Data 624 Homework 6

Warner Alexis

2025-03-23

ARIMA MODELS

Excercise 9.1

Figure 9.32 shows the ACFs for 36 random numbers, 360 random numbers and 1,000 random numbers.

Explain the differences among these figures. Do they all indicate that the data are white noise?

Let’s go through the questions one by one based on **Figure 9.32**.

1a. Explain the differences among these figures. Do they all indicate that the data are white noise?

Differences among the figures: - The **left graph (n = 36)** shows a lot of variation in the autocorrelations (ACFs), with several spikes outside the blue significance bounds. - The **middle graph (n = 360)** shows less variability, and almost all autocorrelation values fall within the blue bounds. - The **right graph (n = 1,000)** shows even less variability and very tight confidence bands; the ACF values stay very close to zero.

Do they all indicate white noise? - Yes, all three plots represent white noise series. The differences in appearance are due to sample size. - In the left graph, the smaller sample size (n=36) causes more random variation, which results in some autocorrelation values falling outside the confidence bounds just by chance. - As the sample size increases (middle and right plots), the estimates of the autocorrelations become more precise, and the values stay within the expected range for white noise.

1b. Why are the critical values at different distances from the mean of zero? Why are the autocorrelations different in each figure when they each refer to white noise?

Critical values and sample size: - The blue dashed lines represent the 95% confidence interval, and for white noise, they are approximately $\pm 1.96/\sqrt{n}$. - As the sample size increases, the critical values get closer to zero because $\sqrt{n}$ increases, reducing the standard error of the autocorrelation estimate. - For n = 36: $\pm 1.96/\sqrt{36} \approx \pm 0.33$ - For n = 360: $\pm 1.96/\sqrt{360} \approx \pm 0.10$ - For n = 1,000: $\pm 1.96/\sqrt{1000} \approx \pm 0.06$

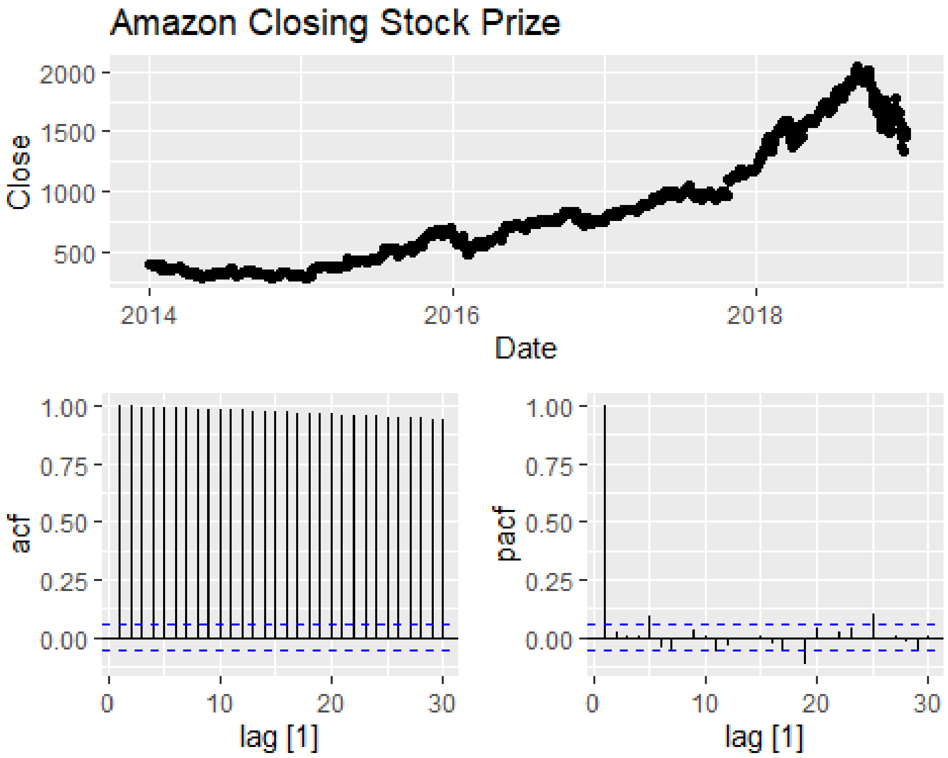
Autocorrelations are different in each figure because: - With smaller samples, random fluctuations appear more prominently, making it look like there's more autocorrelation. - Larger samples provide more accurate estimates of the true autocorrelation (which is zero for white noise), so the plots for larger n show values closer to zero.

Excercise 9.2 A classic example of a non-stationary series are stock prices. Plot the daily closing prices for Amazon stock (contained in gafa_stock), along with the ACF and PACF. Explain how each plot shows that the series is non-stationary and should be differenced.

The time series plot of Amazon's daily closing prices reveals a clear upward trend over time, indicating that the series is non-stationary. This trend suggests that both the mean and variance are not constant, which violates the assumptions of stationarity. The ACF plot further supports this conclusion, showing a slow decay and high correlations across many lags—behavior typical of non-stationary data. In a stationary series, the ACF would drop off quickly after a few lags. Similarly, the PACF plot does not exhibit a sharp cutoff, but instead displays significant partial autocorrelations over several lags, reinforcing the need to difference the series to achieve stationarity.

LOading Libraries

```
library(latex2exp)
library(fpp3)

## Registered S3 method overwritten by 'tsibble':
##   method               from 
##   as_tibble.grouped_df dplyr

## ── Attaching packages ──────────────────────────────────────────── fpp3
1.0.1 ──

## ✓ tibble      3.2.1     ✓ tsibble     1.1.6
## ✓ dplyr       1.1.4     ✓ tsibbledata 0.4.1
## ✓ tidyr       1.3.1     ✓ feasts      0.4.1
## ✓ lubridate   1.9.3     ✓ fable       0.4.1
## ✓ ggplot2     3.5.1

## ── Conflicts ─────────────────────────────────────────────────
fpp3_conflicts ──
## ✕ lubridate::date()    masks base::date()
## ✕ dplyr::filter()      masks stats::filter()
## ✕ tsibble::intersect() masks base::intersect()
## ✕ tsibble::interval()  masks lubridate::interval()
## ✕ dplyr::lag()         masks stats::lag()
## ✕ tsibble::setdiff()   masks base::setdiff()
## ✕ tsibble::union()     masks base::union()

library(fable)
library(tsibble)
library(tsibbledata)
library(latex2exp)
gafa_stock |> filter(Symbol == 'AMZN') |>
  gg_tsdisplay(Close, plot_type =  'partial') +
  labs(title = 'Amazon Closing Stock Prize ' )
```

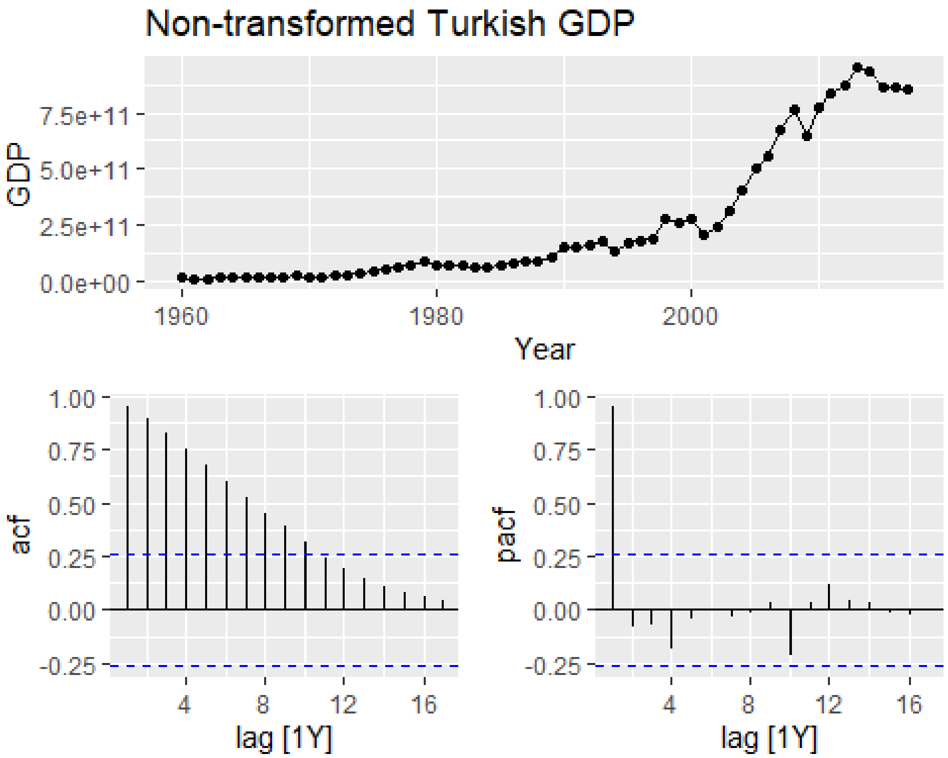


Excercise 9.3 For the following series, find an appropriate Box-Cox transformation and order of differencing in order to obtain stationary data.

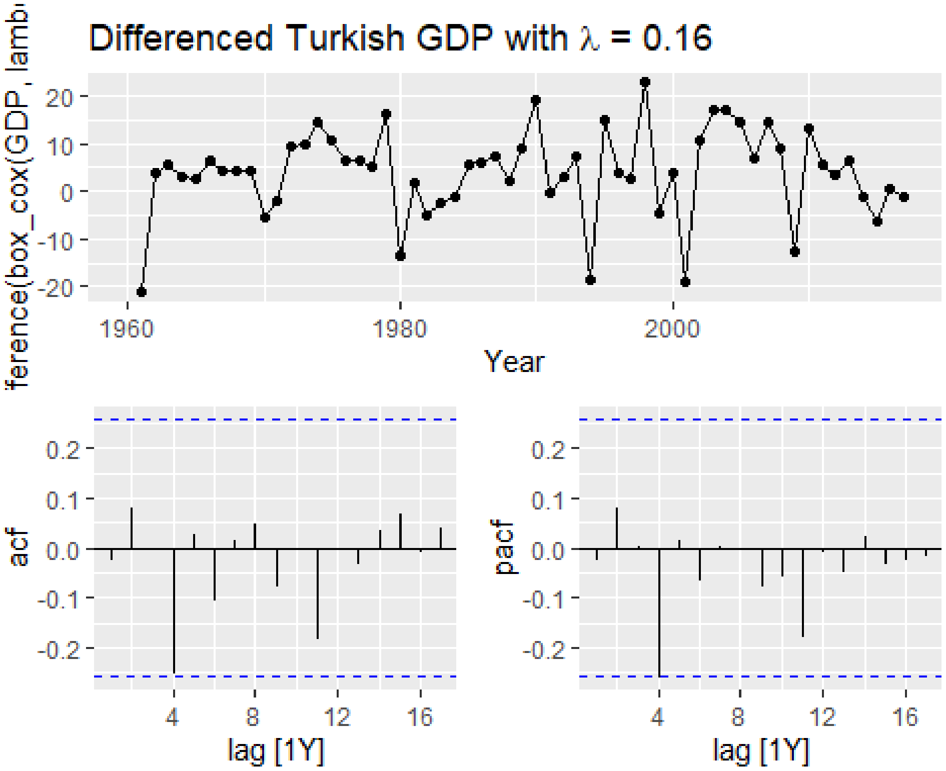
- Turkish GDP from global_economy.
- Accommodation takings in the state of Tasmania from aus_accommodation.
- Monthly sales from souvenirs.

