# Chapter 9

## Assignment 1

Consider aus\_airpassengers, the total number of passengers (in millions) from Australian air carriers for the period 1970-2011.

### a. Use ARIMA() to find an appropriate ARIMA model. What model was selected. Check that the residuals look like white noise. Plot forecasts for the next 10 periods.

data <- aus\_airpassengers %>%

filter(Year >= 1970 & Year <= 2011) %>%

select(Year, Passengers)

arima\_model\_1 <- data %>% model(ARIMA(Passengers))

report(arima\_model\_1)

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

=> the selected model was ARIMA(0,2,1).

Check the residuals:

arima\_model\_1 %>%

gg\_tsresiduals()

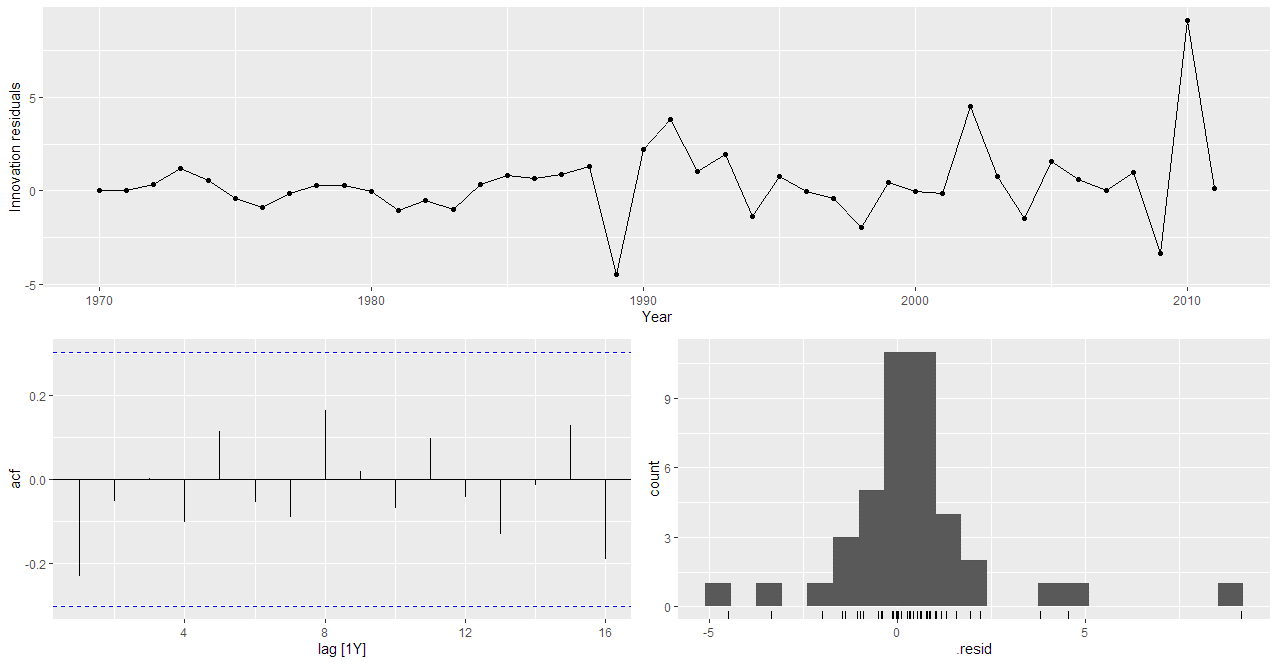


Figure : Residuals graph for ARIMA(0,2,1) model

Plot the forecast for the next 10 years:

next\_10\_years\_forecast <- arima\_model\_1 %>% forecast(h = 10)

next\_10\_years\_forecast %>% autoplot(data) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

