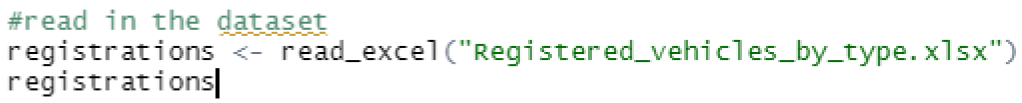
STAT8122 Time Series Assignment 2

Aidan Van Klaveren

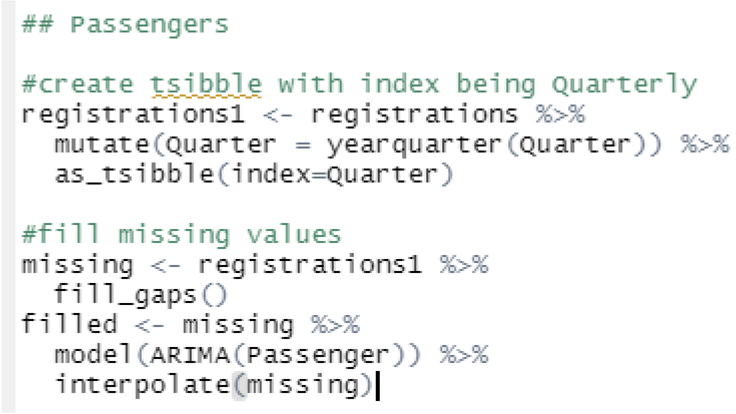
44588070

Question 1



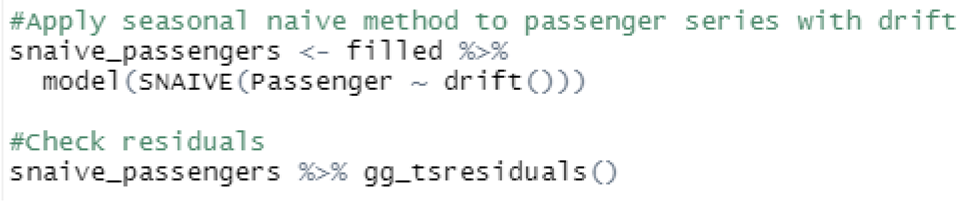
```
#read in the dataset
registrations <- read_excel("Registered_vehicles_by_type.xlsx")
registrations
```

a) Apply seasonal naïve forecast to Passenger series

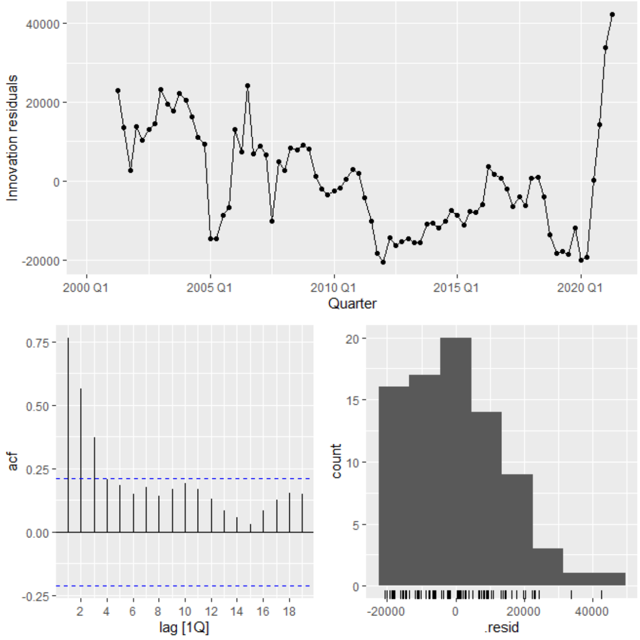


```
## Passengers

#create tsibble with index being Quarterly
registrations1 <- registrations %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(index=Quarter)

#fill missing values
missing <- registrations1 %>%
  fill_gaps()
filled <- missing %>%
  model(ARIMA(Passenger)) %>%
  interpolate(missing)
```

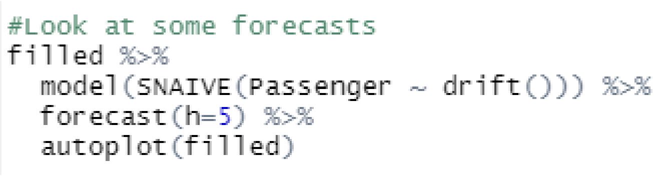
The code cleans the data and adds in ARIMA fitted missing values into the corresponding Quarters.



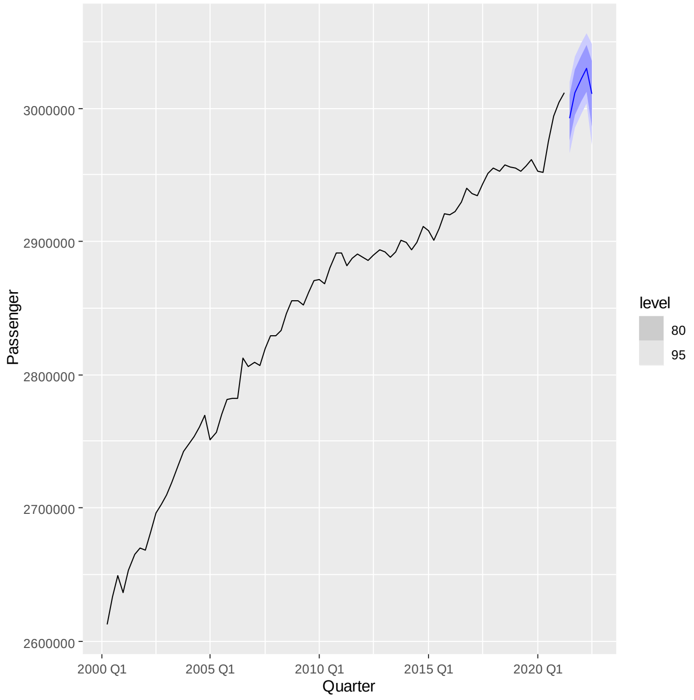
```
#Apply seasonal naive method to passenger series with drift
snaive_passengers <- filled %>%
  model(SNAIVE(Passenger ~ drift()))

#Check residuals
snaive_passengers %>% gg_tsresiduals()
```



The residuals seem significant until lag 3 as they all cross the dotted blue lines. Lag 4 seems to just stay within the bounds. The rest of the series seems to be white-noise. The distributions of residuals do not seem to be normally distributed as the residual are right skewed. The residuals are centred around 0.

