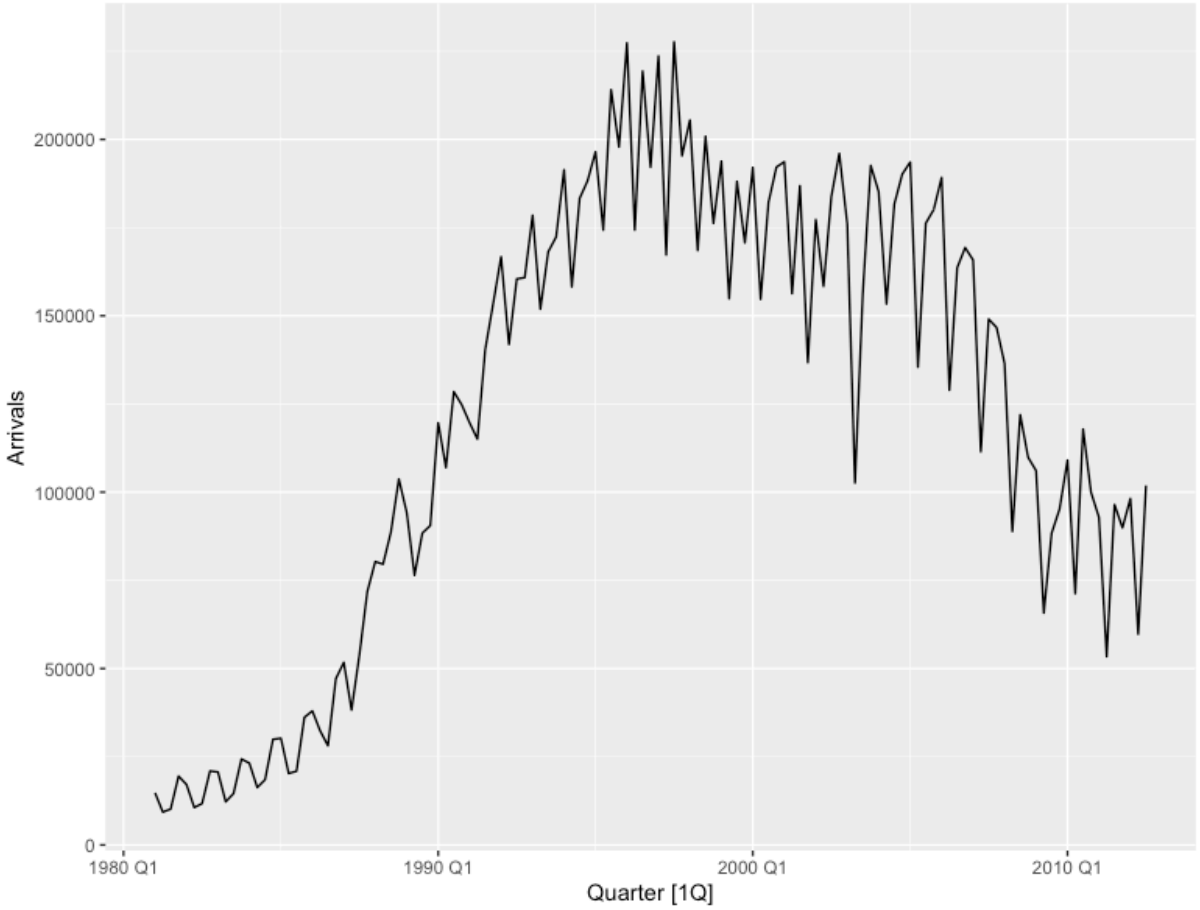
```
gdp %>%
  model(ARIMA(GDP ~ pdq(2,2,2))) %>%
  forecast(h = 10) %>%
  autoplot(gdp)
```



The forecast looks reasonable.

[9.a]
aus_arrivals_jp <- aus_arrivals %>% filter(Origin == "Japan")

aus_arrivals_jp %>% autoplot(Arrivals)