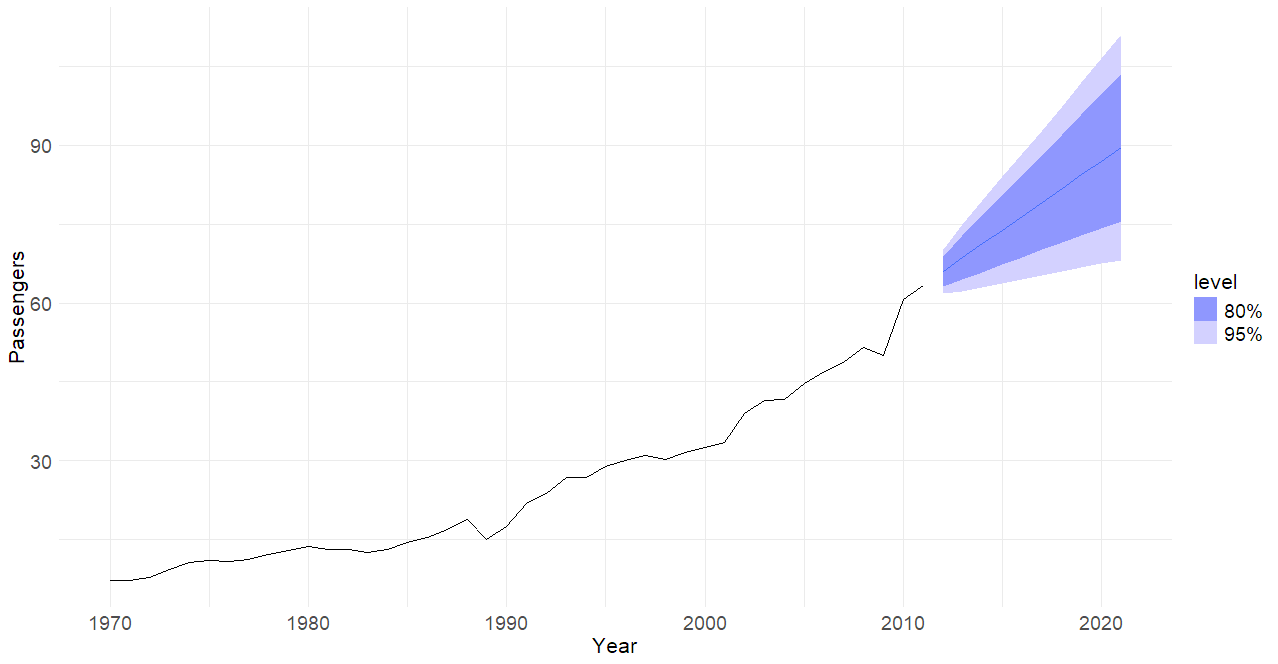


Figure : Next 10 years forecast using ARIMA(0,2,1) model

### b. Write the model in terms of the backshift operator.

Based on the formular in the slide

Based on the result from the report() method, we have:

* Since , there is no autoregressive component. Therefore,.
* . Therefore

Final equation:

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

arima\_model\_2 <- data %>% model(ARIMA(Passengers ~ 1 + pdq(0,1,0)))

report(arima\_model\_2)

Series: Passengers

Model: ARIMA(0,1,0) w/ drift

Coefficients:

constant

1.3669

s.e. 0.3319

sigma^2 estimated as 4.629: log likelihood=-89.08

AIC=182.17 AICc=182.48 BIC=185.59

Plot the forecast:

arima\_model\_1\_2 <- data %>%

model(

PartA = ARIMA(Passengers),

PartC = ARIMA(Passengers ~ 1 + pdq(0,1,0))

)

next\_10\_years\_forecast <- arima\_model\_1\_2 %>% forecast(h = 10)

next\_10\_years\_forecast %>% autoplot(data) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

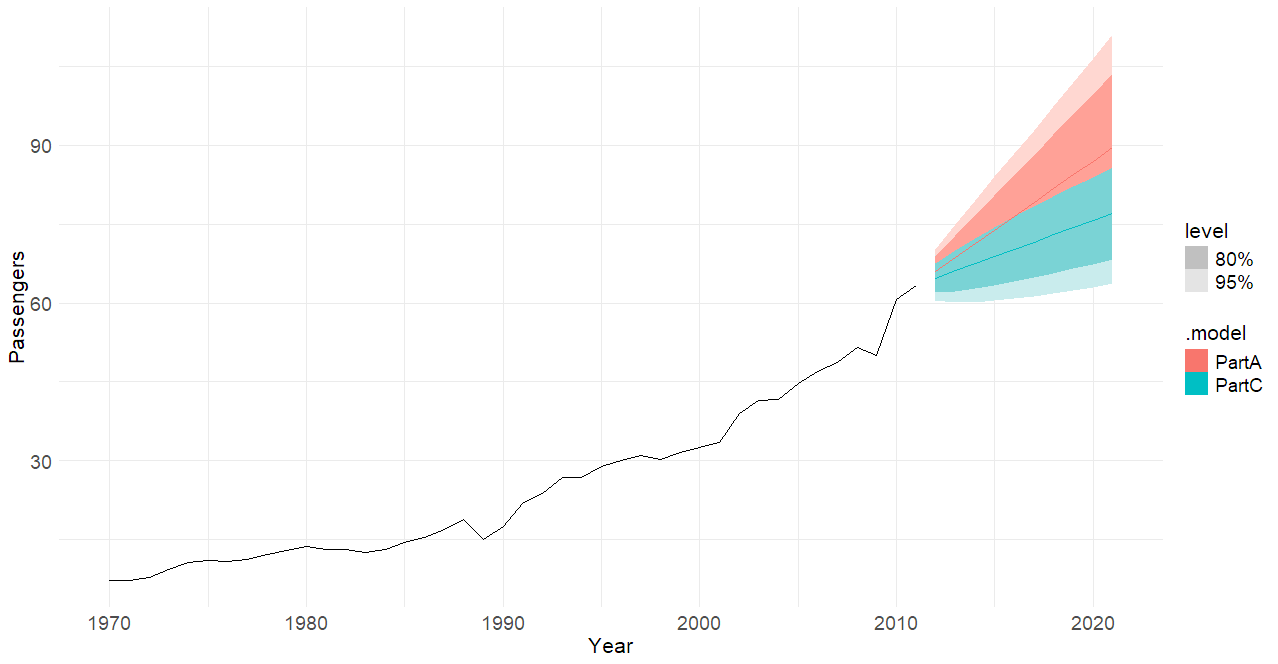


Figure : Next 10 years forecast comparison between ARIMA(0,2,1) and ARIMA(0,1,0) models

ARIMA(0,1,0) w/ drift compared to the auto generated has a larger AIC and BIC values. Therefore, the auto generated ARIMA(0,2,1) outperforms this ARIMA(0,1,0) w/ drift model.