The first set of plots shows the non-transformed Turkish GDP, which displays a strong upward trend over time, indicating clear non-stationarity. This is supported by the autocorrelation function (ACF) plot, which exhibits a slow and steady decay, and the partial autocorrelation function (PACF), which shows a large spike at lag 1 followed by smaller significant lags—both of which are typical signs of a non-stationary time series. In contrast, the second set of plots presents the Turkish GDP after applying a Box-Cox transformation with λ = 0.16 and first-order differencing. The transformed and differenced series no longer shows a trend and appears to fluctuate around a constant mean. Additionally, both the ACF and PACF now show values mostly within the significance bounds and no clear pattern, indicating that the series has achieved stationarity and is now suitable for time series modeling, such as ARIMA.

```
turkey_gdp <- # plot
  global_economy %>%
  filter(Country == "Turkey")
# plot
turkey_gdp %>%
  gg_tsdisplay(GDP, plot_type='partial') +
  labs(title = "Non-transformed Turkish GDP")
```

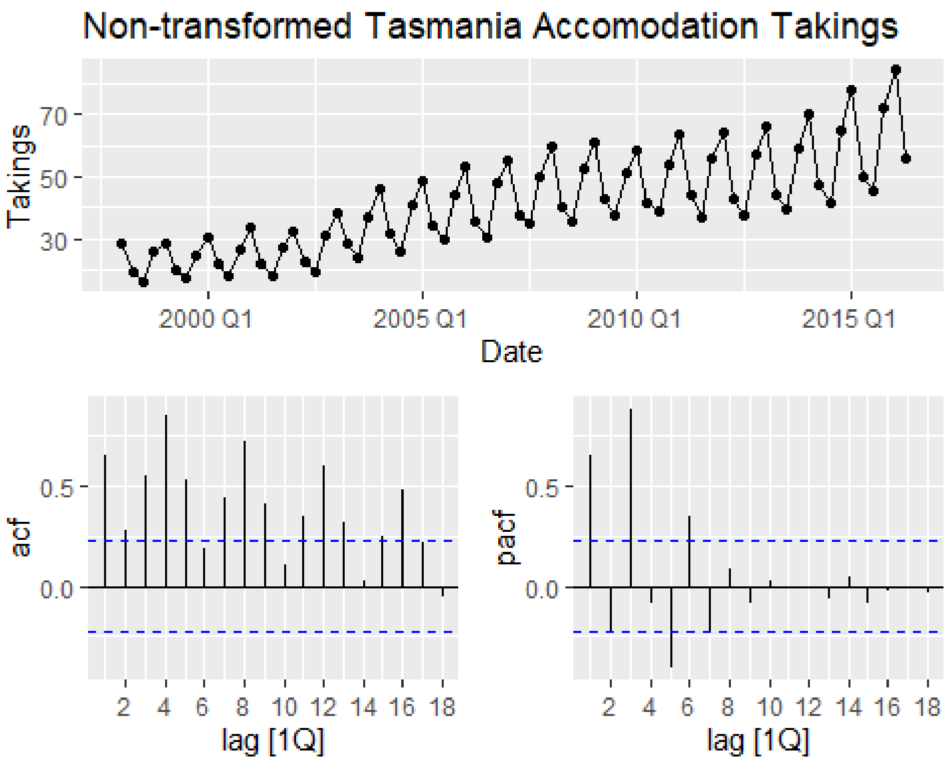


```
# calculate lambda
lambda <- turkey_gdp %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)


turkey_gdp %>%
  features(box_cox(GDP,lambda), unitroot_ndiffs)

## # A tibble: 1 × 2
##   Country ndiffs
##   <fct>    <int>
## 1 Turkey       1

# unit root test


# transformed plot
turkey_gdp %>%
  gg_tsdisplay(difference(box_cox(GDP,lambda)), plot_type='partial') +
  labs(title = TeX(paste0("Differenced Turkish GDP with $\\lambda$ = ",
                          round(lambda,2))))
```

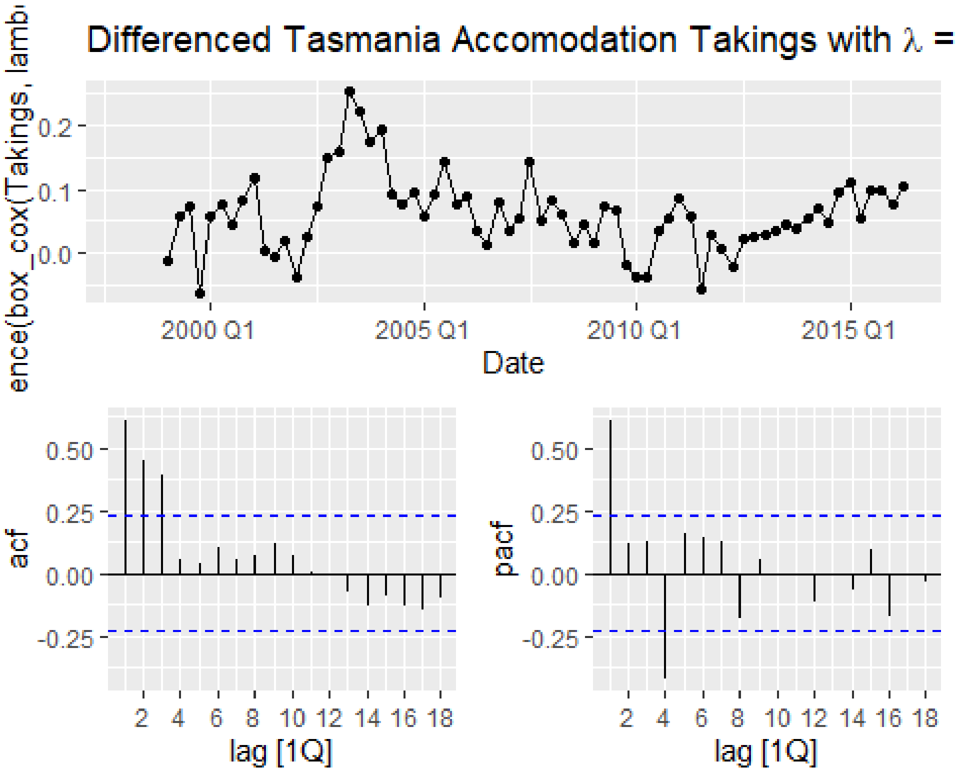


The first set of plots displays the non-transformed Tasmania accommodation takings, which exhibit a clear upward trend and strong seasonal pattern, both indicators of a non-stationary series. The autocorrelation function (ACF) plot shows significant spikes at seasonal lags (e.g., lag 4, 8, 12, etc.), indicating the presence of quarterly seasonality, while the partial autocorrelation function (PACF) also reflects persistent seasonal effects. The second set of plots shows the same series after applying a Box-Cox transformation with λ = 0 (log transformation) and seasonal differencing. The differenced and transformed series appears more stable, with the trend removed and the seasonal pattern largely eliminated. This is confirmed by the ACF and PACF plots, which now show autocorrelations that fall within the confidence bounds or decay more quickly, indicating that the series is now approximately stationary and suitable for further time series modeling.

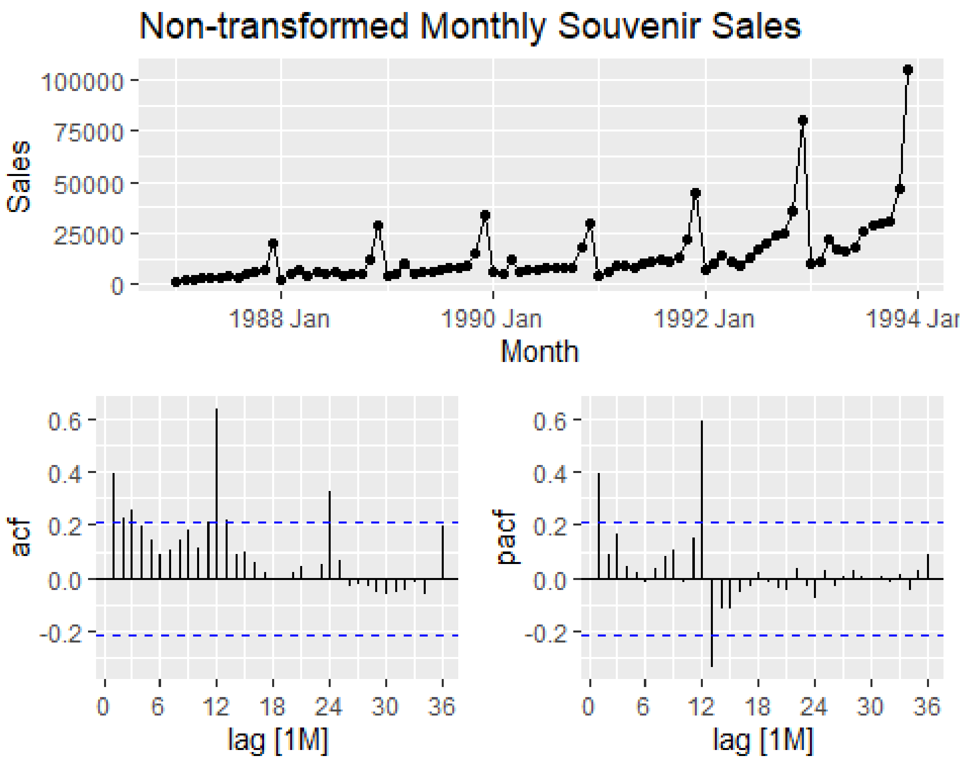
```
tasma_acc <-
  aus_accommodation %>%
  filter(State == "Tasmania")


tasma_acc %>%
  gg_tsdisplay(Takings, plot_type='partial') +
  labs(title = "Non-transformed Tasmania Accomodation Takings")
```