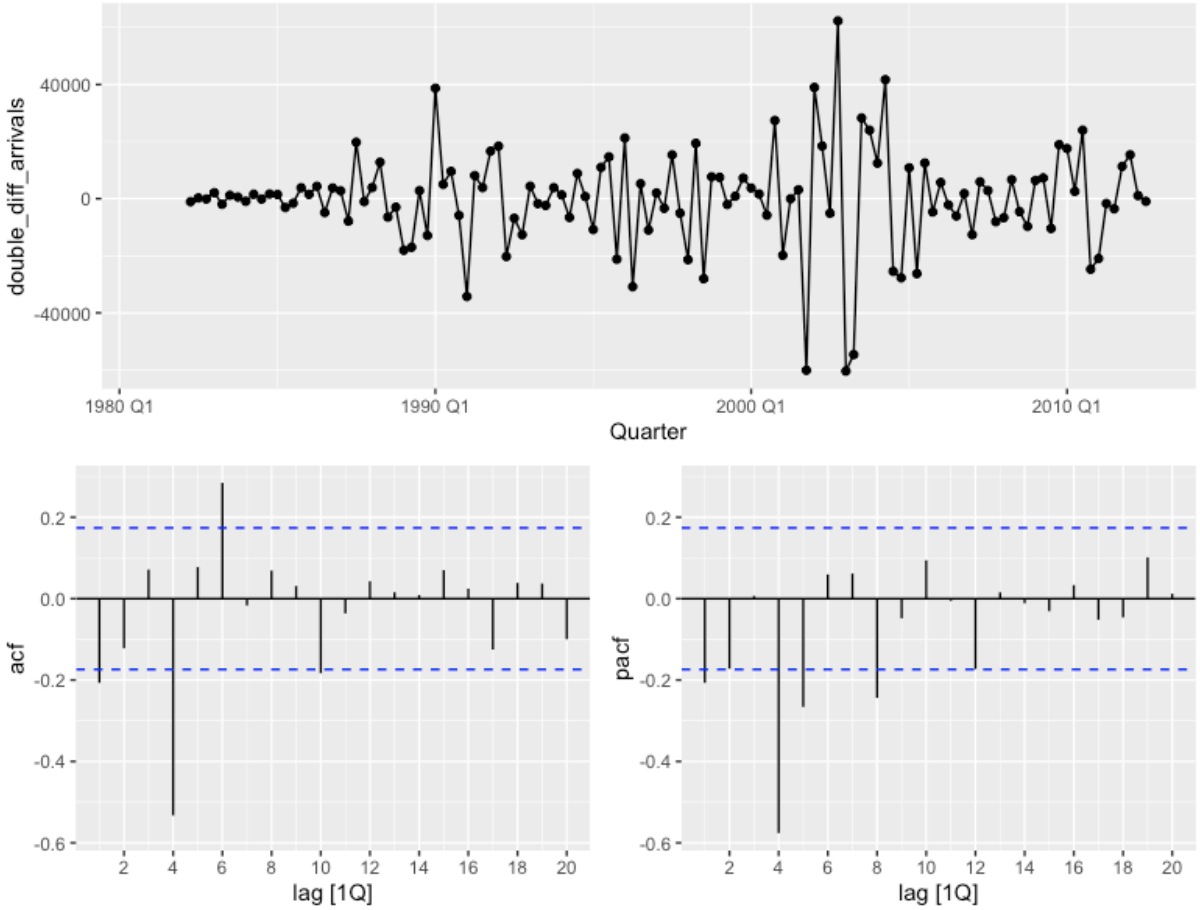


[9.b]

```
aus_arrivals_jp %>% features(Arrivals, unitroot_nsdiffs)

aus_arrivals_jp %>% features(Arrivals %>% difference(lag = 4), unitroot_ndiffs)

aus_arrivals_jp <- aus_arrivals_jp %>%
  mutate(
    double_diff_arrivals = difference(Arrivals, lag = 4) %>% difference(lag = 1)
  )
```

[9.c]

```
aus_arrivals_jp %>% gg_tsdisplay(double_diff_arrivals, plot_type = "partial")
```



There are spikes at lags 1, 4, 6 and 10, with the spikes at lag 4 being extreme.

[9.d]
There are decreasing spikes at multiples of 4. Additional spikes at lags 1 and 5.

[9.e]
Possible model = ARIMA(1,1,1)(1,1,1)[4]

[9.f]

```
fit <- aus_arrivals_jp %>%
  model(
    arima = ARIMA(Arrivals),
    arima_111111 = ARIMA(Arrivals ~ pdq(1,1,1) + PDQ(1,1,1))
  )

fit %>% glance() %>% arrange(AICc)
```

Both models have similar values, although the arima model is slightly better.

[9.g]
Backshift operator =
(1 − Φ[1]B[^4]) * (1 − B)(1 − B[^4]) * y[t] = (1 + θ[1]B) (1 + Θ[1]B[^4]) * ε[t]