### 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.

ARIMA(2,1,2) model with drift:

arima\_model\_1\_2\_3 <- data %>%

model(

PartA = ARIMA(Passengers),

PartC = ARIMA(Passengers ~ 1 + pdq(0,1,0)),

PartD = ARIMA(Passengers ~ 1 + pdq(2,1,2))

)

next\_10\_years\_forecast <- arima\_model\_1\_2\_3 %>% forecast(h = 10)

next\_10\_years\_forecast %>% autoplot(data) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

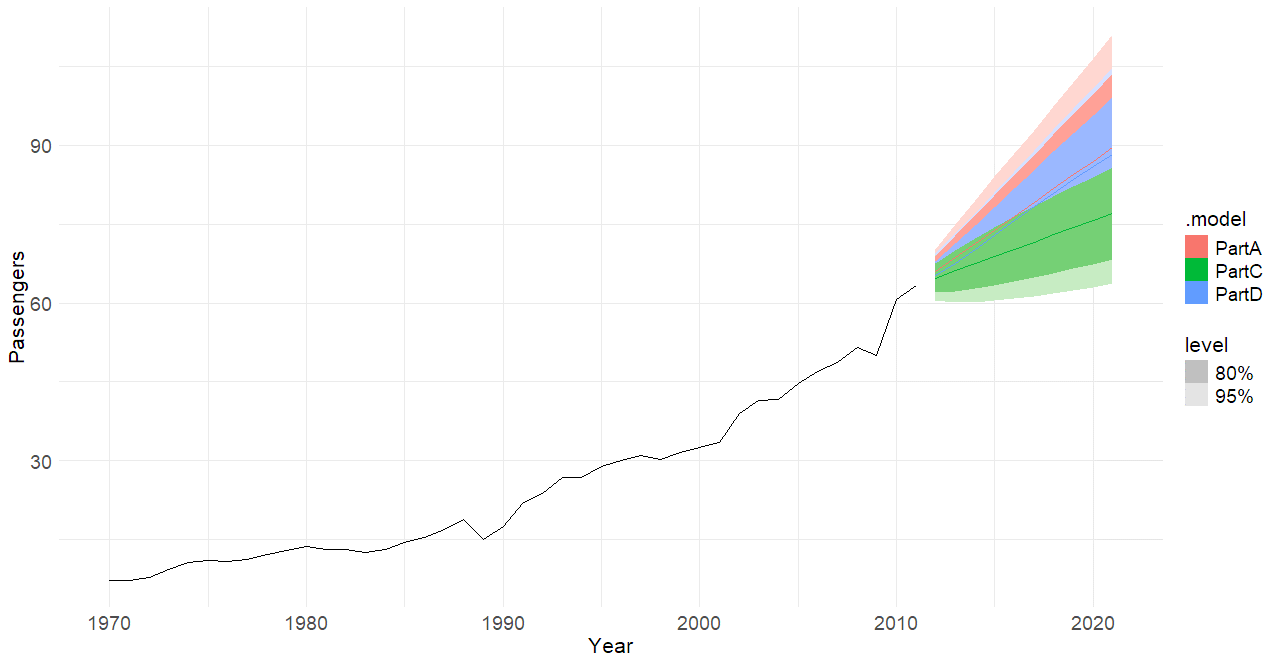


Figure : Next 10 years forecast comparison between ARIMA(2,1,2) and ARIMA(0,2,1), ARIMA(0,1,0) models

The ARIMA(2, 1, 2) model with a constant seems to be a bit similar to the ARIMA(0, 1, 0) model but is a slightly worse fit overall. The best fitting model appears to be Part A model, ARIMA(0, 2, 1).

Removing the constant throws an error and does not produce an ARIMA model.

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

arima\_model\_4 <- data %>% model(ARIMA(Passengers ~ 1 + pdq(0,2,1)))

next\_10\_years\_forecast <- arima\_model\_4 %>% forecast(h = 10)

next\_10\_years\_forecast %>% autoplot(data) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