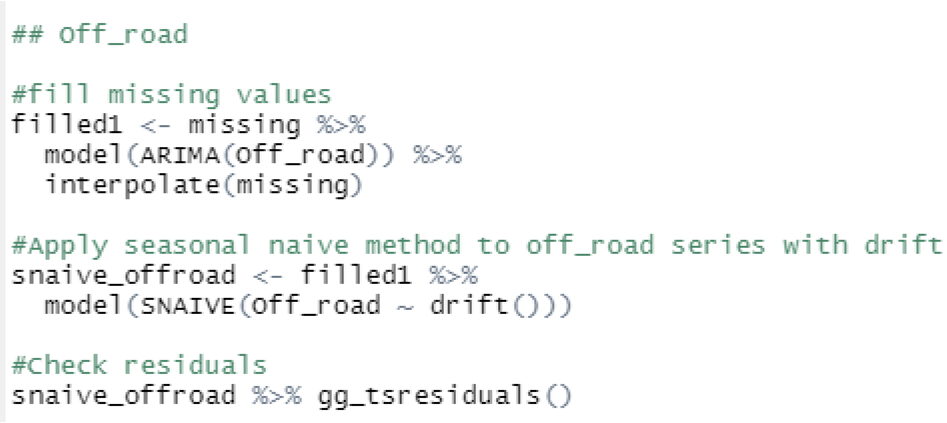


```
#Look at some forecasts
filled %>%
  model(SNAIVE(Passenger ~ drift())) %>%
  forecast(h=5) %>%
  autoplot(filled)
```

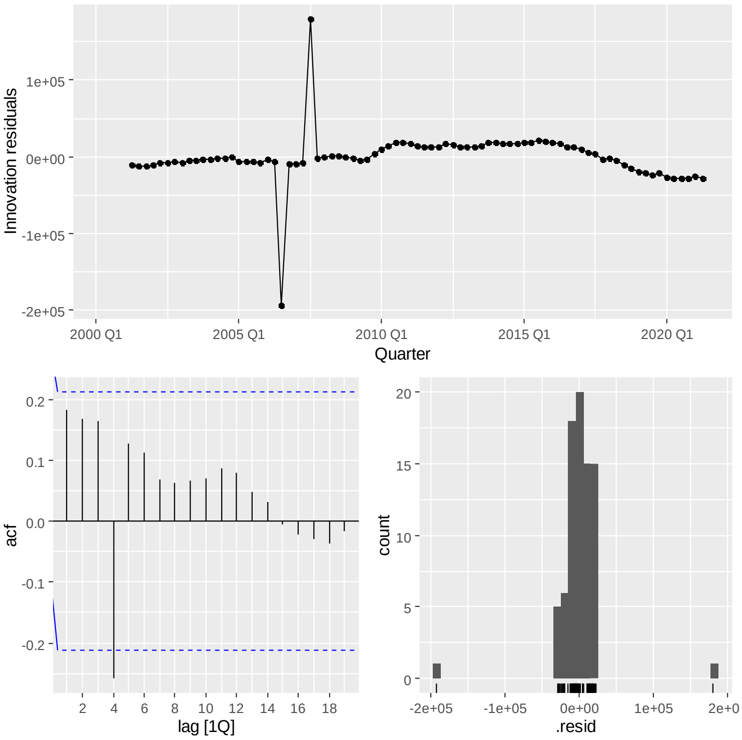


The data has seasonality and trend within the data. The SNAIVE model with drift is the appropriate model.

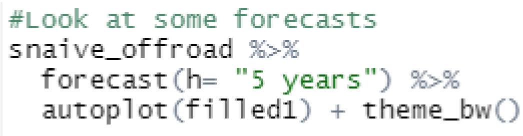
b) Apply seasonal naïve to off road series



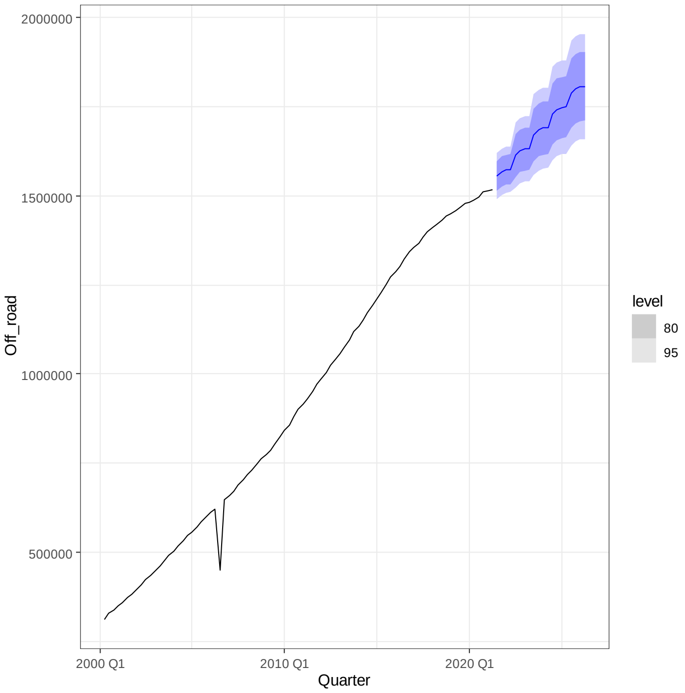
```
## off_road

#fill missing values
filled1 <- missing %>%
  model(ARIMA(off_road)) %>%
  interpolate(missing)

#Apply seasonal naive method to off_road series with drift
snaive_offroad <- filled1 %>%
  model(SNAIVE(off_road ~ drift()))

#Check residuals
snaive_offroad %>% gg_tsresiduals()
```



The residual plot has a spike at lag 4, causing the model to not be white-noise. The other lags stay within the range and are not significant. The residuals do not appear to be normally distributed as a result of 2 large outliers in either direction.

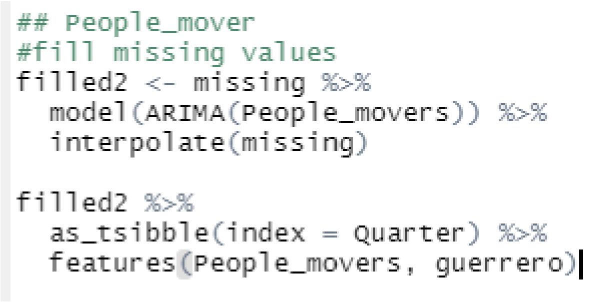


```
#Look at some forecasts
snaive_offroad %>%
  forecast(h= "5 years") %>%
  autoplot(filled1) + theme_bw()
```



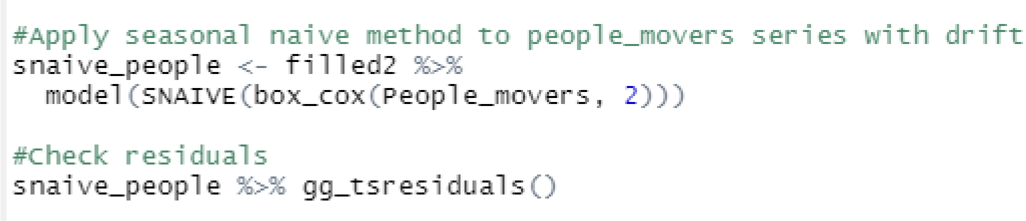
The SNAIVE model with drift follows the trend quite well as it follows the same path. There seems to be more seasonality using the SNAIVE model which doesn’t mirror the past data. Using a NAÏVE model could be more accurate.