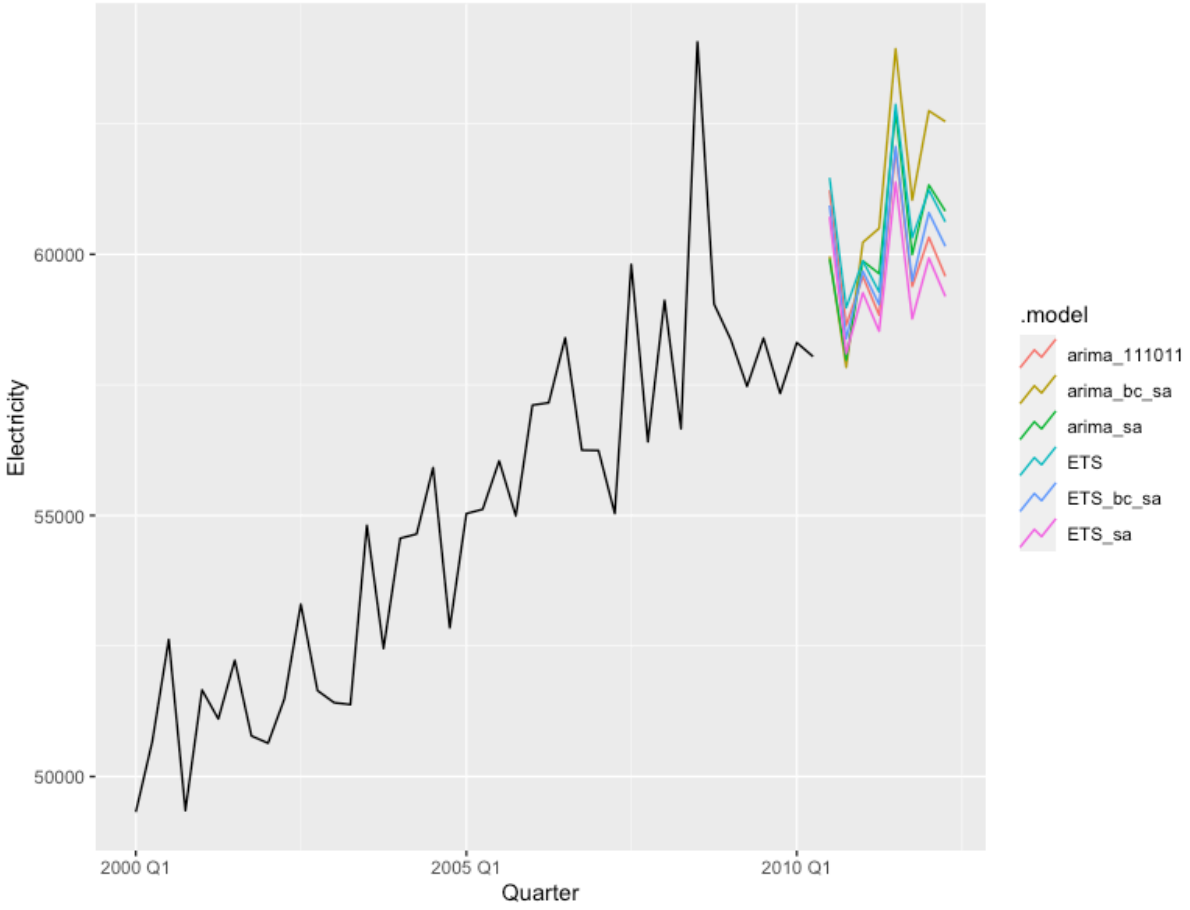
Without Backshift operator =
(1 − Φ[1]B[^4]) * (1 − B[^4] − B + B[^4]) * y[t] = (1 + θ[1]B + Θ[1]B[^4] + Θ[1][^2]B[^5]) * ε[t]

(1 − Φ[1]B[^4] − B[^4] + Φ[1]B[^8] − B + Φ[1]B[^5] + B[^4] - Φ[1]B[^8]) * y[t] = (1 + θ[1]B + Θ[1]B[^4] + Θ[1][^2]B[^5]) * ε[t]
(1 − Φ[1]B[^4] − B + Φ[1]B[^5]) * y[t] = (1 + θ[1]B + Θ[1]B[^4] + Θ[1][^2]B[^5]) * ε[t]

y[t] - Φ[1]y[t-4] - y[t-1] + Φ[1]y[t-5] = ε[t] + θ[1]ε[t-1] + Θ[1]ε[t-4] + Θ[1][^2]ε[t-5]
y[t] = y[t-1] + Φ[1](y[t-4] - y[t-5]) + ε[t] + θ[1](ε[t-1] + ε[t-4]) + Θ[1][^2]ε[t-5]

[12]

```
aus_electricity %>%
  model(
    arima_111011 = ARIMA(box_cox(Electricity, lambda) ~ pdq(1,1,1) + PDQ(0,1,1)),
    ETS = ETS(box_cox(Electricity, lambda)),
    arima_bc_sa = decomposition_model(
      STL(box_cox(Electricity, lambda)),
      ARIMA(season_adjust)
    ),
    ETS_bc_sa = decomposition_model(
      STL(box_cox(Electricity, lambda)),
      ETS(season_adjust)
    ),
    arima_sa = decomposition_model(
      STL(Electricity),
      ARIMA(season_adjust)
    ),
    ETS_sa = decomposition_model(
      STL(Electricity),
      ETS(season_adjust)
    )
  ) %>%
  forecast(h = "24 months") %>%
  autoplot(aus_electricity %>% filter(year(Quarter) >= 2000), level = NULL)
```



Except arima_bc_sa and ETS_sa the remaining models gives similar forecasts and all look reasonable. Non seasonally adjusted models seems to give a slightly better result, although either approach can work.