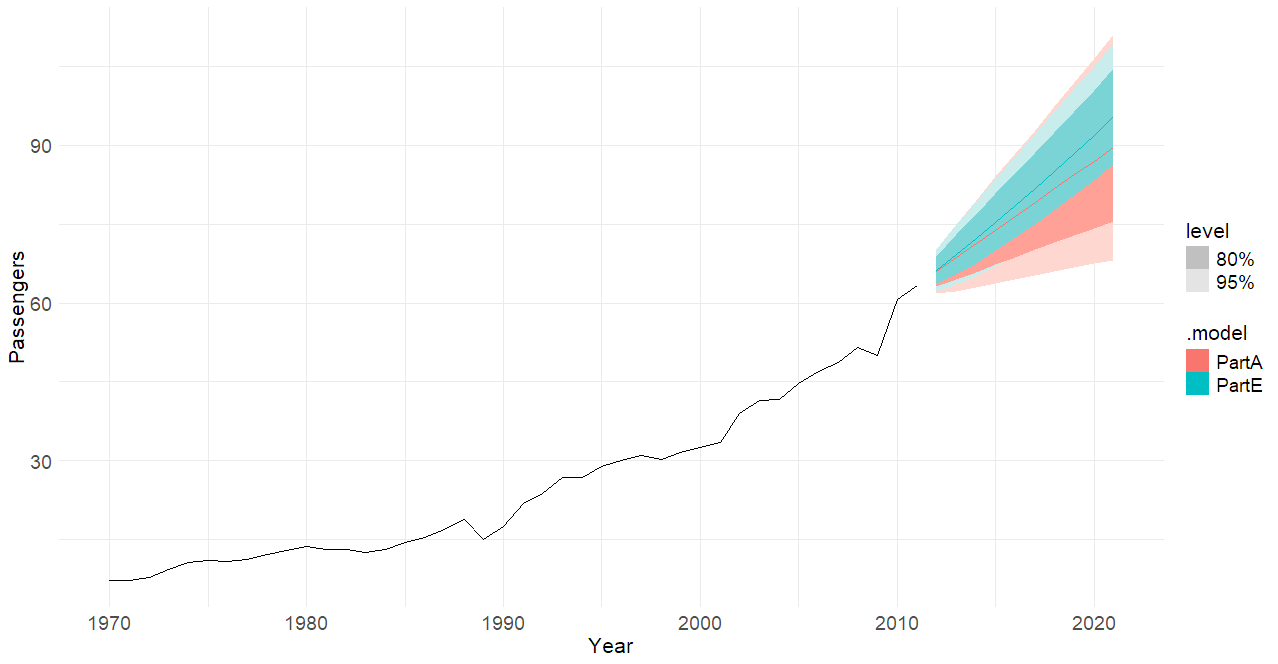


Figure : Next 10 years forecast using ARIMA(0,2,1)

ARIMA(0,2,1) w/ drift performed better than the auto generated ARIMA(0,2,1) model.

## Assignment 2

For the United States GDP series (from global\_economy)

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

us\_gdp <- global\_economy %>%

filter(Country == "United States") %>%

select(Year, GDP)

lambda <- us\_gdp %>%

features(GDP, features = guerrero) %>%

pull(guerrero\_lambda)

us\_gdp <- us\_gdp %>%

mutate(GDP\_transformed = box\_cox(GDP, lambda))

### b. fit a suitable ARIMA model to the transformed data using ARIMA()

arima\_model <- us\_gdp %>%

model(ARIMA(GDP\_transformed))

report(arima\_model)

Series: GDP\_transformed

Model: ARIMA(1,1,0) w/ drift

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

Model found: ARIMA(1,1,0) w/ drift

### c. try some other plausible models by experimenting with the orders chosen

arima\_model\_1\_2\_3 <- us\_gdp %>%

model(

ARIMA\_011 = ARIMA(GDP\_transformed ~ pdq(0,1,1)),

ARIMA\_110 = ARIMA(GDP\_transformed ~ pdq(1,1,0)),

ARIMA\_111 = ARIMA(GDP\_transformed ~ pdq(1,1,1))

)

glance(arima\_model\_1\_2\_3)

# A tibble: 3 × 8

.model sigma2 log\_lik AIC AICc BIC ar\_roots ma\_roots

<chr> <dbl> <dbl> <dbl> <dbl> <dbl> <list> <list>

1 ARIMA\_011 5689. -326. 659. 659. 665. <cpl [0]> <cpl [1]>

2 ARIMA\_110 5479. -325. 657. 657. 663. <cpl [1]> <cpl [0]>

3 ARIMA\_111 5580. -325. 659. 659. 667. <cpl [1]> <cpl [1]>

Model: ARIMA(1,1,0) has the lowest AICc (657) and lowest sigma2 (5479). ARIMA(1,1,0) is simpler than ARIMA(1,1,1) and fits better than ARIMA(0,1,1).

### d. choose what you think is the best model and check the residual diagnostics

best\_model <- arima\_model\_1\_2\_3 %>% select(ARIMA\_110)

best\_model %>% gg\_tsresiduals()