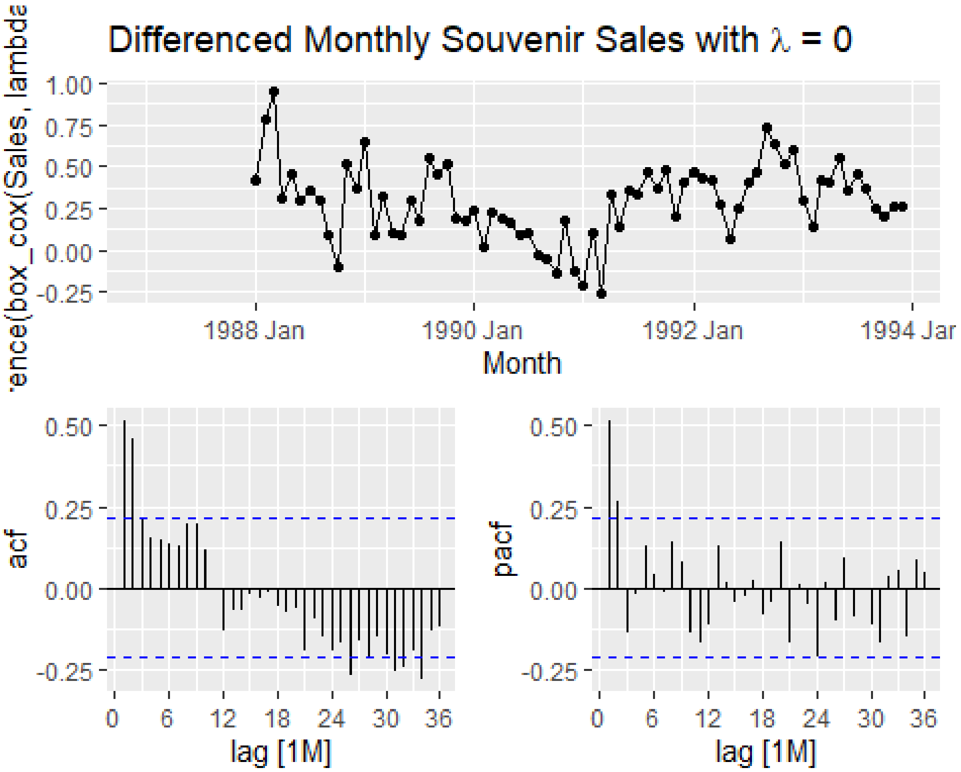
```
# calculate lambda
lambda <-tasma_acc %>%
  features(Takings, features = guerrero) %>%
  pull(lambda_guerrero)

#unit root test
tasma_acc %>%
  features(box_cox(Takings,lambda), unitroot_nsdiffs)

## # A tibble: 1 × 2
##   State    nsdiffs
##   <chr>      <int>
## 1 Tasmania       1

tasma_acc %>%
  gg_tsdisplay(difference(box_cox(Takings,lambda), 4), plot_type='partial') +
  labs(title = TeX(paste0("Differenced Tasmania Accomodation Takings with
$\\lambda$ = ",
                        round(lambda,2))))

## Warning: Removed 4 rows containing missing values or values outside the
scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the
scale range
## (`geom_point()`).
```



```
# plot
souvenirs %>%
  gg_tsdisplay(Sales, plot_type='partial', lag = 36) +
  labs(title = "Non-transformed Monthly Souvenir Sales")
```



```
# calculate lambda
lambda <- souvenirs %>%
  features(Sales, features = guerrero) %>%
  pull(lambda_guerrero)

# unit root test
souvenirs %>%
  features(box_cox(Sales,lambda), unitroot_nsdiffs)

## # A tibble: 1 × 1
##   nsdiffs
##     <int>
## 1       1

souvenirs %>%
  gg_tsdisplay(difference(box_cox(Sales,lambda), 12), plot_type='partial',
lag = 36) +
  labs(title = TeX(paste0("Differenced Monthly Souvenir Sales with $\\lambda$
= ",
                          round(lambda,2))))
```

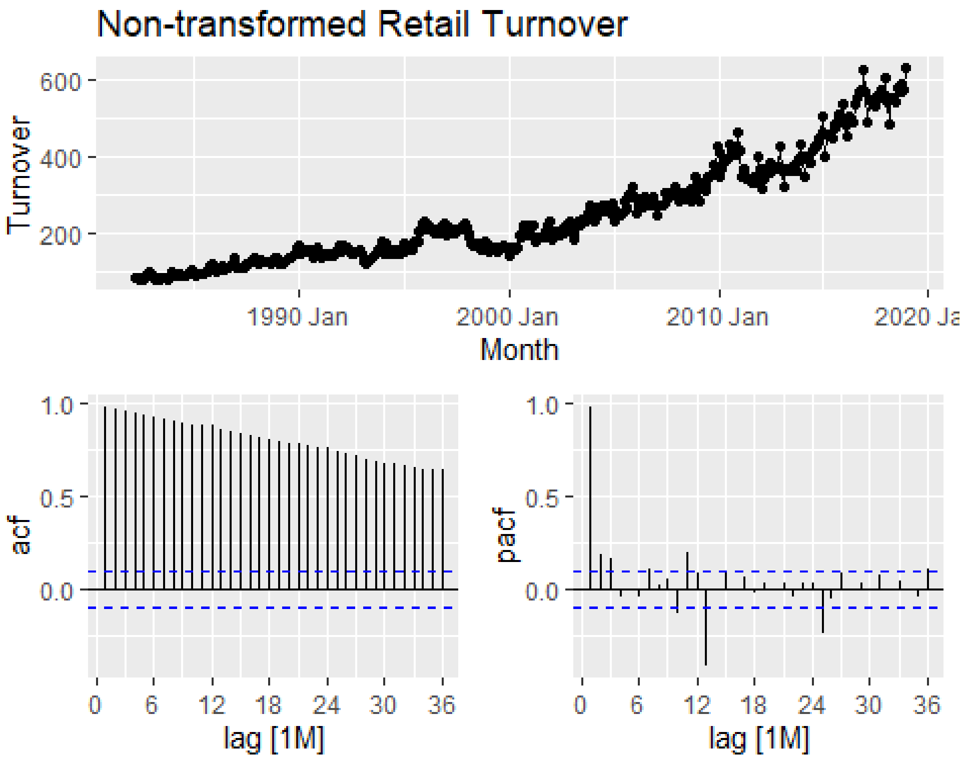


Excercise 9.5 For your retail data (from Exercise 7 in Section 2.10), find the appropriate order of differencing (after transformation if necessary) to obtain stationary data.

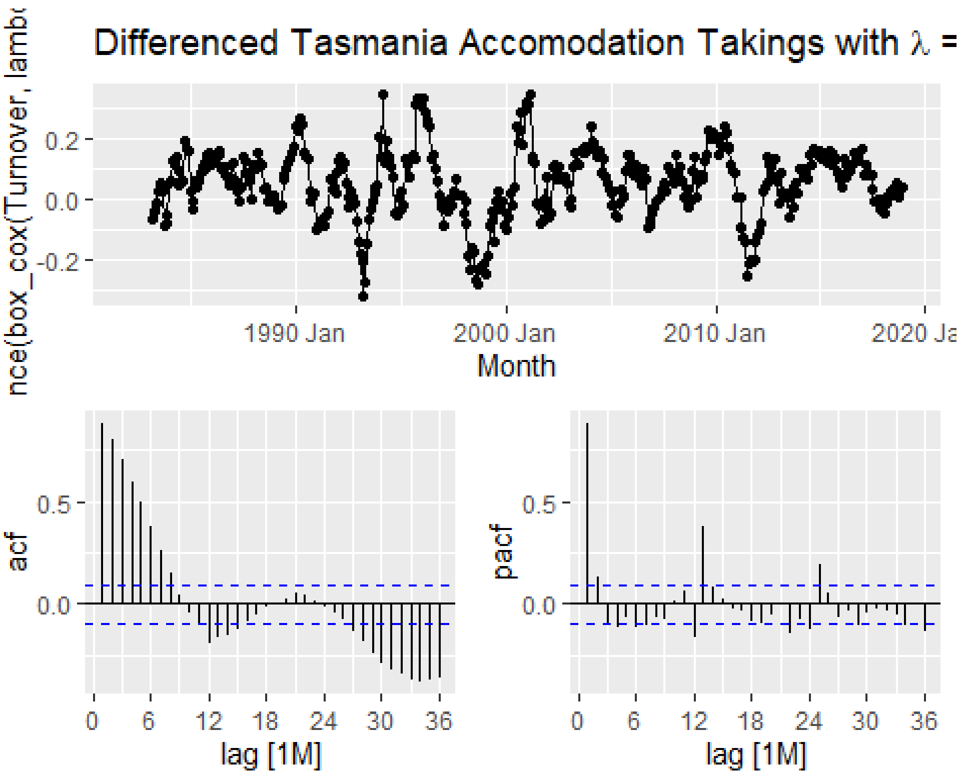
The first set of plots presents the non-transformed retail turnover data for Tasmania, which displays a strong upward trend and increasing variance over time—both signs of non-stationarity. The ACF shows very slow decay, with significant autocorrelations extending across many lags, while the PACF shows a large spike at lag 1 followed by smaller but still significant values. These patterns confirm that the series is non-stationary in both mean and variance. In the second set of plots, the data have been transformed using a Box-Cox transformation with λ = 0.27 and differenced once. The resulting series appears more stable, with the trend largely removed and the variance stabilized. The ACF now shows a much quicker decay, and the PACF displays a reduced number of significant lags. Although some seasonality may remain, the primary trend has been addressed.