[13.a]

```
fit <- tourism %>%
  filter(Region %in% c("Snowy Mountains", "Melbourne")) %>%
  model(ARIMA(Trips))
```

[13.b]

```
fc <- fit %>% forecast(h = 12)
```

[13.c]

```
fc %>%
  autoplot(tourism) +
  facet_grid(Region ~ Purpose, scales = "free_y")
```

Snowy Mountains - Business, Other & Visiting have naive forecasts. Melbourne - Holiday & Other have MA(1) models and the forecasts are similar to naive forecasts. These do not capture any seasonal fluctuations, but the forecasts ranges look reasonable.The other forecasts look reasonable.

[14.a]

```
set.seed(123)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))
```

```
lambda <- myseries %>% features(Turnover, guerrero) %>% pull(lambda_guerrero)

myseries_train <- myseries %>% filter(year(Month) <= 2010)
myseries_test <- myseries %>% anti_join(myseries_train)

fit <- myseries_train %>%
  model(
    Arima = ARIMA(box_cox(Turnover, lambda)),
    SNaive = SNAIVE(Turnover),
    AM = ETS(Turnover ~ error() + trend("A") + season("M")),
    AdM = ETS(Turnover ~ error() + trend("Ad") + season("M"))
  )
```

[14.b]

```
fc <- fit %>% forecast(new_data = myseries_test)

fc %>% autoplot(myseries, level = NULL)

fc %>% accuracy(myseries) %>% arrange(RMSE)
```

ETS AM model is better than ARIMA model. However, ARIMA is much better than the AdM and SNaive models.

[14.c]

```
fc <- fit %>% forecast(h = "10 years")

fc %>%
  hilo(level = 95) %>%
  select(.model, Month, .mean, "95%") %>%
  filter(year(Month) == 2019)
```

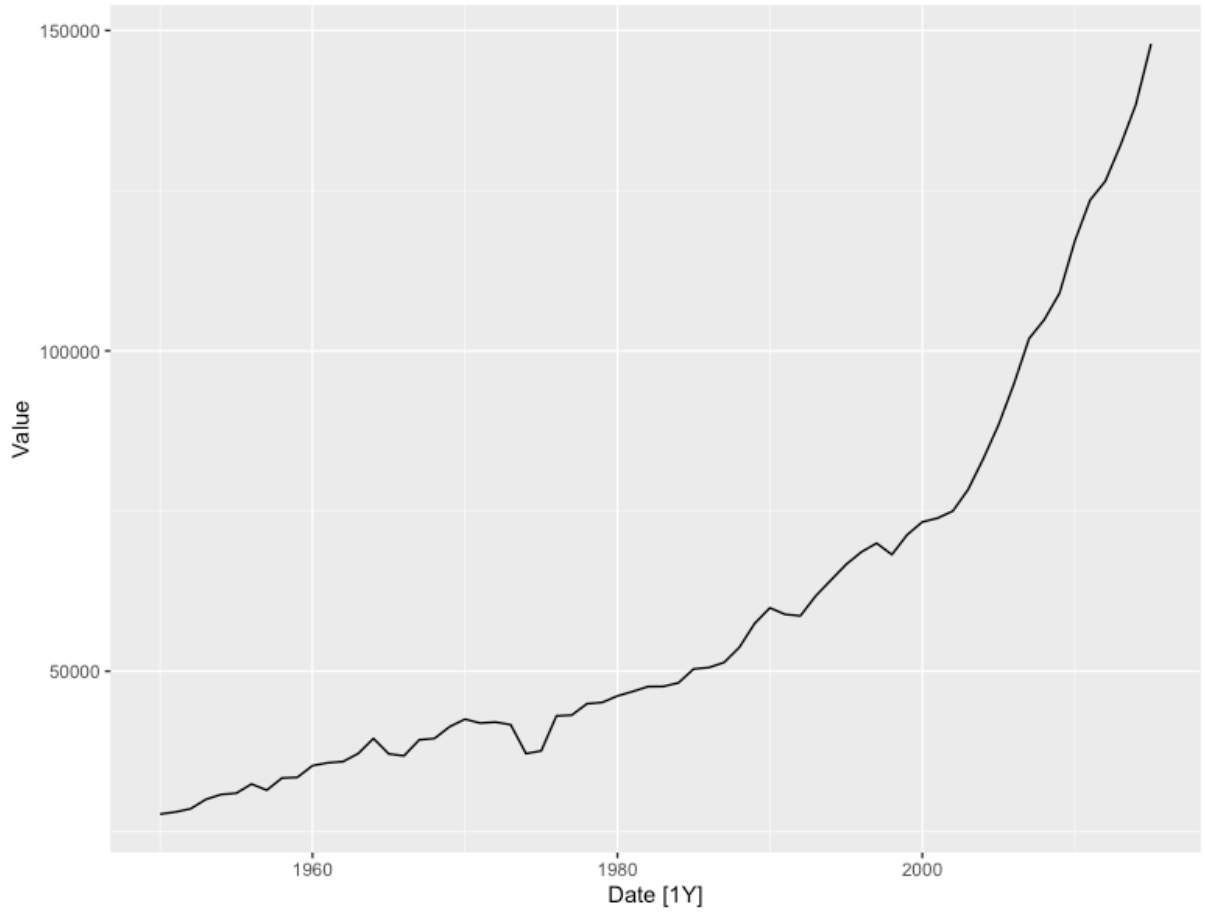
The RMSE for year 2019 = 115.6, which is higher than for test set (87). The forecasts are all within the 95% confidence interval. However, considering that the model has built using data till 2010, this error is reasonable and it shows that the model is still good even after nearly a decade.