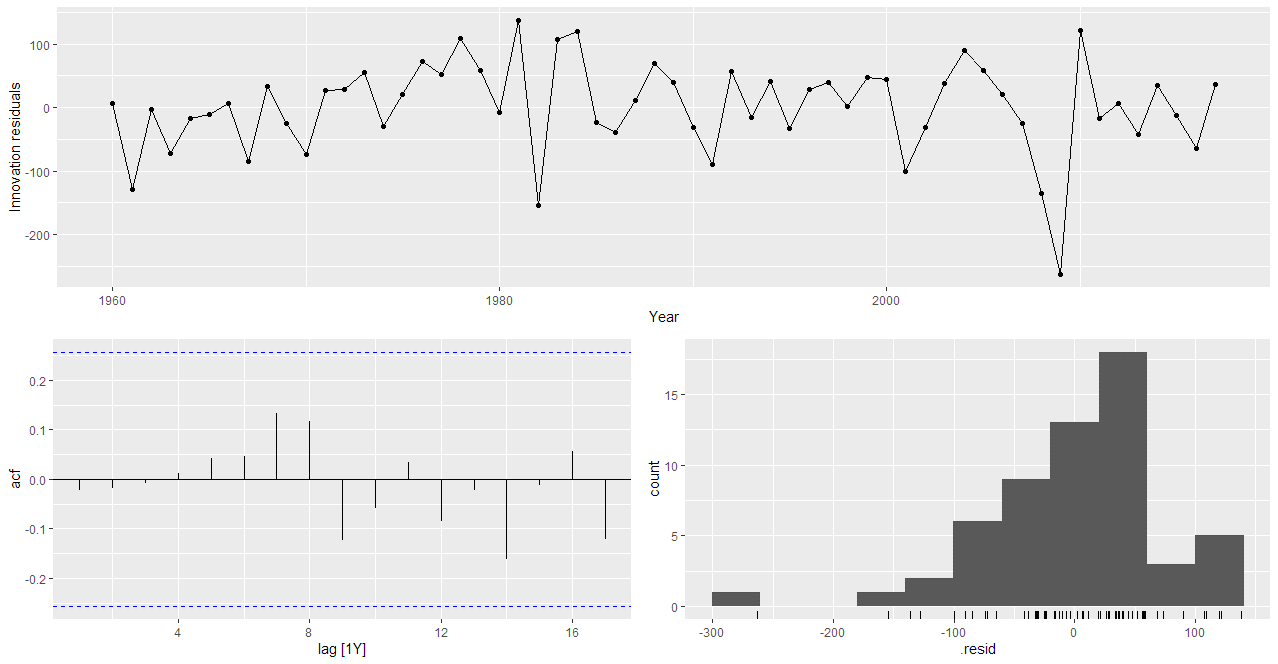


Figure : Residuals plot for ARIMA(1,1,0) model

best\_model %>% augment() %>% features(.innov, ljung\_box, lag = 10)

# A tibble: 1 × 3

.model lb\_stat lb\_pvalue

<chr> <dbl> <dbl>

1 ARIMA\_110 3.81 0.955

The ARIMA(1,1,0) model shows no autocorrelation or trend in the residuals, and despite slight skewness, the residuals behave like white noise. Overall, the model is a good fit and suitable for forecasting.

### e. produce forecasts of your fitted model. Do the forecasts look reasonable?

forecast\_arima <- best\_model %>% forecast(h = "10 years")

forecast\_arima %>%

autoplot(us\_gdp) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

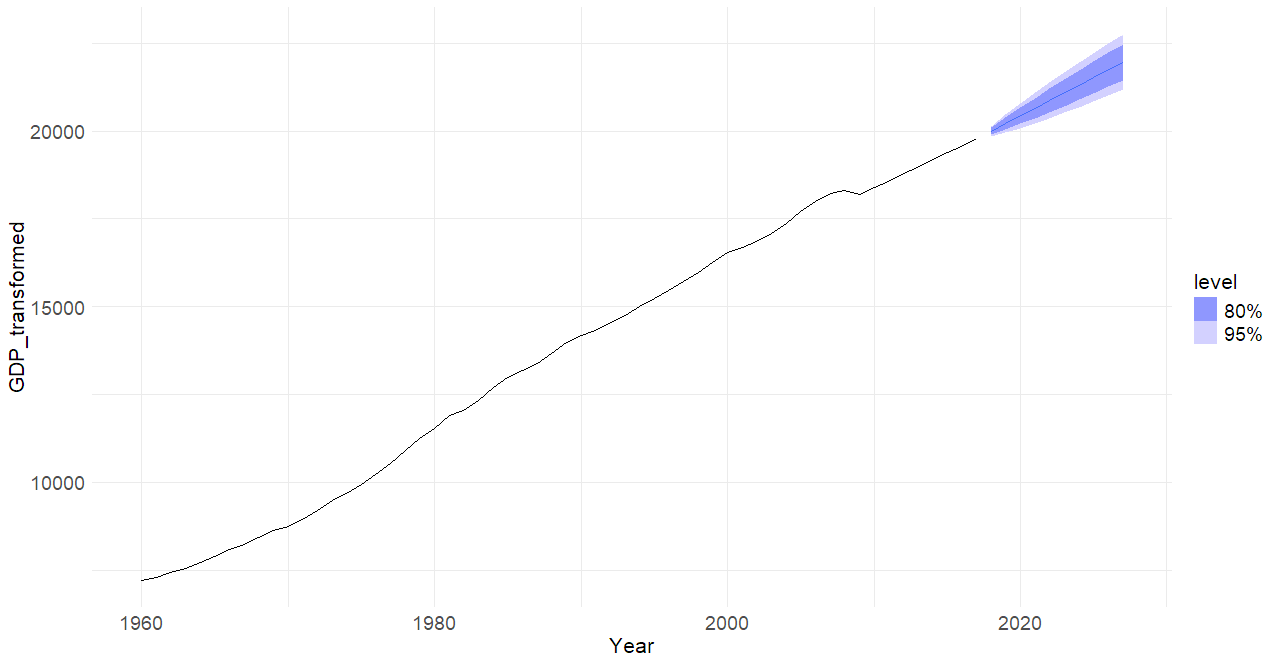


Figure : Forecast using for ARIMA(1,1,0) model

The forecasts look reasonable.