c) Apply seasonal naïve to people mover series

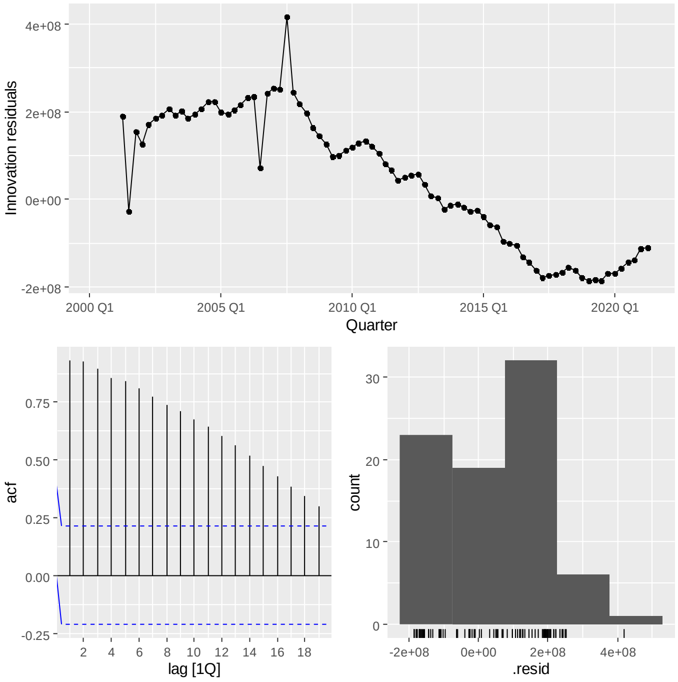


```
## People_mover
#fill missing values
filled2 <- missing %>%
  model(ARIMA(People_movers)) %>%
  interpolate(missing)

filled2 %>%
  as_tsibble(index = Quarter) %>%
  features(People_movers, guerrero)
```

Guerrero recommend a lambda of 2

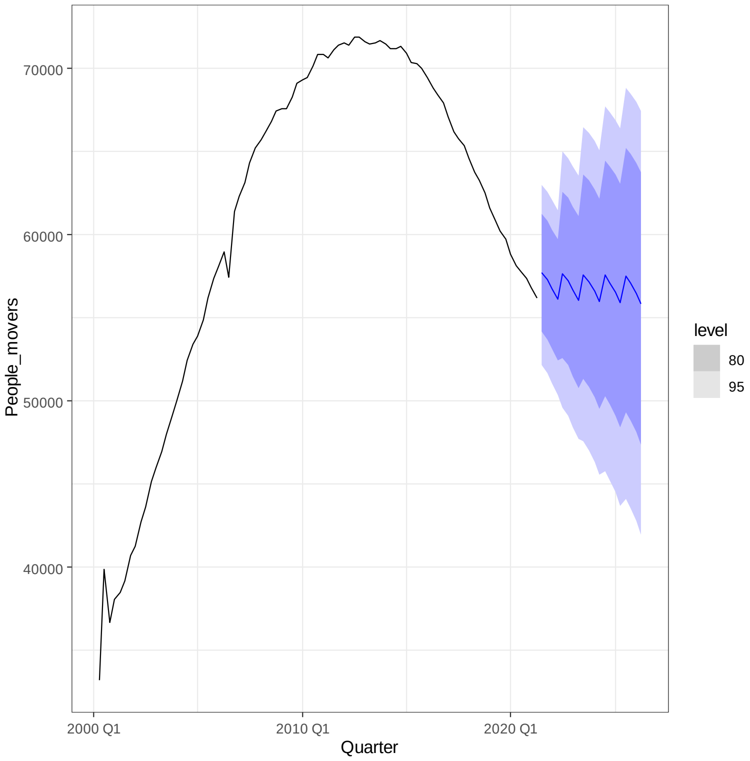


```
#Apply seasonal naive method to people_movers series with drift
snaive_people <- filled2 %>%
  model(SNAIVE(box_cox(People_movers, 2)))

#Check residuals
snaive_people %>% gg_tsresiduals()
```



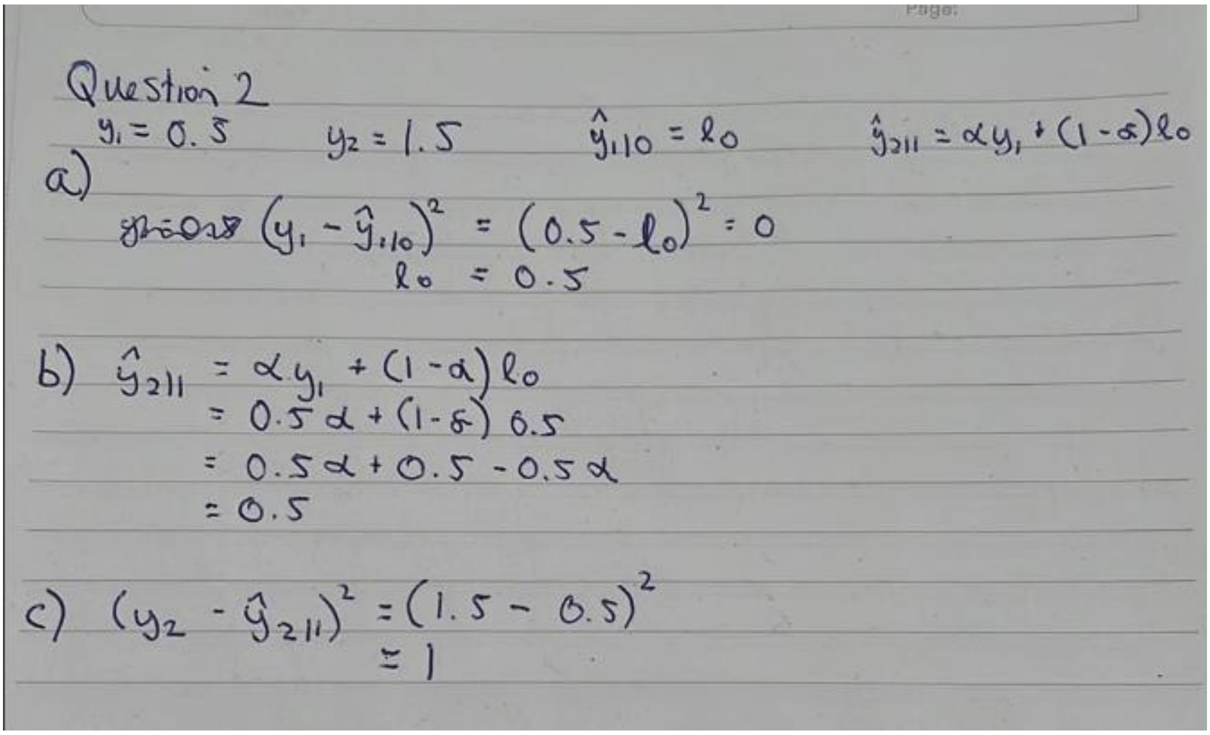
From the residuals we can see there is a lot of autocorrelations in the residuals but this decreases over the lags. The residuals are not normally distributed and not centred around 0. This is likely from the parabolic curve of the and failing to capture the negative trend in the later section of the data.

```
#Look at some forecasts
snaive_people %>%
  forecast(h= "5 years") %>%
  autoplot(filled2) + theme_bw()
```



The model captures the possible variation in the model with larger confidence interval. This makes sense because of the size of the volatility. The mean is centred in the middle of the interval. The SNAIVE model seems to fluctuate more than the actual data, NAÏVE method might be more appropriate.