[17.a]

```
y <- as_tsibble(Quandl::Quandl("PSE/ANNINC992I_PALL_IN"))

y <- y %>% mutate(Date = year(Date))
```

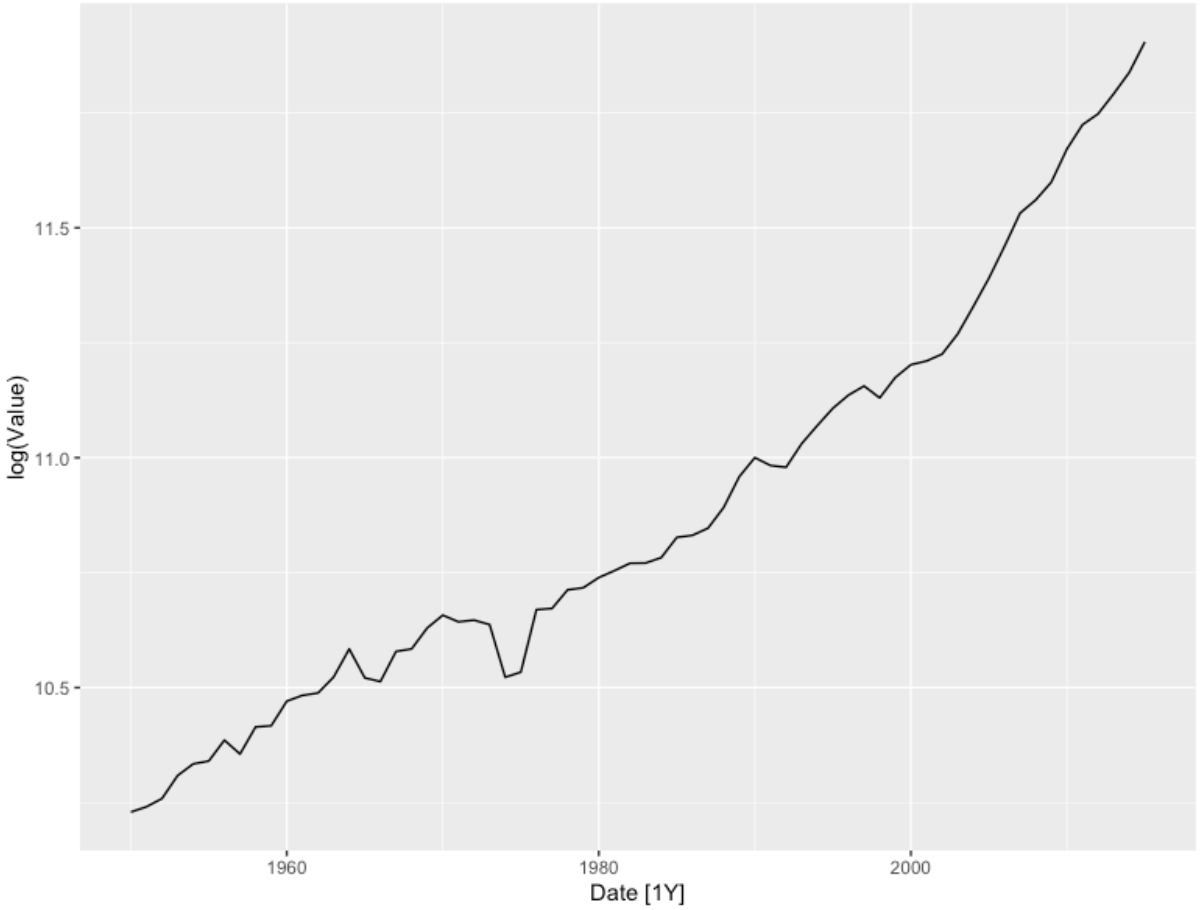
[17.b]

```
y %>% autoplot(Value)
```