### f. compare the results with what you would obtain using ETS() (with no transformation).

ets\_model <- us\_gdp %>%

model(

ETS = ETS(GDP)

)

forecast\_ets <- ets\_model %>% forecast(h = "10 years")

forecast\_ets %>%

autoplot(us\_gdp) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

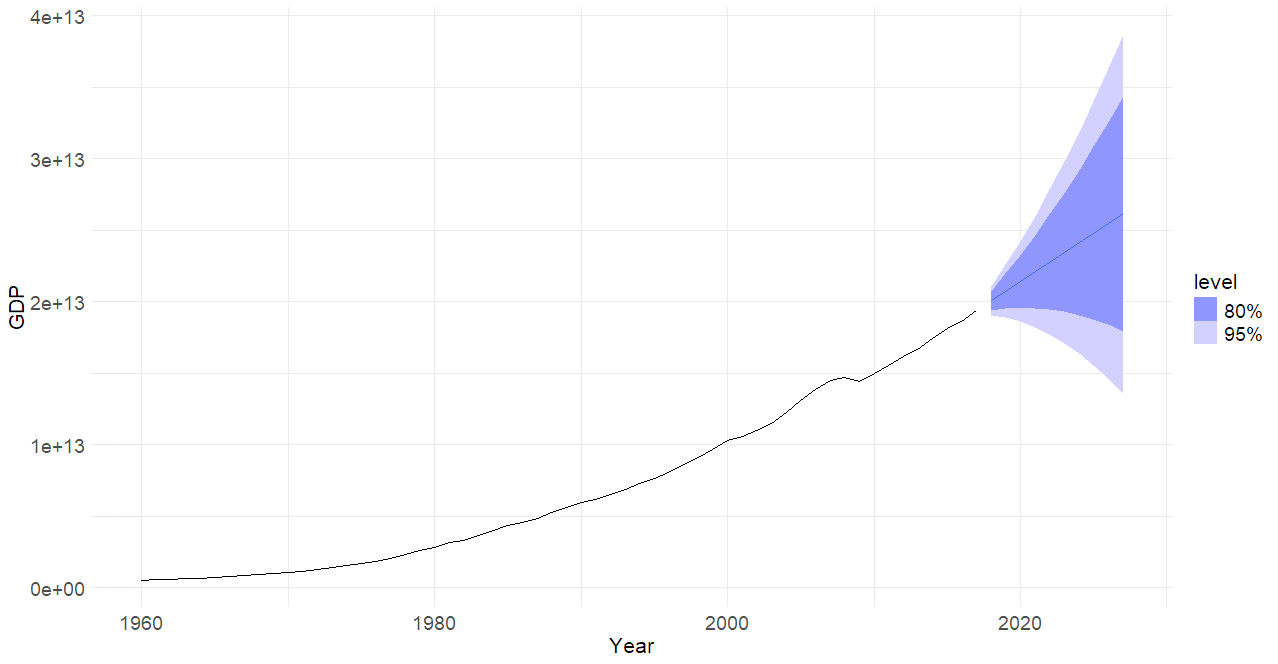


Figure : Forecast using for ETS (no transformation) model

The ARIMA model has better fit metrics than the ETS model. The ETS model also has a much wider prediction interval compared to my selected ARIMA(1,1,0) model.

## Assignment 3

Consider the file found in the “File for Homework Assignments” folder on BlackBoard with the title KIC9832227.csv. It contains data on the from the light curves obtained from the Hubble Space Telescope. KIC 9832227 is a contact binary star system in the constellation Cygnus, located about 2,060 light-years away. Physicists can tell many characteristics about the star by examining these light curves. In this assignment you will use the first 90% of the data to train your ETS model and forecast the next 10% of the data. The first few lines of the data file are shown below. The first column list what set the data is from. The second column is Barycentric Julian Date (BJD) is the Julian Date (JD) corrected for differences in the Earth's position with respect to the barycentre of the Solar System. In the third column I have converted it to Julian Date and Time for you. The last column list the flux of the star.

### a. Plot the ndiv\_PDCSAP\_FLUX series versus Julian Date and discuss the main features of the data.

library(tidyverse)

library(lubridate)

library(fpp3)

library(dplyr)

kic\_df <- read\_csv("data/KIC9832227.csv")

kic\_df <- kic\_df %>%

rename(JD\_Time = "JD Time")