- **Box-Cox transformation** with **λ = 0.27** is appropriate to stabilize variance.
- **First-order differencing** is sufficient to remove the trend and achieve approximate stationarity.
- Additional **seasonal differencing** (e.g., lag 12) could be explored if residual seasonality is detected during modeling.

```
set.seed(000)
myseries <- aus_retail %>%
  filter(`Series ID` == sample(aus_retail$`Series ID`,1))

# plot
myseries %>%
  gg_tsdisplay(Turnover, plot_type='partial', lag = 36) +
  labs(title = "Non-transformed Retail Turnover")
```



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

# unit root test
myseries %>%
  features(box_cox(Turnover, lambda), unitroot_nsdiffs)

## # A tibble: 1 × 3
##   State           Industry                nsdiffs
##   <chr>           <chr>                     <int>
## 1 New South Wales Takeaway food services        1

myseries %>%
  gg_tsdisplay(difference(box_cox(Turnover,lambda), 12), plot_type='partial',
lag = 36) +
  labs(title = TeX(paste0("Differenced Tasmania Accomodation Takings with
$\\lambda$ = ",
                          round(lambda,2))))
```



Excercise 9.6

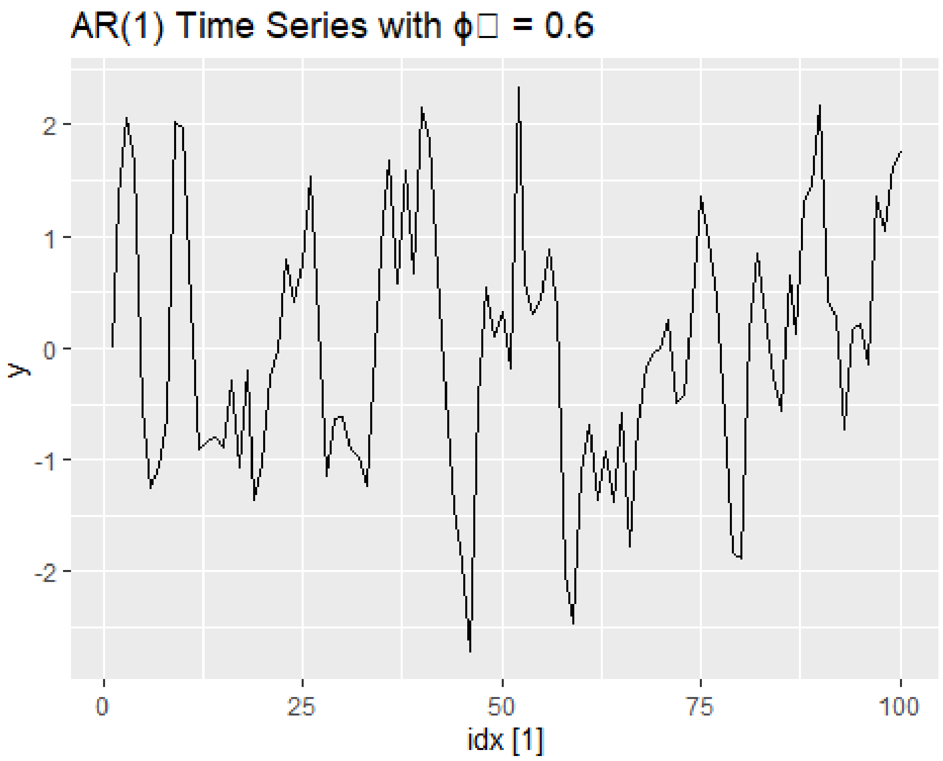
Simulate and plot some data from simple ARIMA models.

Use the following R code to generate data from an AR(1) model with

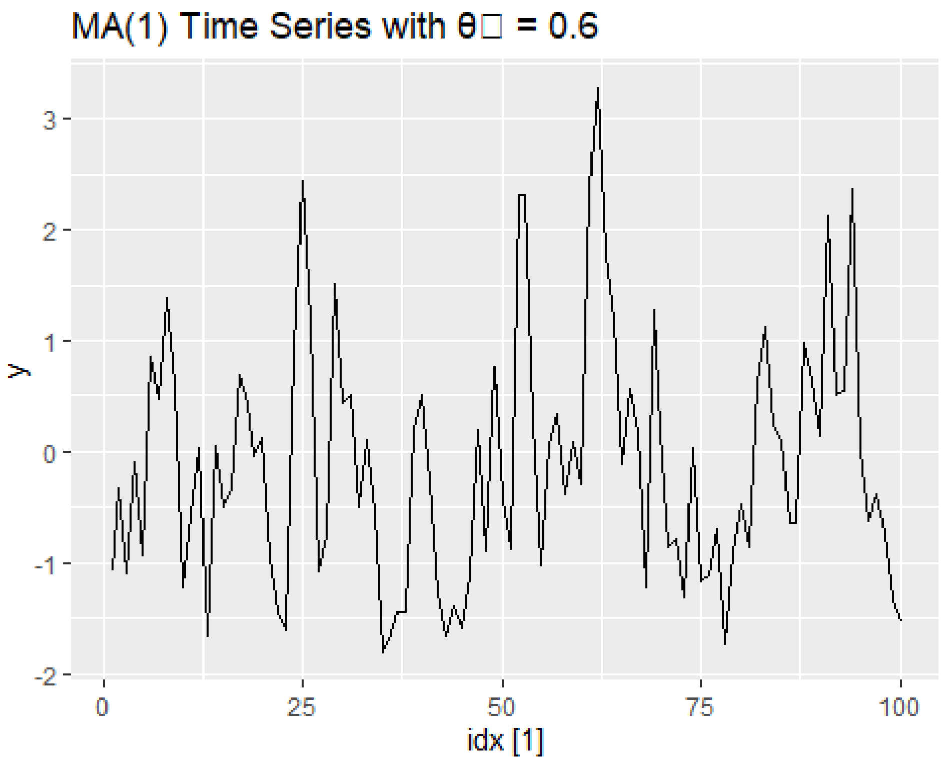
This exercise explores the behavior of various ARIMA models through simulation and visualization. Starting with an AR(1) model with $\phi_1 = 0.6$, the series shows moderate persistence, and increasing ϕ_1 leads to smoother, more correlated patterns, while lower values make the series resemble white noise. The MA(1) model with $\theta_1 = 0.6$ generates a series influenced by past shocks, though the effect dissipates quickly compared to AR models. Combining both effects, the ARMA(1,1) model demonstrates smoother yet still responsive behavior, balancing the autocorrelation from AR and the noise filtering of MA. A non-stationary AR(2) model with $\phi_1 = -0.8$ and $\phi_2 = 0.3$ produces a series with irregular and unstable dynamics, often with large swings and no consistent mean. Comparing the ARMA(1,1) and AR(2) plots highlights the contrast between a stable, stationary process and one that lacks mean reversion, emphasizing the importance of parameter selection in time series modeling.

```r
# 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)
```

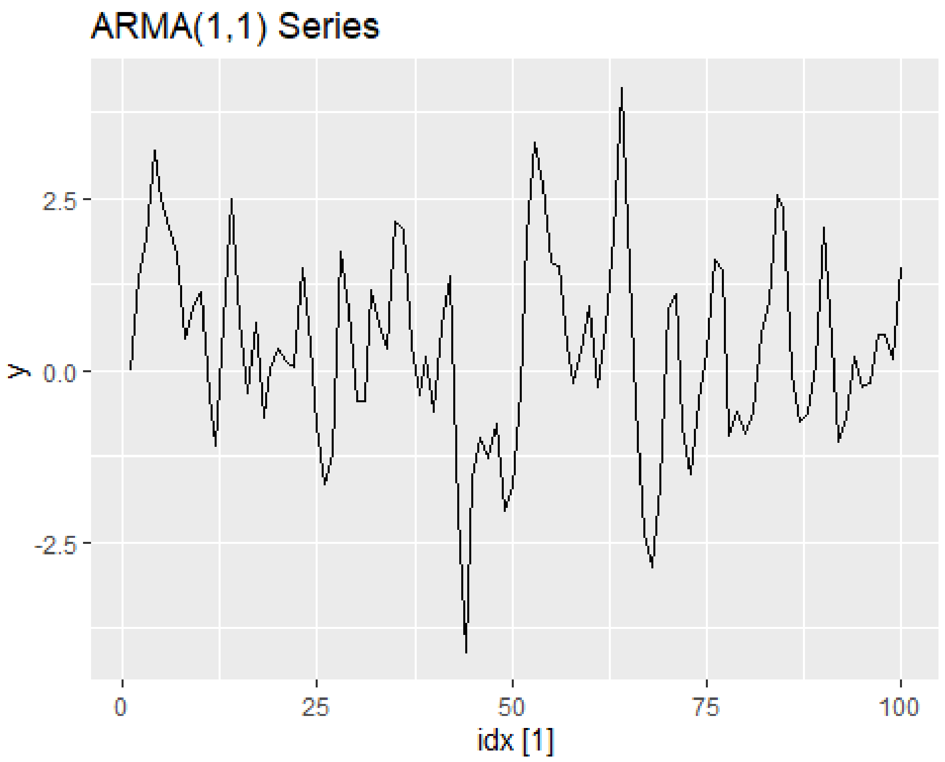
```
# b
sim %>%
  autoplot(y) +
  labs(title = "AR(1) Time Series with φ₁ = 0.6", y = "y")
```



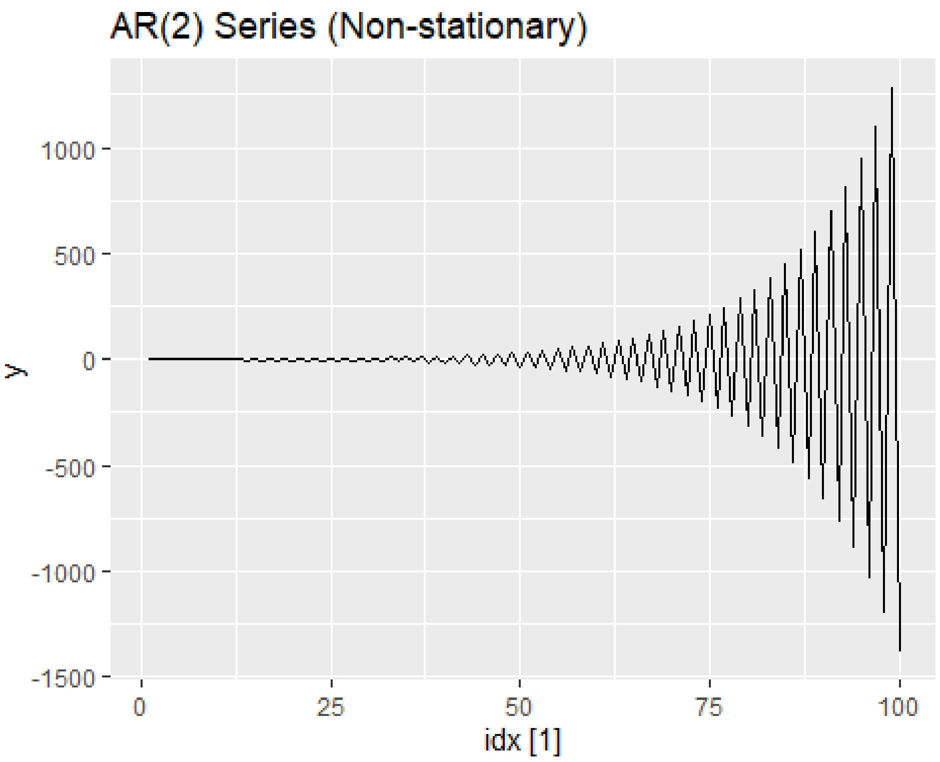
```
# c
e <- rnorm(101)
y <- numeric(100)
for(i in 1:100){
  y[i] <- e[i+1] + 0.6 * e[i]
}
sim_ma <- tsibble(idx = seq_len(100), y = y, index = idx)