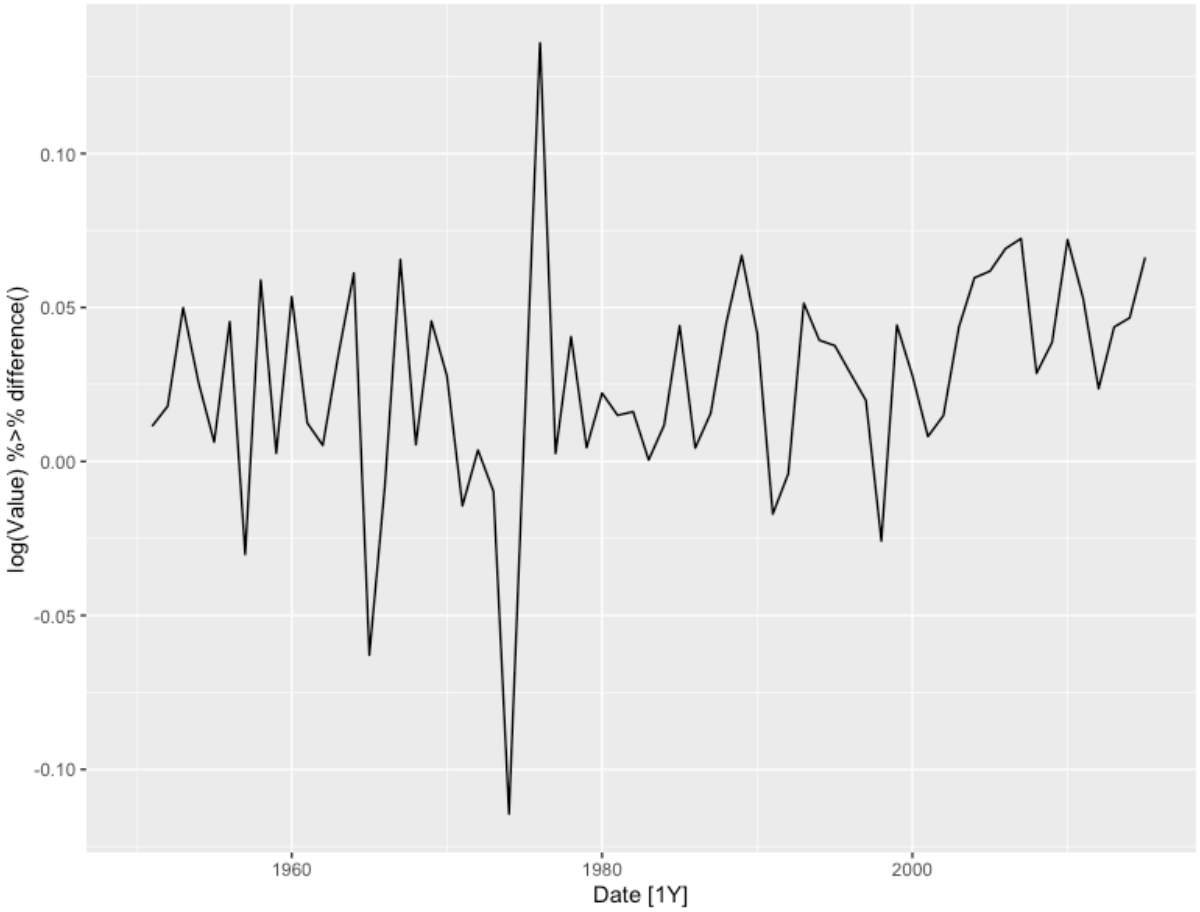


This is annual data, with increasing trend, which exponentially increases after 2000. Log transform might help stabalize the series. Also, first difference might be required.