kic\_df %>%

ggplot(aes(x = TIME, y = ndiv\_PDCSAP\_FLUX)) +

geom\_line(size=1) +

labs(title = "Flux vs Julian Date",

y = "Flux of the star",

x = "Julian Date") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

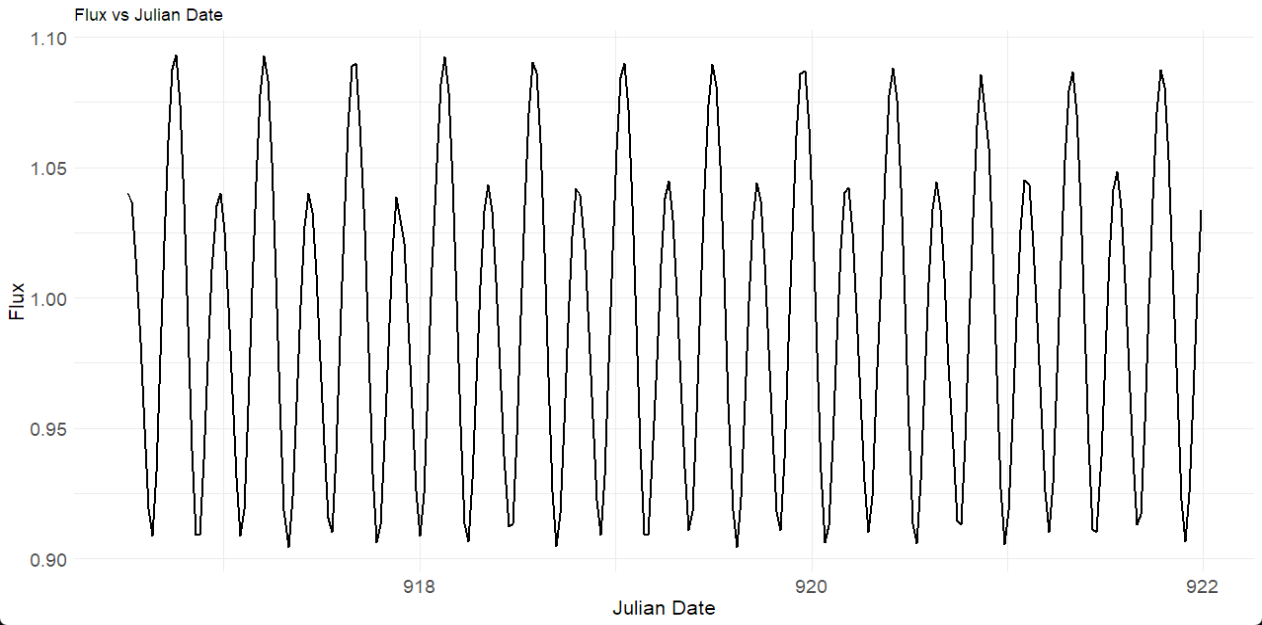


Figure : Time vs Flux plot

This line graph from the KIC 9832227 contact binary star system displays a seasonal pattern with no apparent trend. The pattern repeats roughly every 0.5 days, corresponding to the system's orbital cycle.

### b. Use an appropriate ETS model to forecast the series and plot the forecasts. (you may use your code from Chapter 8).

kic\_ts <- kic\_df %>%

mutate(index = row\_number()) %>%

as\_tsibble(index = index)

total\_rows <- nrow(kic\_ts)

split\_point <- floor(0.9 \* total\_rows)

train\_ts <- kic\_ts[1:split\_point, ]

test\_ts <- kic\_ts[(split\_point + 1):total\_rows, ]

aan\_model <- train\_ts %>%

model(ETS(ndiv\_PDCSAP\_FLUX ~ error("A") + trend("A") + season("N")))

ets\_fc <- aan\_model %>%

forecast(h = nrow(test\_ts))

ets\_fc %>%

autoplot(train\_ts) +

labs(y = "Flux of the star", x="Time index") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

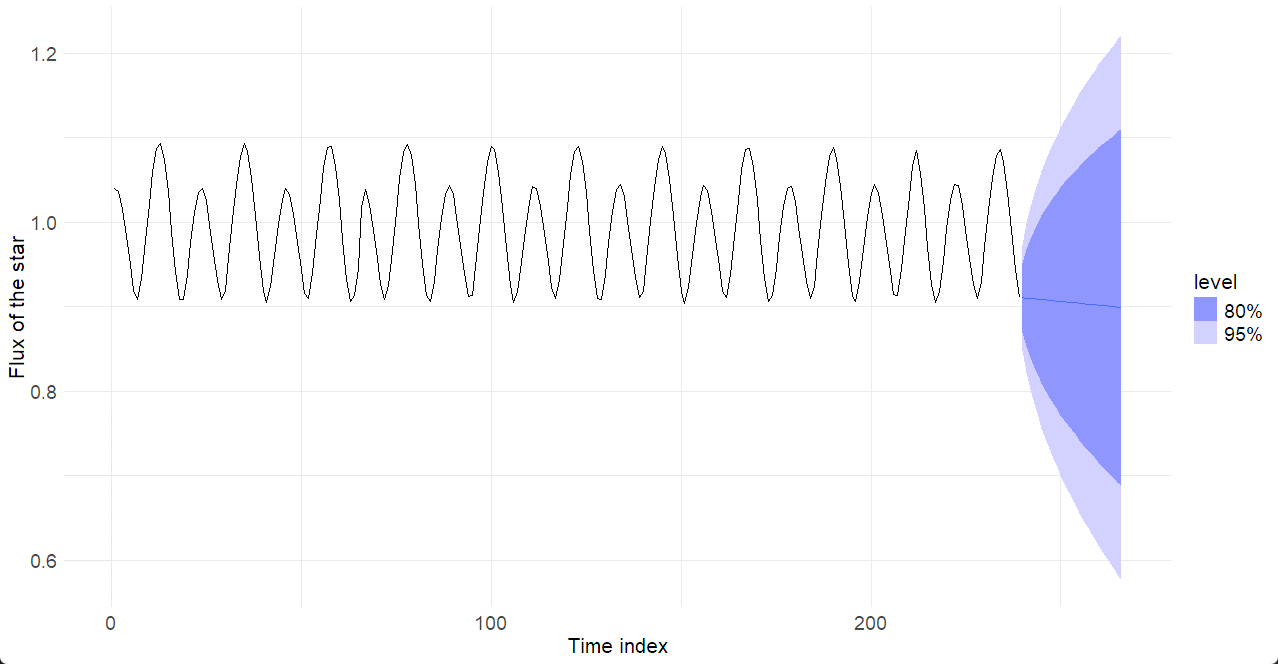
) 