#d
sim_ma %>%
  autoplot(y) +
  labs(title = "MA(1) Time Series with θ₁ = 0.6", y = "y")
```



```
#e
e <- rnorm(101)
y <- numeric(100)
for(i in 2:100){
  y[i] <- 0.6 * y[i - 1] + e[i] + 0.6 * e[i - 1]
}
sim_arma <- tsibble(idx = seq_len(100), y = y, index = idx)
autoplot(sim_arma, y) + labs(title = "ARMA(1,1) Series", y = "y")
```



```r
# f
e <- rnorm(102)
y <- numeric(100)
for(i in 3:100){
  y[i] <- -0.8 * y[i - 1] + 0.3 * y[i - 2] + e[i]
}
sim_ar2 <- tsibble(idx = seq_len(100), y = y, index = idx)
autoplot(sim_ar2, y) + labs(title = "AR(2) Series (Non-stationary)", y = "y")
```



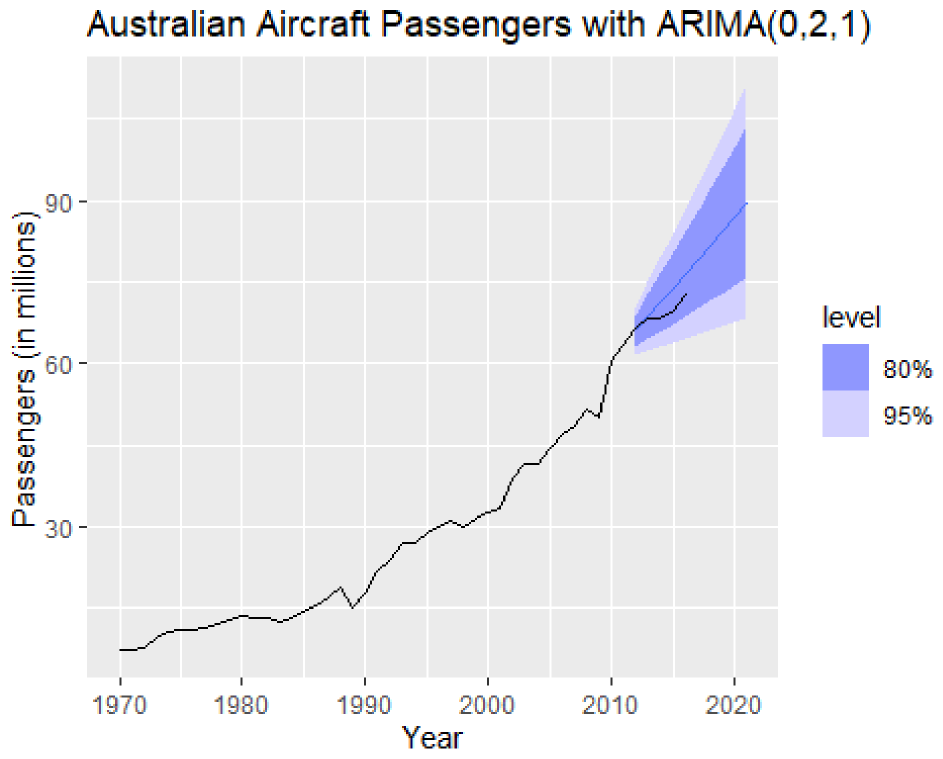
Excercise 9.7

The model can be written in term the baskshift operator

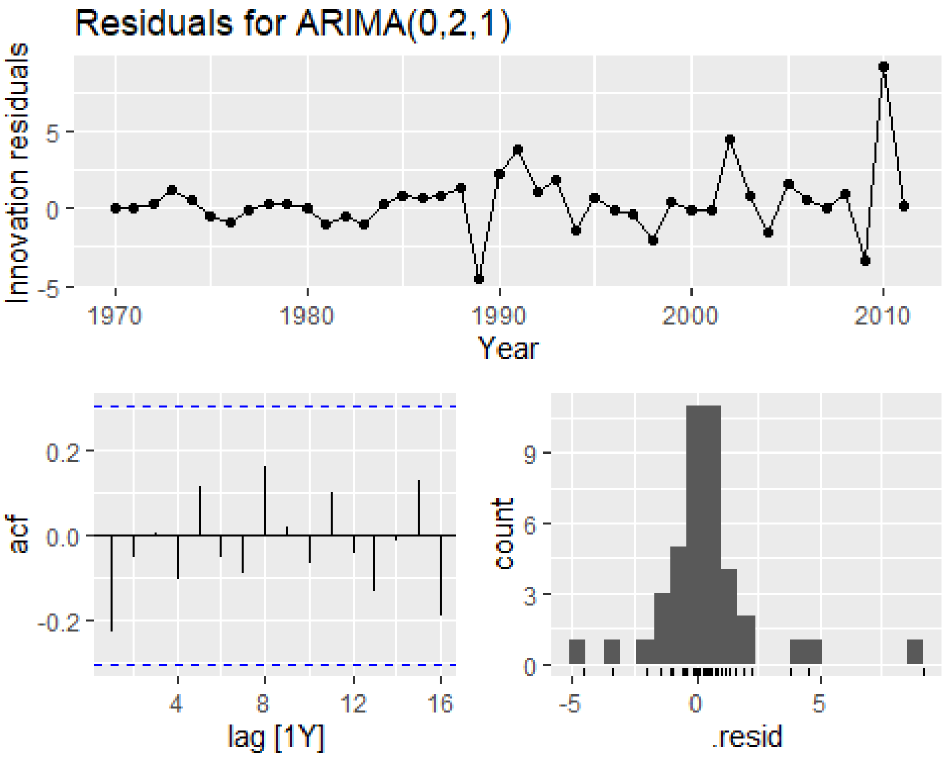
$$y_t = -0.87\varepsilon_{t-1} + \varepsilon_t$$

$$(1 - B)^2 y_t = (1 - 0.87B)\varepsilon_t$$

```
fit <- aus_airpassengers %>%
  filter(Year < 2012) %>%
  model(ARIMA(Passengers))

report(fit)

## Series: Passengers
## Model: ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.8756
## s.e.   0.0722
##
## sigma^2 estimated as 4.671:  log likelihood=-87.8
## AIC=179.61   AICc=179.93   BIC=182.99

fit %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "Australian Aircraft Passengers with ARIMA(0,2,1)", y =
"Passengers (in millions)")
```