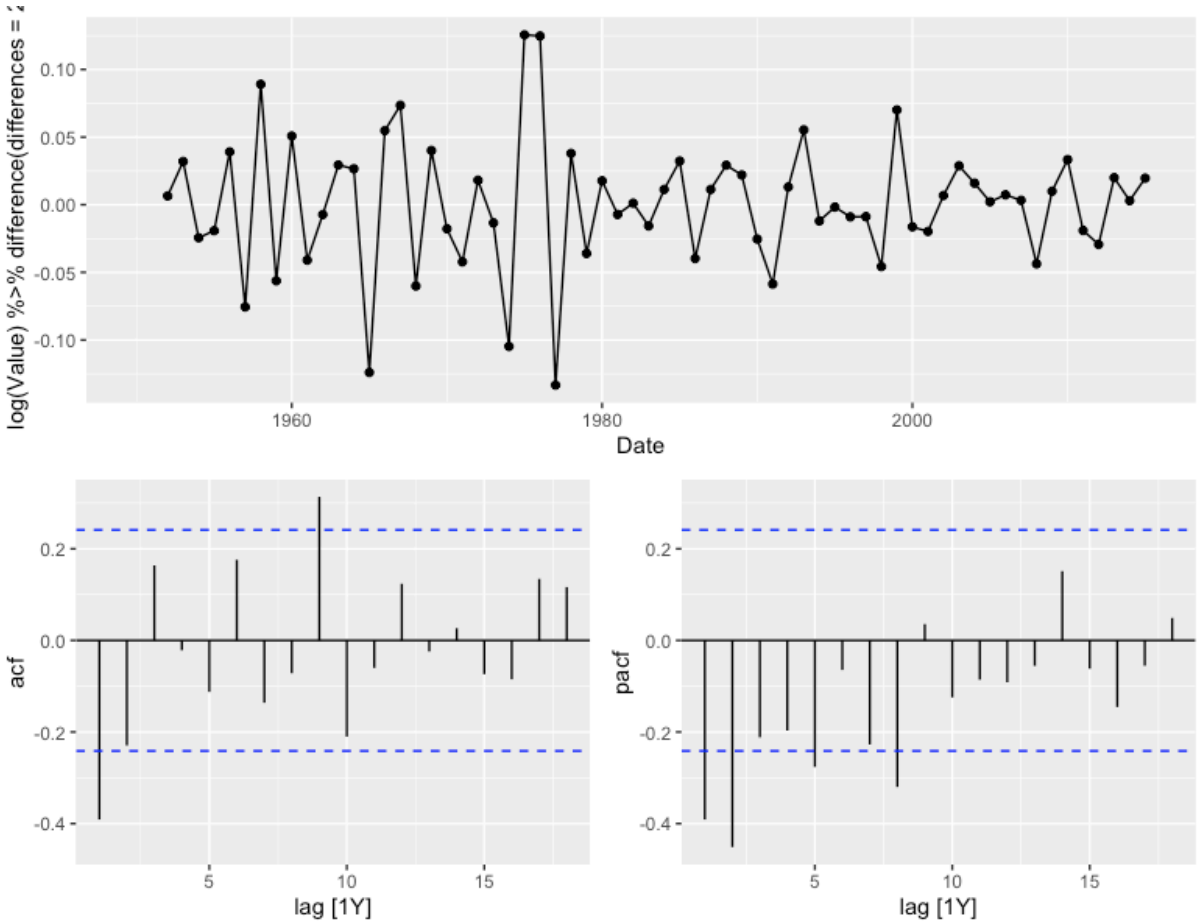
y %>% autoplot(log(Value))



y %>% autoplot(log(Value) %>% difference())



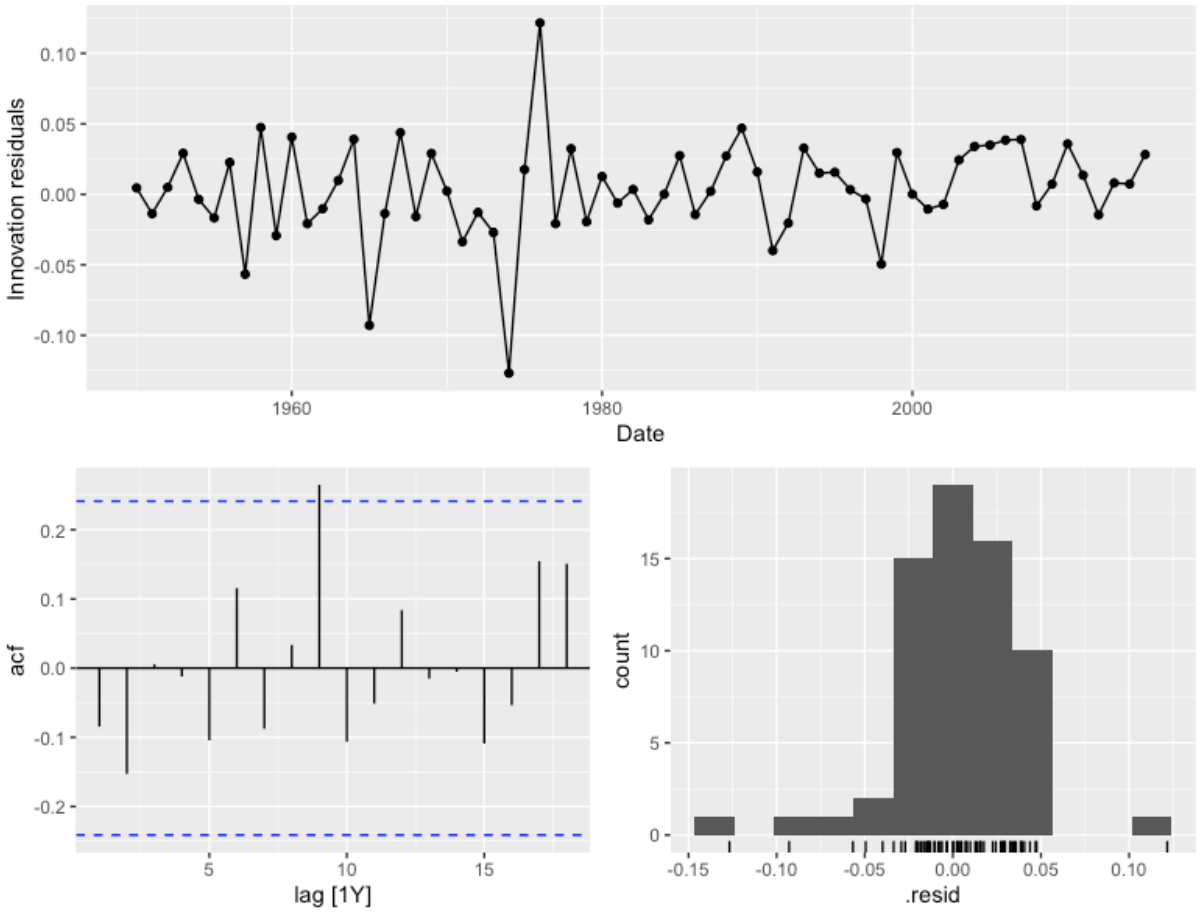
```
y %>% features(log(Value), unitroot_ndiffs)

y %>%
  gg_tsdisplay(log(Value) %>% difference(differences = 2), plot_type = "partial")
```