Figure : Forecast using ETS model

The ETS model produced poor forecast result since it has missing data in the TIME column. It has a very wide prediction interval.

### c. Use an appropriate ARIMA model to forecast the series and plot the forecasts.

arima\_model <- train\_ts %>%

model(ARIMA(ndiv\_PDCSAP\_FLUX))

arima\_fc <- arima\_model %>% forecast(h = nrow(test\_ts))

# Plot

arima\_fc %>%

autoplot(train\_ts, level = NULL) +

labs(y = "Flux of the star", x="Time index") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

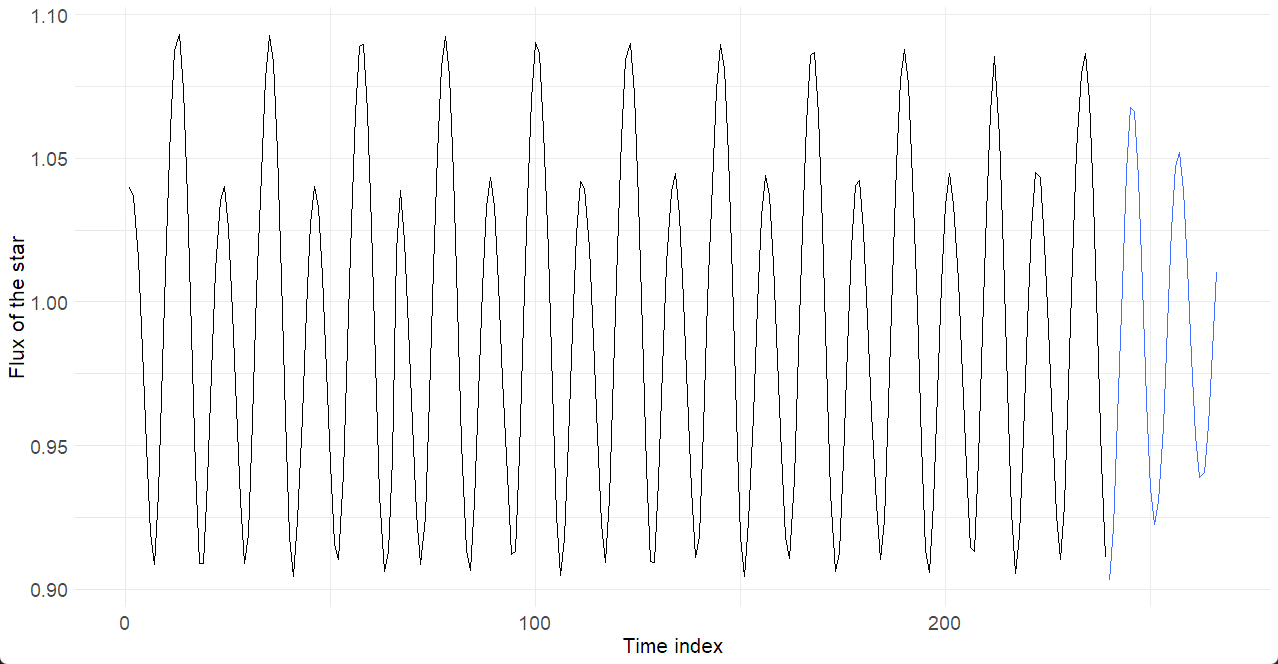


Figure : Forecast using ARIMA model

I can see that ARIMA model produced a better result.

### d. Which model did a better job? Defend your answer with data.

bind\_rows(

accuracy(ets\_fc, test\_ts) %>% mutate(model = "ETS"),

accuracy(arima\_fc, test\_ts) %>% mutate(model = "ARIMA")

)

# A tibble: 2 × 11

.model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1 model

<chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>

1 "ETS(ndiv\_PDCSAP\_FLUX ~ error(\"A\") + trend(\"A\") + season(\"N\"))" Test 0.0847 0.102 0.0847 8.27 8.27 NaN NaN 0.790 ETS

2 "ARIMA(ndiv\_PDCSAP\_FLUX)" Test 0.00234 0.0261 0.0228 0.164 2.29 NaN NaN 0.871 ARIMA

The ARIMA model outperforms the ETS model across all accuracy metrics, including lower RMSE (0.0261 vs 0.102) and MAPE (2.29% vs 8.27%). This indicates ARIMA provides significantly more accurate forecasts for the flux series than ETS. While ACF1 is slightly lower for ETS, both values suggest some residual autocorrelation. Overall, ARIMA is the better model.