```
fit %>%
  gg_tsresiduals() +
  labs(title = "Residuals for ARIMA(0,2,1)")
```

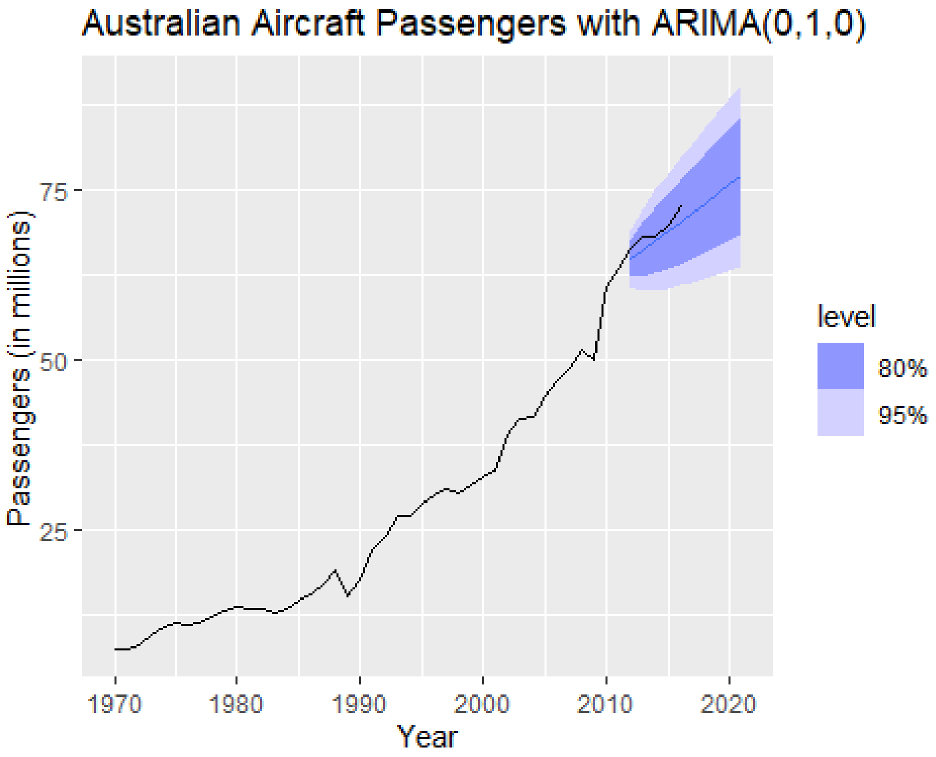


-

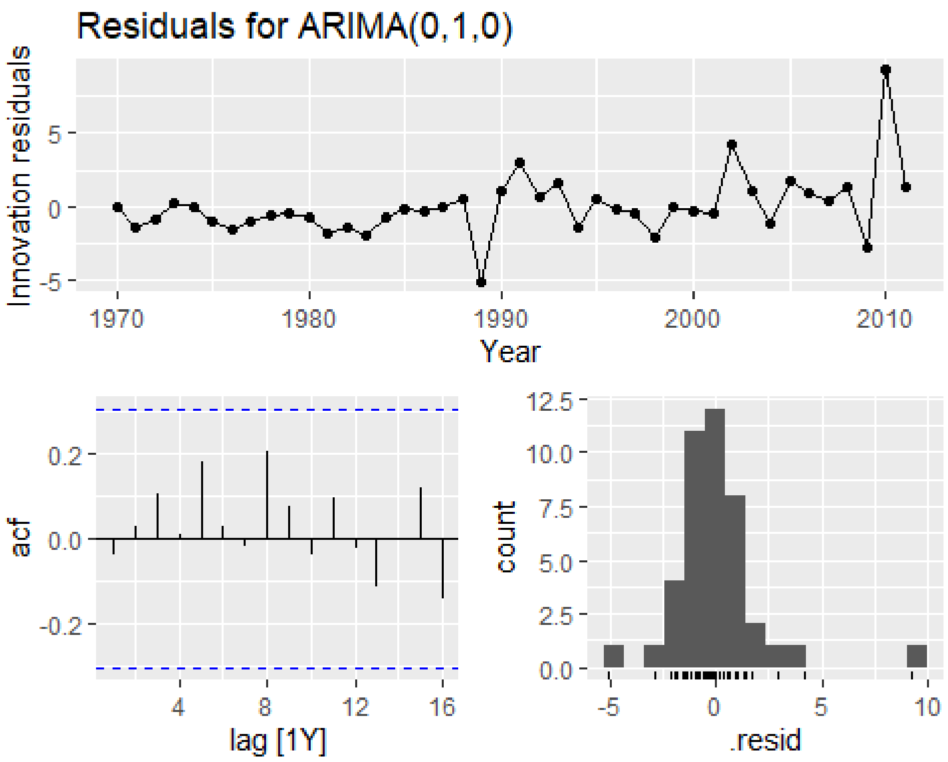
  c. Plot forecasts from an ARIMA(0,1,0) model with drift and compare these to part a.

The ARIMA model from part (a) forecasted values that were higher than the actual observations, whereas this ARIMA model predicted values that were lower than the actual outcomes. Additionally, the slope of the forecasts appears to be more gradual, indicating a slower rate of change over time.

```
fit2 <-aus_airpassengers %>%
  filter(Year < 2012) %>%
  model(ARIMA(Passengers ~ pdq(0,1,0)))

fit2 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "Australian Aircraft Passengers with ARIMA(0,1,0)", y =
"Passengers (in millions)")
```



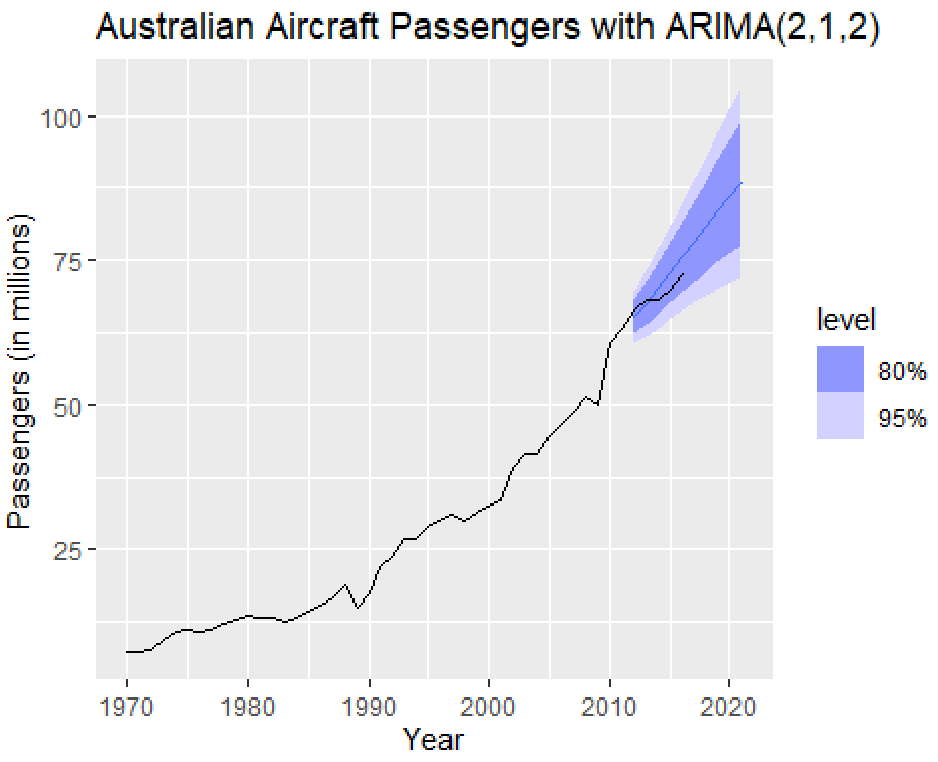
```
fit2 %>%
  gg_tsresiduals() +
  labs(title = "Residuals for ARIMA(0,1,0)")
```



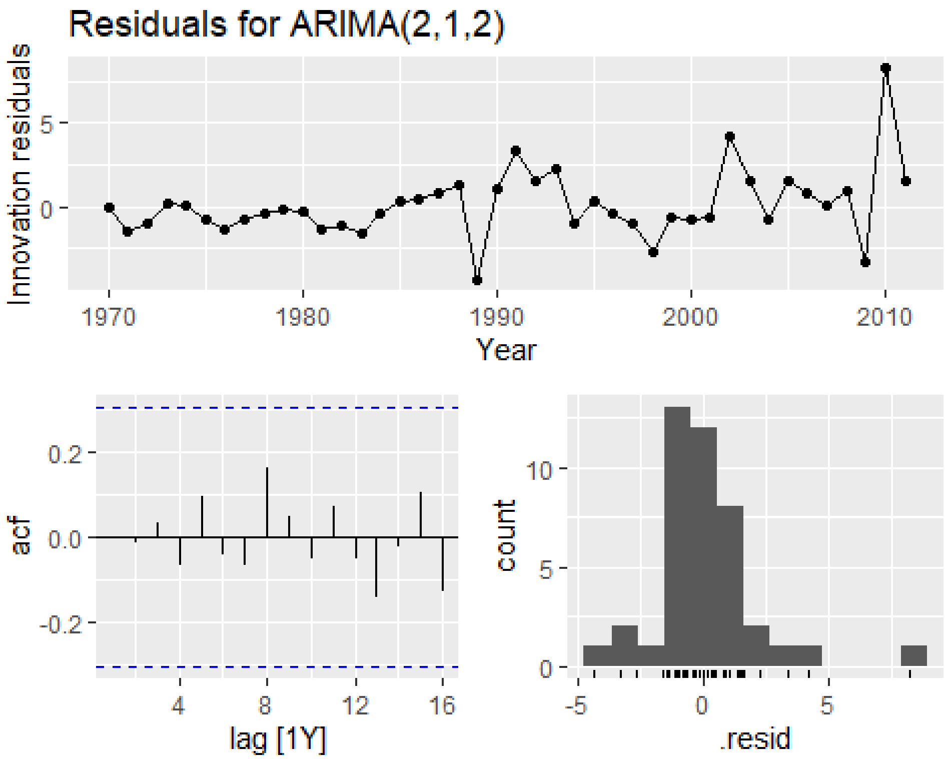
- d. Plot forecasts from an ARIMA(2,1,2) model with drift and compare these to parts a and c. Remove the constant and see what happens

It is more similar to part (a), and the residuals appear to resemble white noise. When the constant is removed, the model effectively becomes a null model. Omitting the constant results in the forecast including a polynomial trend of order $d - 1$—which, in this case, is 0—rendering the series non-stationary.

```
fit3 <-aus_airpassengers %>%
  filter(Year < 2012) %>%
  model(ARIMA(Passengers ~ pdq(2,1,2)))