The PACF has a spike at lag 2 followed by a small spike at lag 5 and 8. A MA(2) might work.
The ACF has a spike at lag 1 followed by a spike at lag 9. An AR(1) might work.
Model - ARIMA(1,2,2)

[17.c]

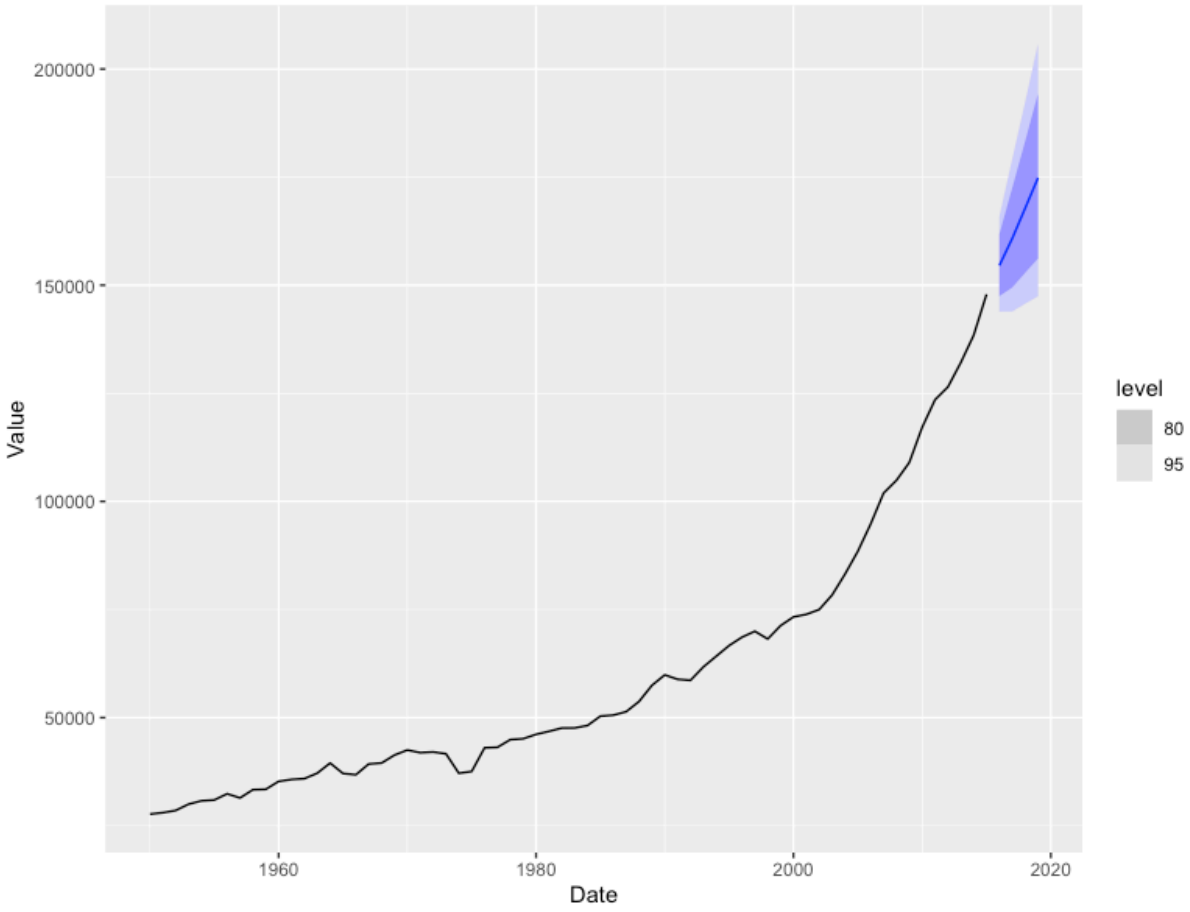
```
fit <- y %>% model(ARIMA(log(Value) ~ pdq(1,2,2)))

fit %>% gg_tsresiduals()
```



augment(fit) %>% features(.innov, ljung_box, lag = 10, dof = 3)

[17.d]

```
fit %>%
  forecast(h = 4) %>%
  autoplot(y)
```


200000
150000
100000
50000
Value
1960
1980
2000
2020
Date
level
80
95