Question 2



Question 2

$y_1 = 0.5$ $y_2 = 1.5$ $\hat{y}_{1|0} = \ell_0$ $\hat{y}_{2|1} = \alpha y_1 + (1-\alpha)\ell_0$

a)

$(y_1 - \hat{y}_{1|0})^2 = (0.5 - \ell_0)^2 = 0$

$\ell_0 = 0.5$

b) $\hat{y}_{2|1} = \alpha y_1 + (1-\alpha)\ell_0$

$= 0.5\alpha + (1-\alpha)\,0.5$

$= 0.5\alpha + 0.5 - 0.5\alpha$

$= 0.5$

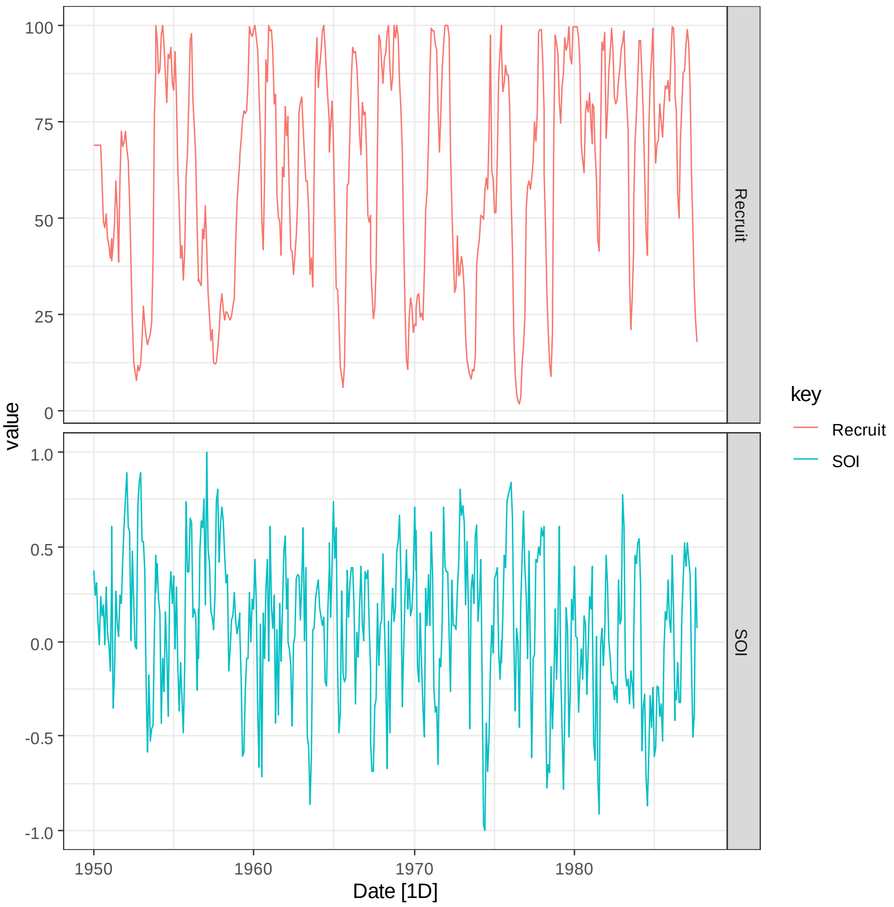
c) $(y_2 - \hat{y}_{2|1})^2 = (1.5 - 0.5)^2$

$= 1$

Question 3

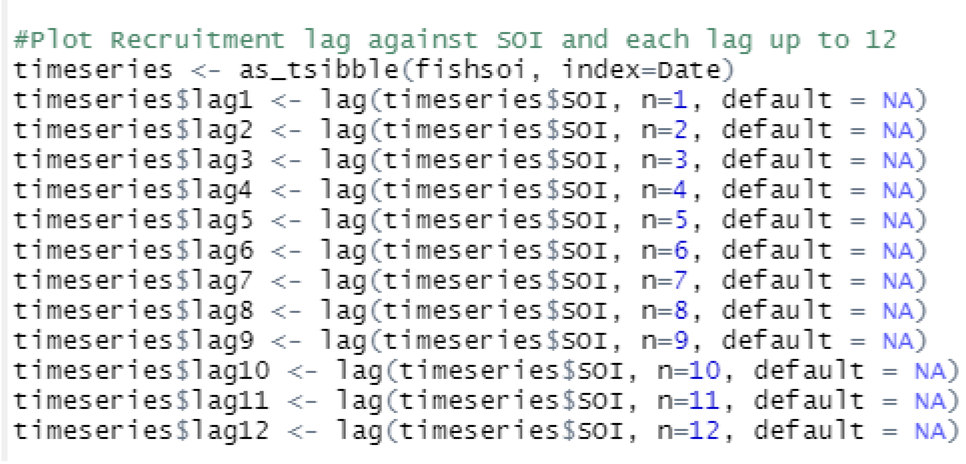
a) Produce time series plots of SOI and Recruitment

```
fishsoi <- readr::read_csv("FISHSOI.csv")
date <- as.Date("1950/01/01")
len <- 453
dates <- seq(date, by= "month", length.out = len)
fishsoi$Date <- dates

#time series plot for SOI and Recruitment
timeseries <- as_tsibble(fishsoi, index=Date)

timeseries %>%
  pivot_longer(1:2, names_to="key", values_to="value") %>%
  autoplot(.vars = value) +
  facet_grid(vars(key), scales = "free_y") + theme_bw()
```

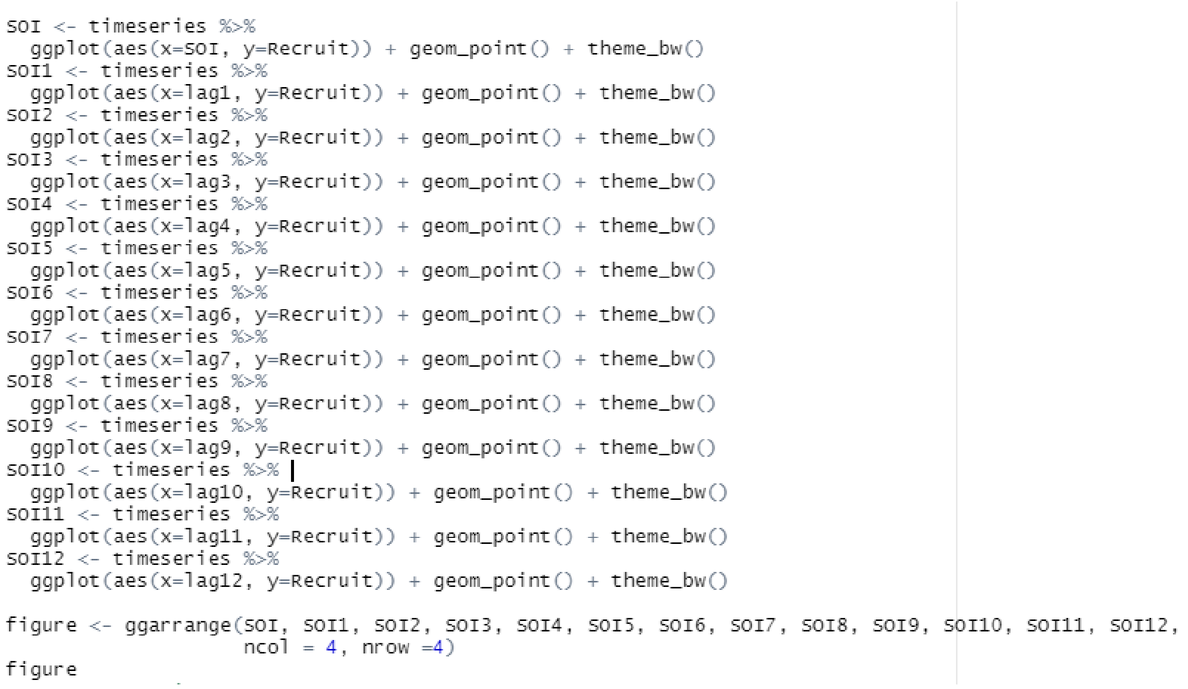


The plot for recruitment appears to be less volatile, it has higher peaks and troughs with changes being a lot slower over time. The plot for SOI seems to be more volatile, it has smaller peaks and troughs. SOI seems to have more seasonality.