fit3 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "Australian Aircraft Passengers with ARIMA(2,1,2)", y =
"Passengers (in millions)")
```



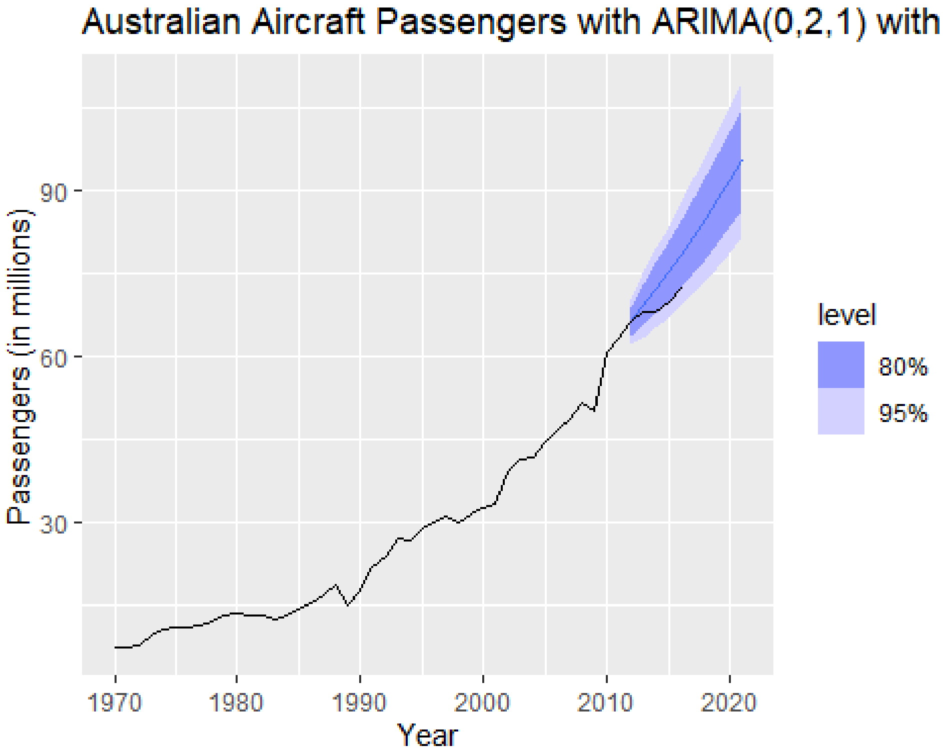
```
fit3 %>%
  gg_tsresiduals() +
  labs(title = "Residuals for ARIMA(2,1,2)")
```



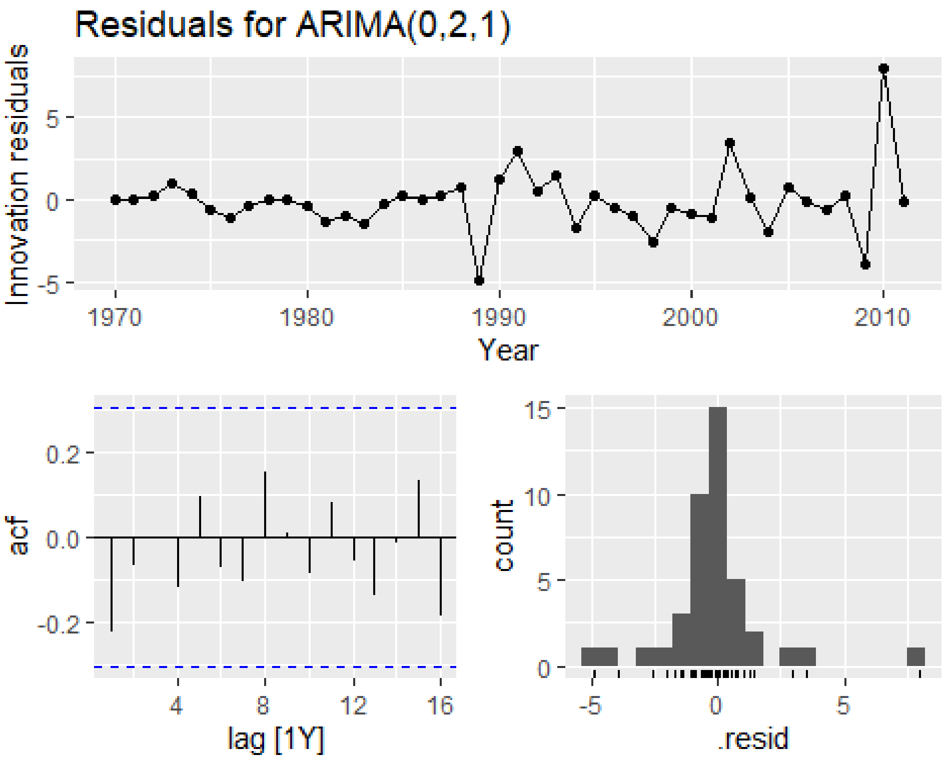
```
#removing constant
fit4 <-aus_airpassengers %>%
  filter(Year < 2012) %>%
  model(ARIMA(Passengers ~ 0 + pdq(2,1,2)))

report(fit4)

## Series: Passengers
## Model: NULL model
## NULL model
```

e. Plot forecasts from an ARIMA(0,2,1) model with a constant. What happens?

```
fit5 <-aus_airpassengers %>%
  filter(Year < 2012) %>%
  model(ARIMA(Passengers ~ 1 + pdq(0,2,1)))

## Warning: Model specification induces a quadratic or higher order
polynomial trend.
## This is generally discouraged, consider removing the constant or reducing
the number of differences.

fit5 %>%
  forecast(h=10) %>%
  autoplot(aus_airpassengers) +
  labs(title = "Australian Aircraft Passengers with ARIMA(0,2,1) with
constant", y = "Passengers (in millions)")
```



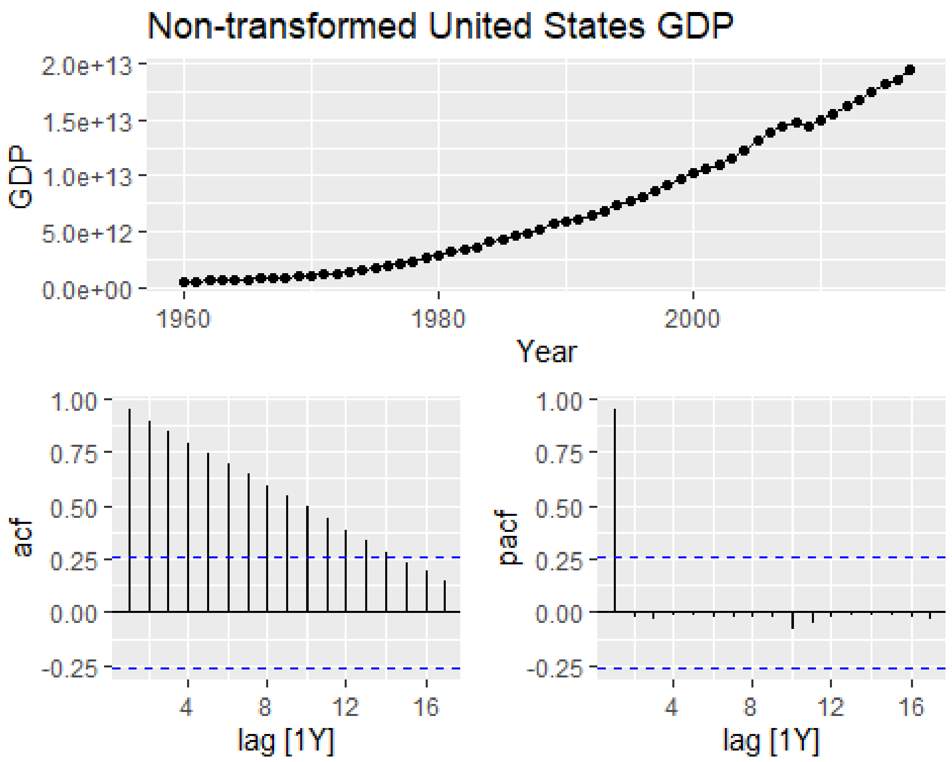
```
fit5 %>%
  gg_tsresiduals() +
  labs(title = "Residuals for ARIMA(0,2,1) ")
```



Exercise 9.8 or the United States GDP series (from global_economy):

- if necessary, find a suitable Box-Cox transformation for the data;
- fit a suitable ARIMA model to the transformed data using ARIMA();
- try some other plausible models by experimenting with the orders chosen;
- choose what you think is the best model and check the residual diagnostics;
- produce forecasts of your fitted model. Do the forecasts look reasonable?
- compare the results with what you would obtain using ETS() (with no transformation).

```
usa_gdp <- global_economy |> filter(Code == 'USA')
usa_gdp |> gg_tsdisplay(GDP, plot_type = 'partial') + labs(title = "Non-
transformed United States GDP")
```



-

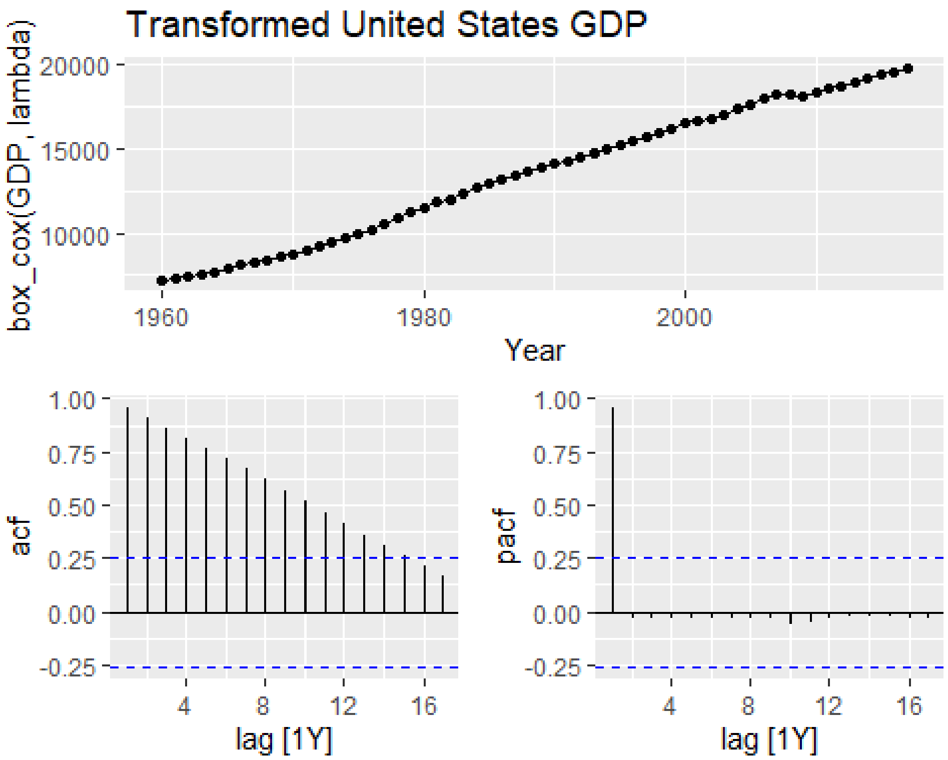
    a. if necessary, find a suitable Box-Cox transformation for the data;

ARIMA(1,1,0) with a drift was fitted to the tansformed data.

```
lambda  <-usa_gdp %>%
  features(GDP, features = guerrero) %>%
  pull(lambda_guerrero)
lambda

## [1] 0.2819443

# fit a suitable ARIMA model to the transformed data using ARIMA();
fit_usa <- global_economy %>%
  filter(Code == "USA") %>%
  model(ARIMA(box_cox(GDP, lambda)))

report(fit_usa)

## Series: GDP
## Model: ARIMA(1,1,0) w/ drift
## Transformation: box_cox(GDP, lambda)
##
## Coefficients:
##          ar1  constant
##       0.4586  118.1822
## s.e.  0.1198    9.5047
```

```
##
## sigma^2 estimated as 5479:  log likelihood=-325.32
## AIC=656.65   AICc=657.1   BIC=662.78
```

```
# transformed plot
usa_gdp %>%
  gg_tsdisplay(box_cox(GDP,lambda), plot_type='partial') +
  labs(title = "Transformed United States GDP")
```

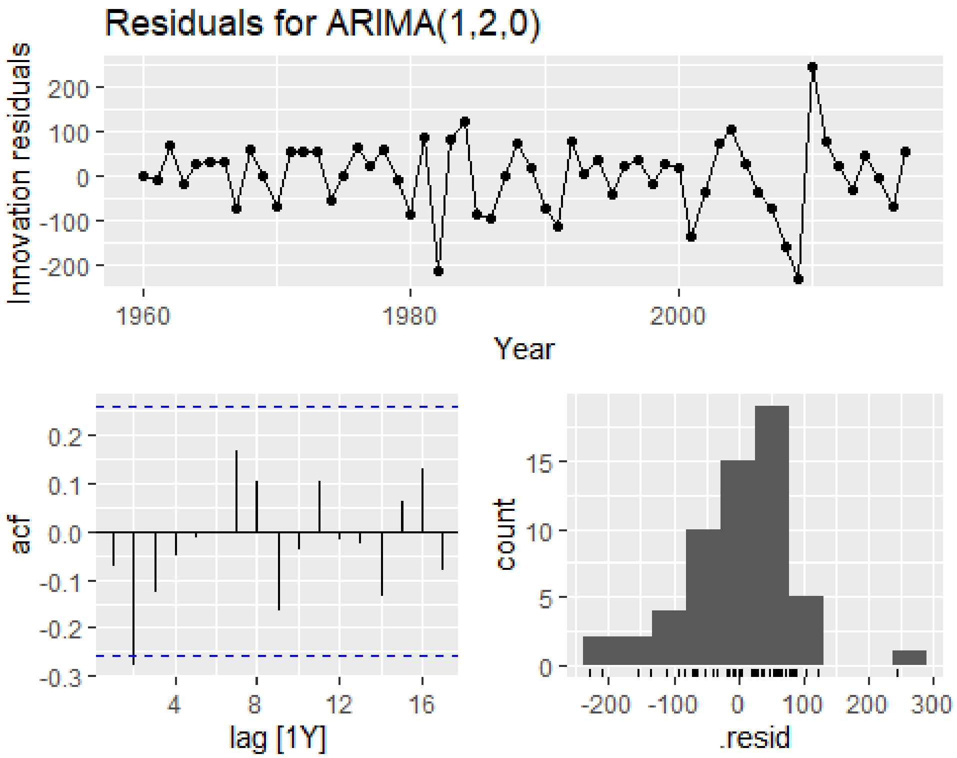


```
# unit root test
usa_gdp %>%
  features(box_cox(GDP,lambda), unitroot_ndiffs)
```

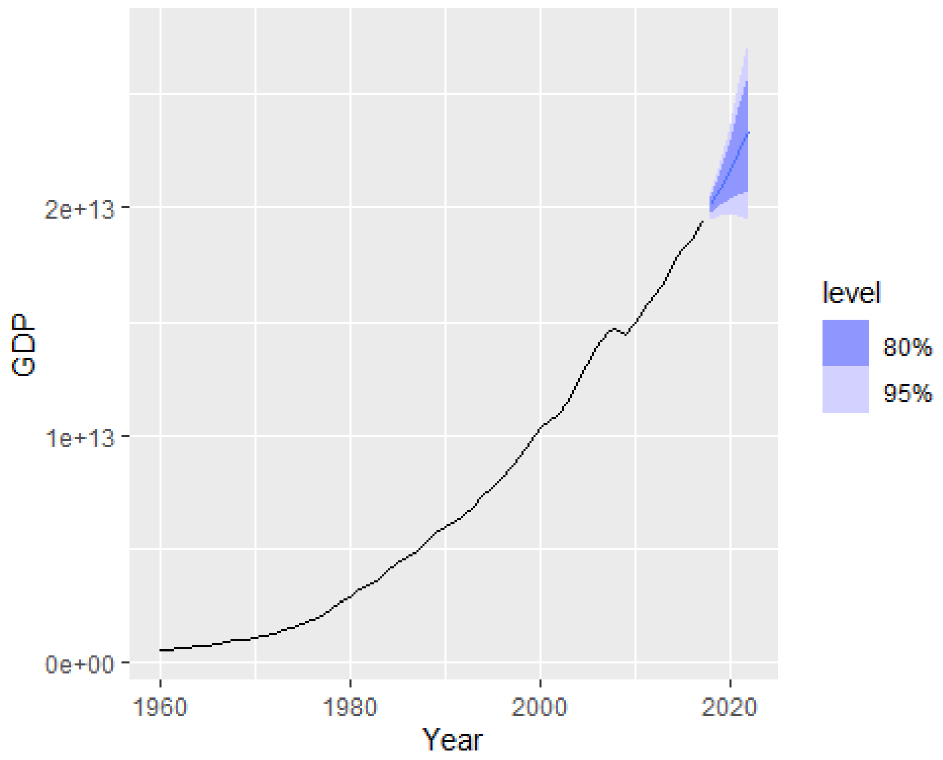
```
## # A tibble: 1 × 2
##   Country       ndiffs
##   <fct>          <int>
## 1 United States      1
```

```
# modeling several
usa_fit <- global_economy %>%
  filter(Code == "USA") %>%
  model(arima110 = ARIMA(box_cox(GDP,lambda) ~ pdq(1,1,0)),
        arima120 = ARIMA(box_cox(GDP,lambda) ~ pdq(1,2,0)),
        arima210 = ARIMA(box_cox(GDP,lambda) ~ pdq(2,1,0)),
        arima212 = ARIMA(box_cox(GDP,lambda) ~ pdq(2,1,2)),
        arima111 = ARIMA(box_cox(GDP,lambda) ~ pdq(1,1,1)))
```

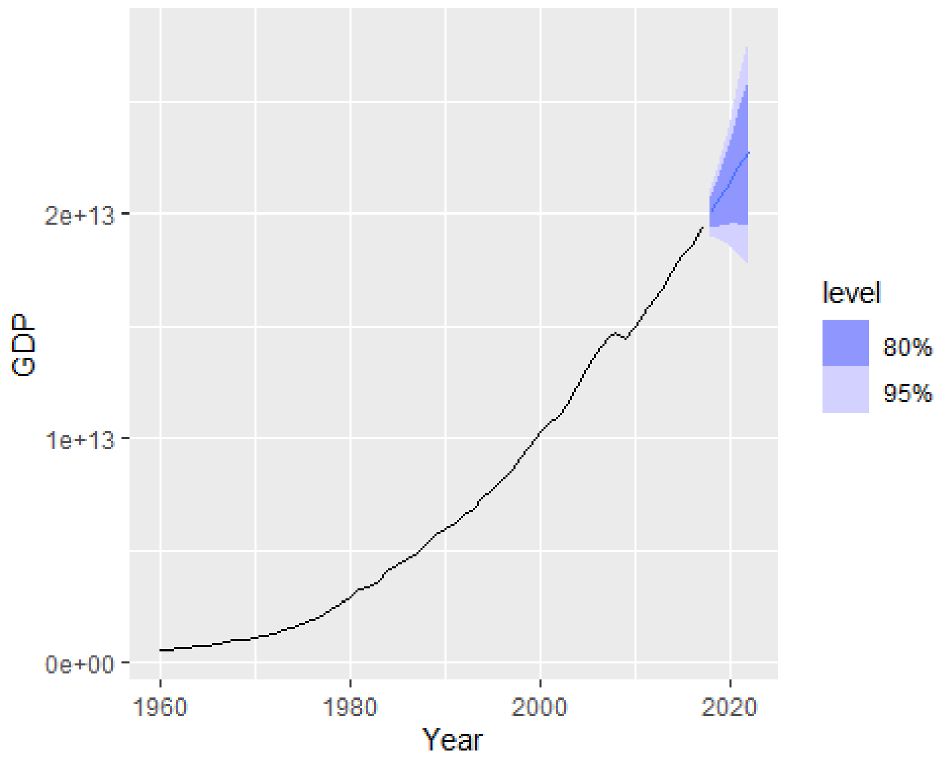
```
glance(usa_fit) %>% arrange(AICc) %>% select(.model:BIC)

## # A tibble: 5 × 6
##   .model   sigma2 log_lik   AIC  AICc   BIC
##   <chr>     <dbl>   <dbl> <dbl> <dbl> <dbl>
## 1 arima120  6780.   -326.  656.  656.  660.
## 2 arima110  5479.   -325.  657.  657.  663.
## 3 arima111  5580.   -325.  659.  659.  667.
## 4 arima210  5580.   -325.  659.  659.  667.
## 5 arima212  5734.   -325.  662.  664.  674.

# d. choose what you think is the best model and check the residual
diagnostics;
usa_fit %>%
  select(arima120) %>%
  gg_tsresiduals() +
  ggtitle("Residuals for ARIMA(1,2,0)")
```



```
# e. produce forecasts of your fitted model. Do the forecasts look
reasonable?
usa_fit %>%
  forecast(h=5) %>%
  filter(.model=='arima120') %>%
  autoplot(global_economy)
```



```
# f. compare the results with what you would obtain using ETS() (with no
transformation).
fit_ets <- global_economy %>%
  filter(Code == "USA") %>%
  model(ETS(GDP))

report(fit_ets)

## Series: GDP
## Model: ETS(M,A,N)
##   Smoothing parameters:
##     alpha = 0.9990876
##     beta  = 0.5011949
##
##   Initial states:
##          l[0]         b[0]
##  448093333334 64917355687
##
##   sigma^2:  7e-04
##
##      AIC     AICc      BIC
## 3190.787 3191.941 3201.089

fit_ets %>%
  forecast(h=5) %>%
  autoplot(global_economy)
```



The analysis of the United States GDP series from the `global_economy` dataset involved applying a Box-Cox transformation to stabilize the variance, followed by fitting an ARIMA(1,2,0) model. The transformed series showed a clear upward trend and strong autocorrelation, indicating the need for differencing. After fitting the ARIMA model, the residual diagnostics suggested that the residuals behaved like white noise, with no significant autocorrelation and an approximately normal distribution. Forecasts from the ARIMA model were reasonable, showing a smooth upward trajectory with appropriate uncertainty bounds. A comparison with the ETS model (without transformation) produced similar forecast behavior, suggesting that both approaches captured the underlying trend effectively. However, ARIMA may offer better interpretability due to its explicit handling of differencing and autocorrelation.