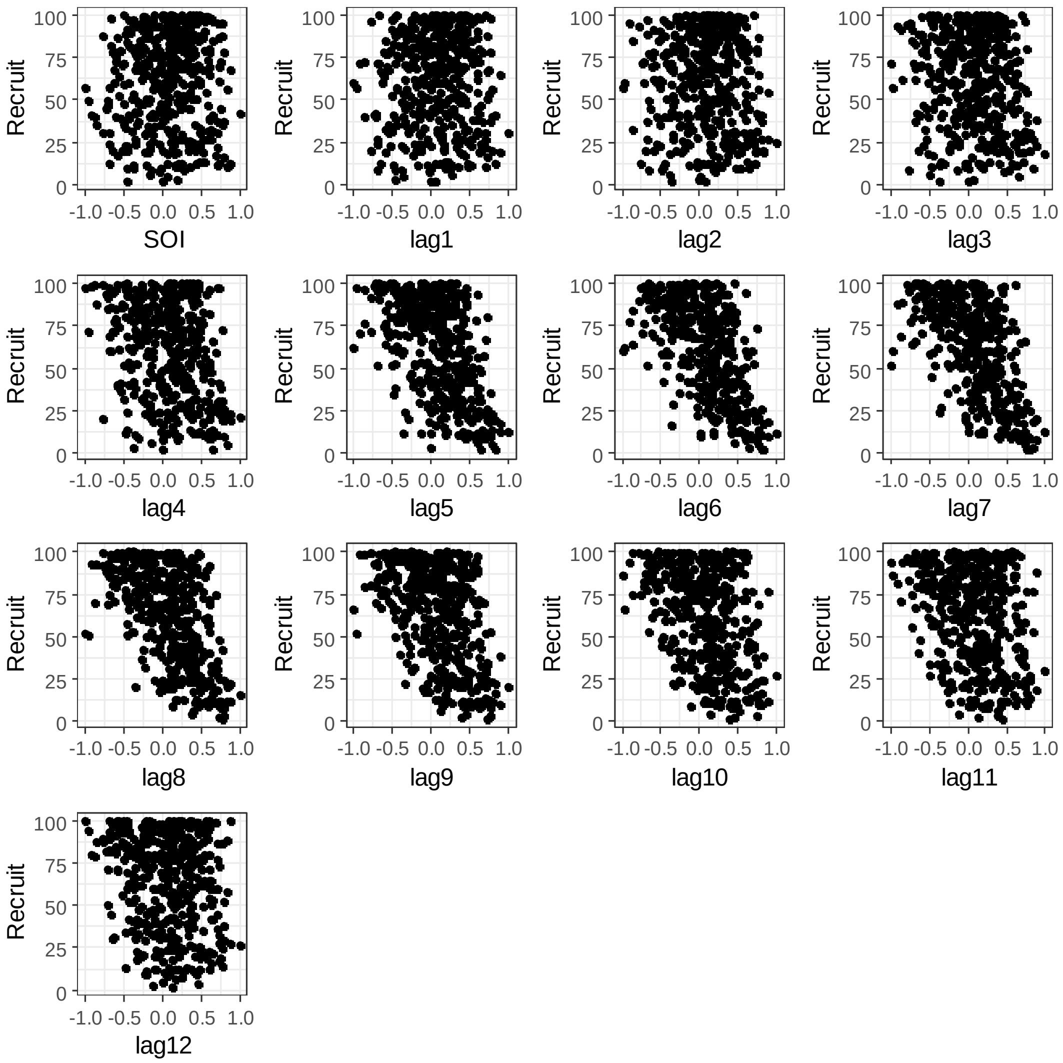
b) Plot recruitment against SOI, and each lag of SOI



```
#Plot Recruitment lag against SOI and each lag up to 12
timeseries <- as_tsibble(fishsoi, index=Date)
timeseries$lag1 <- lag(timeseries$SOI, n=1, default = NA)
timeseries$lag2 <- lag(timeseries$SOI, n=2, default = NA)
timeseries$lag3 <- lag(timeseries$SOI, n=3, default = NA)
timeseries$lag4 <- lag(timeseries$SOI, n=4, default = NA)
timeseries$lag5 <- lag(timeseries$SOI, n=5, default = NA)
timeseries$lag6 <- lag(timeseries$SOI, n=6, default = NA)
timeseries$lag7 <- lag(timeseries$SOI, n=7, default = NA)
timeseries$lag8 <- lag(timeseries$SOI, n=8, default = NA)
timeseries$lag9 <- lag(timeseries$SOI, n=9, default = NA)
timeseries$lag10 <- lag(timeseries$SOI, n=10, default = NA)
timeseries$lag11 <- lag(timeseries$SOI, n=11, default = NA)
timeseries$lag12 <- lag(timeseries$SOI, n=12, default = NA)
```

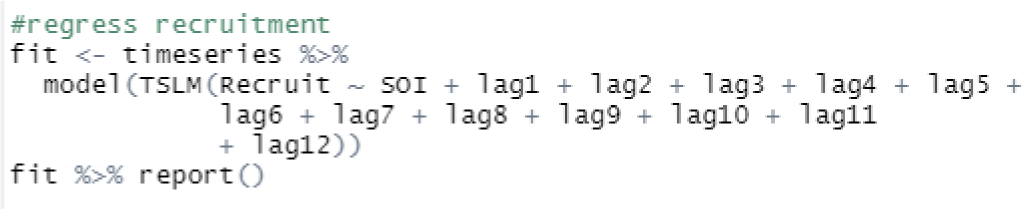


```
SOI <- timeseries %>%
  ggplot(aes(x=SOI, y=Recruit)) + geom_point() + theme_bw()
SOI1 <- timeseries %>%
  ggplot(aes(x=lag1, y=Recruit)) + geom_point() + theme_bw()
SOI2 <- timeseries %>%
  ggplot(aes(x=lag2, y=Recruit)) + geom_point() + theme_bw()
SOI3 <- timeseries %>%
  ggplot(aes(x=lag3, y=Recruit)) + geom_point() + theme_bw()
SOI4 <- timeseries %>%
  ggplot(aes(x=lag4, y=Recruit)) + geom_point() + theme_bw()
SOI5 <- timeseries %>%
  ggplot(aes(x=lag5, y=Recruit)) + geom_point() + theme_bw()
SOI6 <- timeseries %>%
  ggplot(aes(x=lag6, y=Recruit)) + geom_point() + theme_bw()
SOI7 <- timeseries %>%
  ggplot(aes(x=lag7, y=Recruit)) + geom_point() + theme_bw()
SOI8 <- timeseries %>%
  ggplot(aes(x=lag8, y=Recruit)) + geom_point() + theme_bw()
SOI9 <- timeseries %>%
  ggplot(aes(x=lag9, y=Recruit)) + geom_point() + theme_bw()
SOI10 <- timeseries %>%
  ggplot(aes(x=lag10, y=Recruit)) + geom_point() + theme_bw()
SOI11 <- timeseries %>%
  ggplot(aes(x=lag11, y=Recruit)) + geom_point() + theme_bw()
SOI12 <- timeseries %>%
  ggplot(aes(x=lag12, y=Recruit)) + geom_point() + theme_bw()

figure <- ggarrange(SOI, SOI1, SOI2, SOI3, SOI4, SOI5, SOI6, SOI7, SOI8, SOI9, SOI10, SOI11, SOI12,
                    ncol = 4, nrow =4)
figure
```

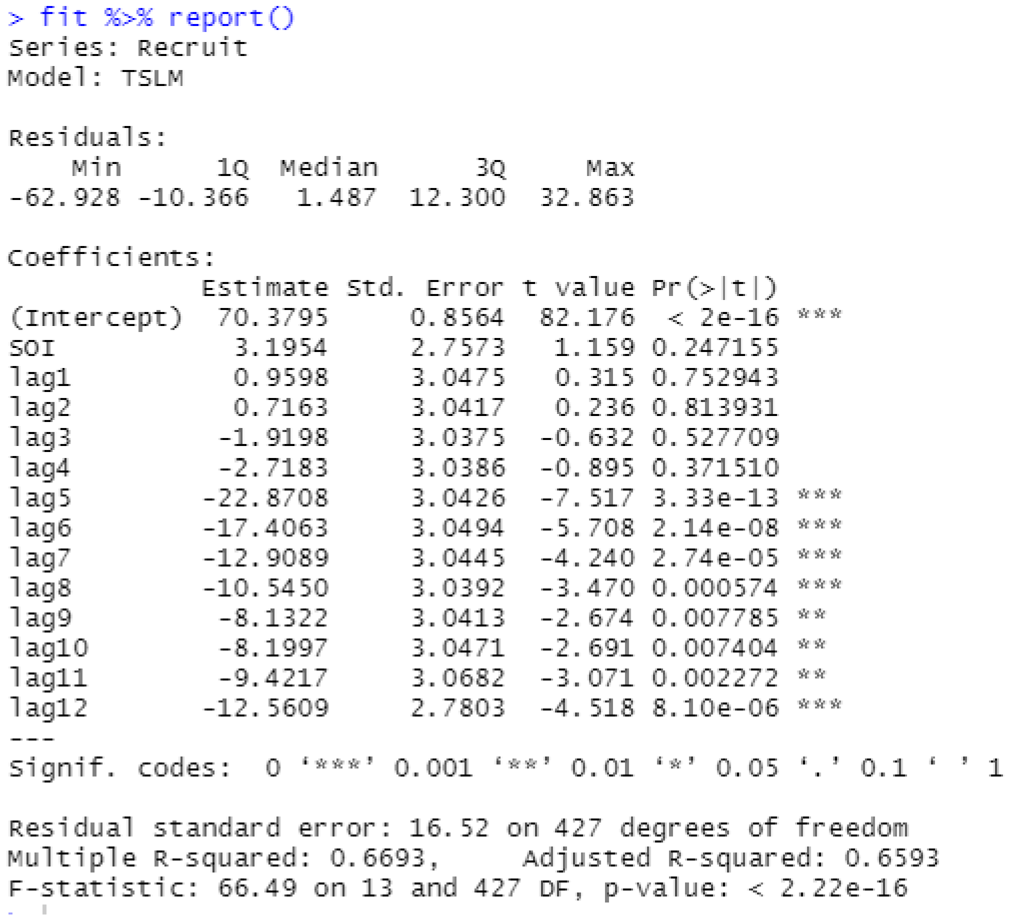


The scatterplots seem pretty random in for SOI, and lags 1 to 4, meaning there isn't much correlation between the variables. From lags 5 to lag 12 there seems to be a relatively strong negative relationship between recruitment and SOI.

c) Regress recruitment on SOI and each lag of SOI up to lag 12. Assess model fit.



```
#regress recruitment
fit <- timeseries %>%
  model(TSLM(Recruit ~ SOI + lag1 + lag2 + lag3 + lag4 + lag5 +
               lag6 + lag7 + lag8 + lag9 + lag10 + lag11
               + lag12))
fit %>% report()
```

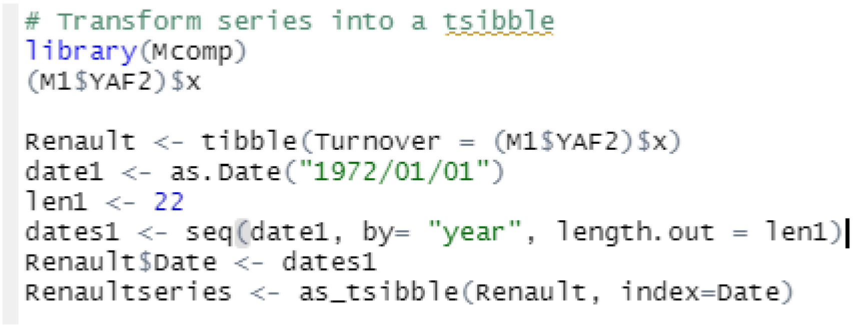


```
> fit %>% report()
Series: Recruit
Model: TSLM

Residuals:
    Min      1Q  Median      3Q     Max
-62.928 -10.366   1.487  12.300  32.863

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  70.3795     0.8564  82.176  < 2e-16 ***
SOI           3.1954     2.7573   1.159 0.247155
lag1          0.9598     3.0475   0.315 0.752943
lag2          0.7163     3.0417   0.236 0.813931
lag3         -1.9198     3.0375  -0.632 0.527709
lag4         -2.7183     3.0386  -0.895 0.371510
lag5        -22.8708     3.0426  -7.517 3.33e-13 ***
lag6        -17.4063     3.0494  -5.708 2.14e-08 ***
lag7        -12.9089     3.0445  -4.240 2.74e-05 ***
lag8        -10.5450     3.0392  -3.470 0.000574 ***
lag9         -8.1322     3.0413  -2.674 0.007785 **
lag10        -8.1997     3.0471  -2.691 0.007404 **
lag11        -9.4217     3.0682  -3.071 0.002272 **
lag12       -12.5609     2.7803  -4.518 8.10e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.52 on 427 degrees of freedom
Multiple R-squared: 0.6693,	Adjusted R-squared: 0.6593
F-statistic: 66.49 on 13 and 427 DF, p-value: < 2.22e-16
```

The results from the linear regression show that the lagged variables are useful for predicting future values of recruitment. Lag 5 to Lag12 is statistically significant for the prediction, whereas SOI and Lag1 to Lag4. This regression account for 66.93% of the variation within the data. There is still 32.07% of variation that is unexplained. This could be improved by adding other variables in the climate that would affect level of recruitment.

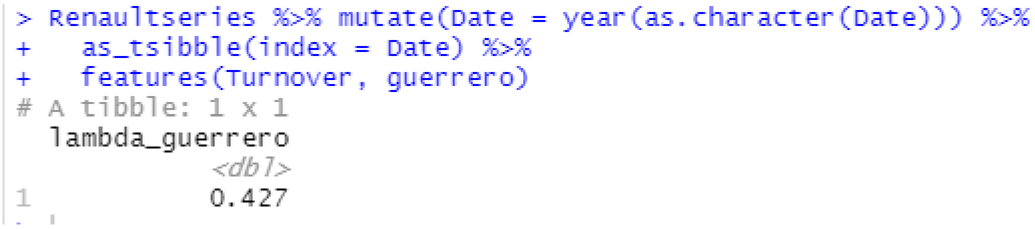
Question 4

a) Transform into tsibble data



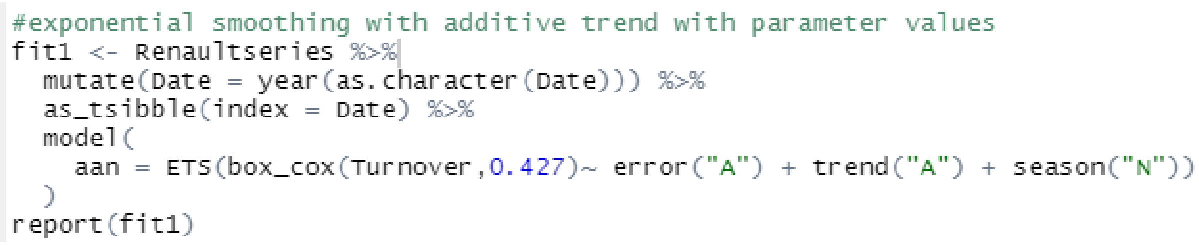
```
# Transform series into a tsibble
library(Mcomp)
(M1$YAF2)$x

Renault <- tibble(Turnover = (M1$YAF2)$x)
date1 <- as.Date("1972/01/01")
len1 <- 22
dates1 <- seq(date1, by= "year", length.out = len1)
Renault$Date <- dates1
Renaultseries <- as_tsibble(Renault, index=Date)
```

b) Use exponential smoothing with additive trend, ETS(A,A,N). write down the fitted component form with the estimated parameter values

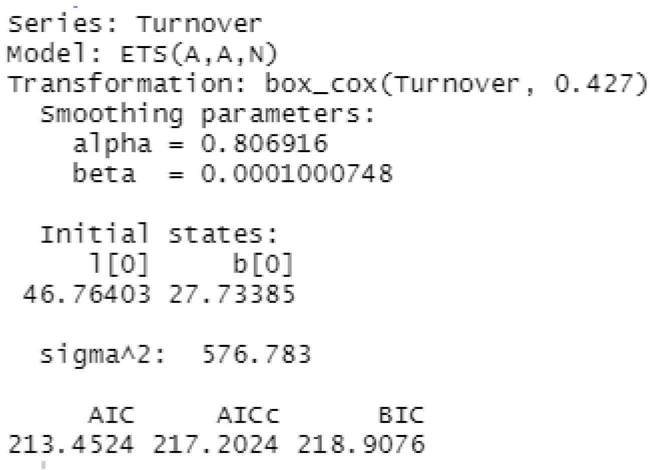


```
> Renaultseries %>% mutate(Date = year(as.character(Date))) %>%
+   as_tsibble(index = Date) %>%
+   features(Turnover, guerrero)
# A tibble: 1 x 1
  lambda_guerrero
            <dbl>
1           0.427
```

Due to the original dataset being fairly exponential, a strong transformation is needed to account for this increasing variance. The Guerrero method recommended using a lambda of 0.427

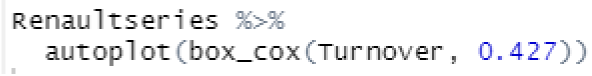


```
#exponential smoothing with additive trend with parameter values
fit1 <- Renaultseries %>%
  mutate(Date = year(as.character(Date))) %>%
  as_tsibble(index = Date) %>%
  model(
    aan = ETS(box_cox(Turnover,0.427)~ error("A") + trend("A") + season("N"))
  )
report(fit1)
```

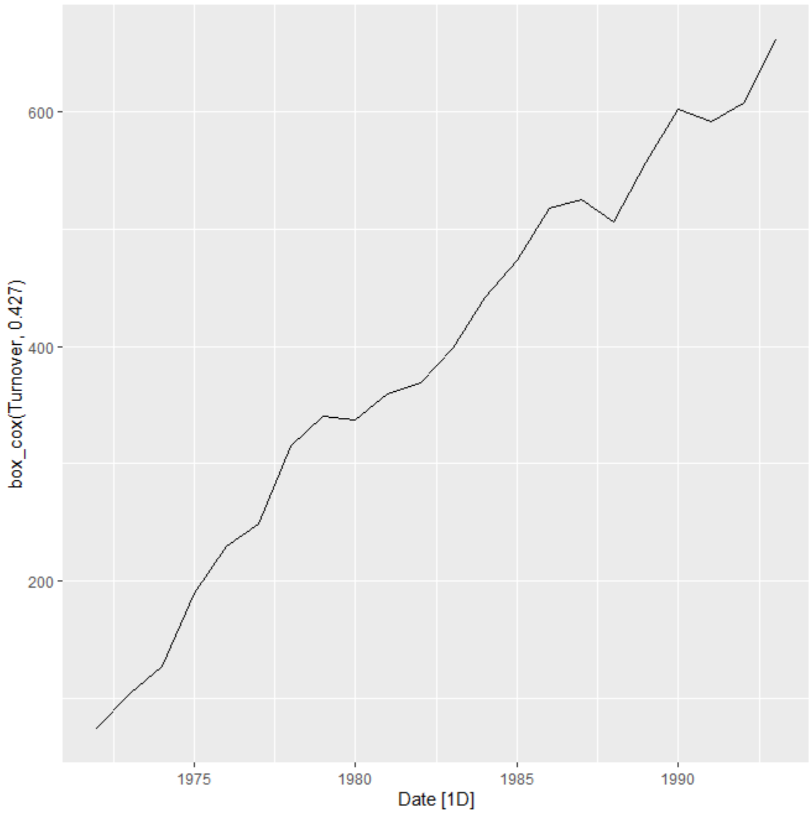


```
Series: Turnover
Model: ETS(A,A,N)
Transformation: box_cox(Turnover, 0.427)
  Smoothing parameters:
    alpha = 0.806916
    beta  = 0.0001000748

  Initial states:
     l[0]     b[0]
 46.76403 27.73385

  sigma^2:  576.783

     AIC     AICc      BIC
213.4524 217.2024 218.9076
```

The ETS(A,A,N) model uses additive error, additive trend, and no seasonality. The box cox transformation is included in the model to reduce the amount of variance. The selected alpha value can be seen, the value is closer to 1 than 0 meaning that more recent values have higher weight on the series. The beta value is relatively low meaning there is a larger focus on longer-term trend. Initial states of l0 and b0 can be seen from the output. Variance is shown from sigma2 which is significantly less than when using no transformation. AIC value of 213 was the highest AIC I could achieve compared to other transformations tested.

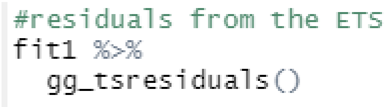


```
Renaultseries %>%
  autoplot(box_cox(Turnover, 0.427))
```

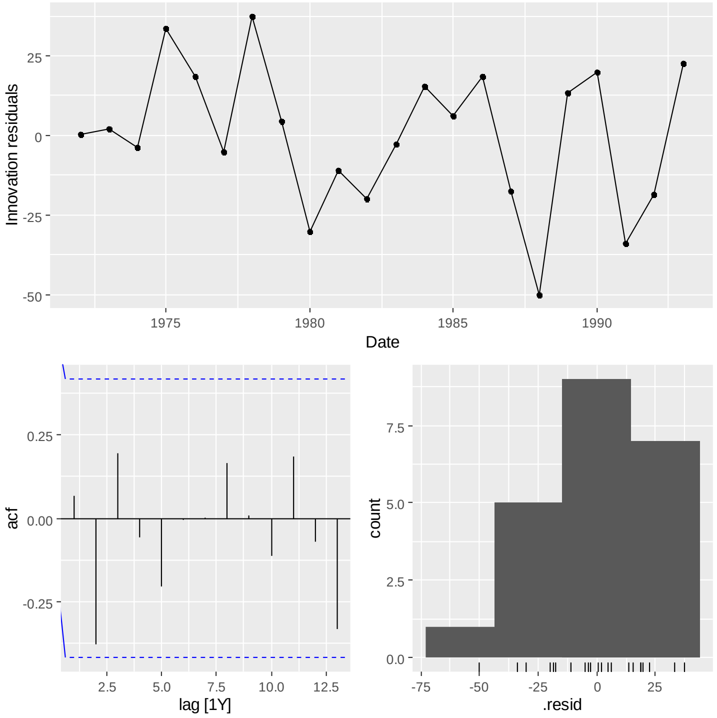


The plot shows the line after transformation which is a lot smoother than the original time-series.

c) Do the residuals from the fitted ETS(A,A,N) look like whitenoise?

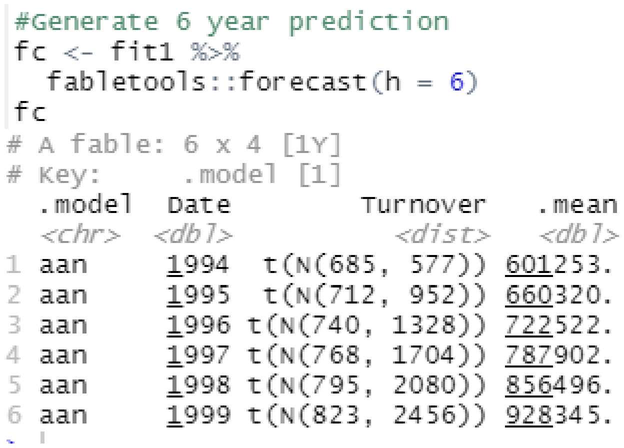


```
#residuals from the ETS
fit1 %>%
  gg_tsresiduals()
```

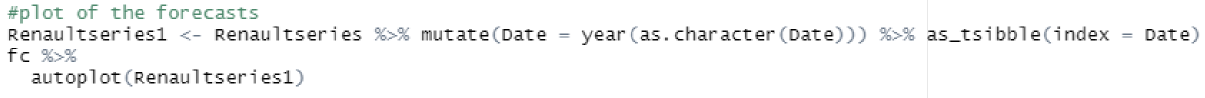


The ACF plot shows there are no significant lags in the data meaning the no lag exceeds the blue line threshold. The data appears to follow white noise. The distribution does not appear to be normal as it is left tailed, however this could be due to little variance in the data, meaning there is not enough of a spread.

d) Use the fitted ETS(A,A,N) to generate a 6-year prediction, illustrate with a plot

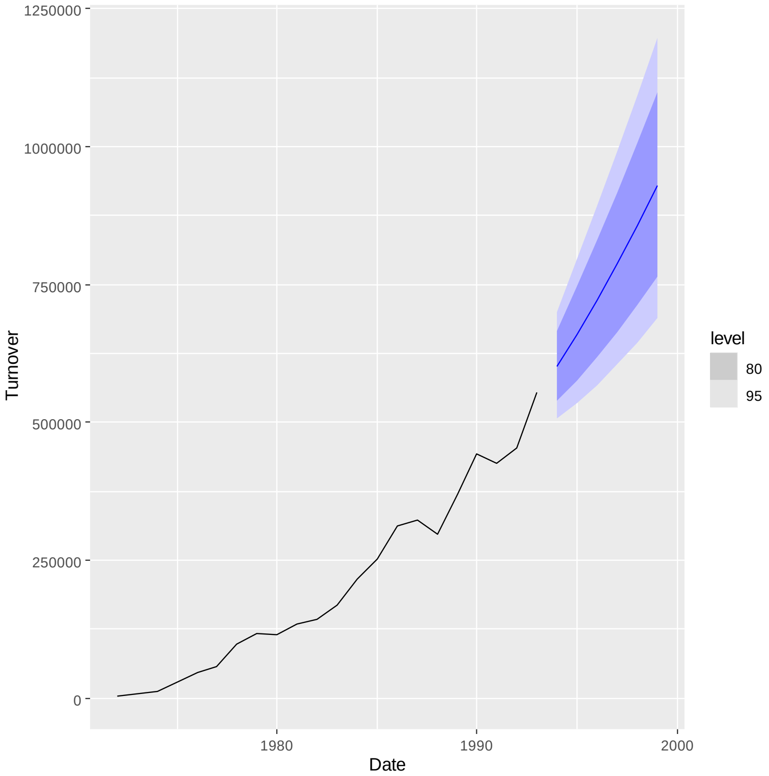


```
#Generate 6 year prediction
fc <- fit1 %>%
  fabletools::forecast(h = 6)
fc
# A fable: 6 x 4 [1Y]
# Key:     .model [1]
  .model  Date         Turnover    .mean
  <chr>   <dbl>          <dist>    <dbl>
1 aan     1994   t(N(685, 577)) 601253.
2 aan     1995   t(N(712, 952)) 660320.
3 aan     1996 t(N(740, 1328)) 722522.
4 aan     1997 t(N(768, 1704)) 787902.
5 aan     1998 t(N(795, 2080)) 856496.
6 aan     1999 t(N(823, 2456)) 928345.
```



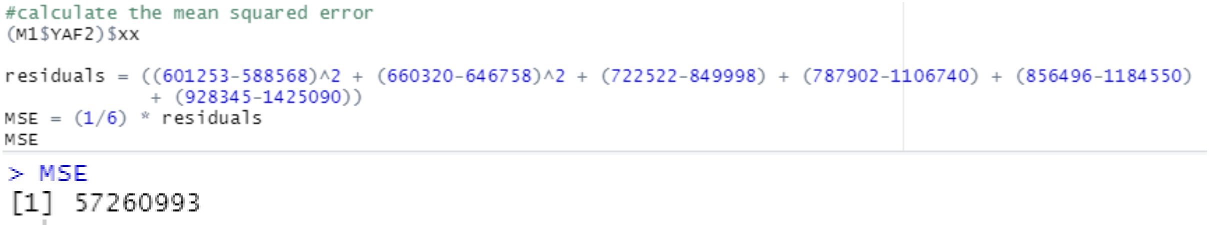
```
#plot of the forecasts
Renaultseries1 <- Renaultseries %>% mutate(Date = year(as.character(Date))) %>% as_tsibble(index = Date)
fc %>%
  autoplot(Renaultseries1)
```

The forecasted values can be seen from the .mean column which is calculated from the ETS(A,A,N) model.



The graph shows the mean value where the data is likely to follow, we can see that it is a relatively good fit and contains a confidence level range for where the data can lie within. The error bands seem fairly slim, so the data has been captured well.

e) Calculate the mean squared error, between the 6-year prediction and the true future data included



```
#calculate the mean squared error
(M1$YAF2)$xx

residuals = ((601253-588568)^2 + (660320-646758)^2 + (722522-849998) + (787902-1106740) + (856496-1184550)
             + (928345-1425090))
MSE = (1/6) * residuals
MSE
```

```
> MSE
[1] 57260993
```

The mean squared error is the mean difference between the predicted value and the actual value squared. The above code shows the calculation and the result of MSE = 57,260,993