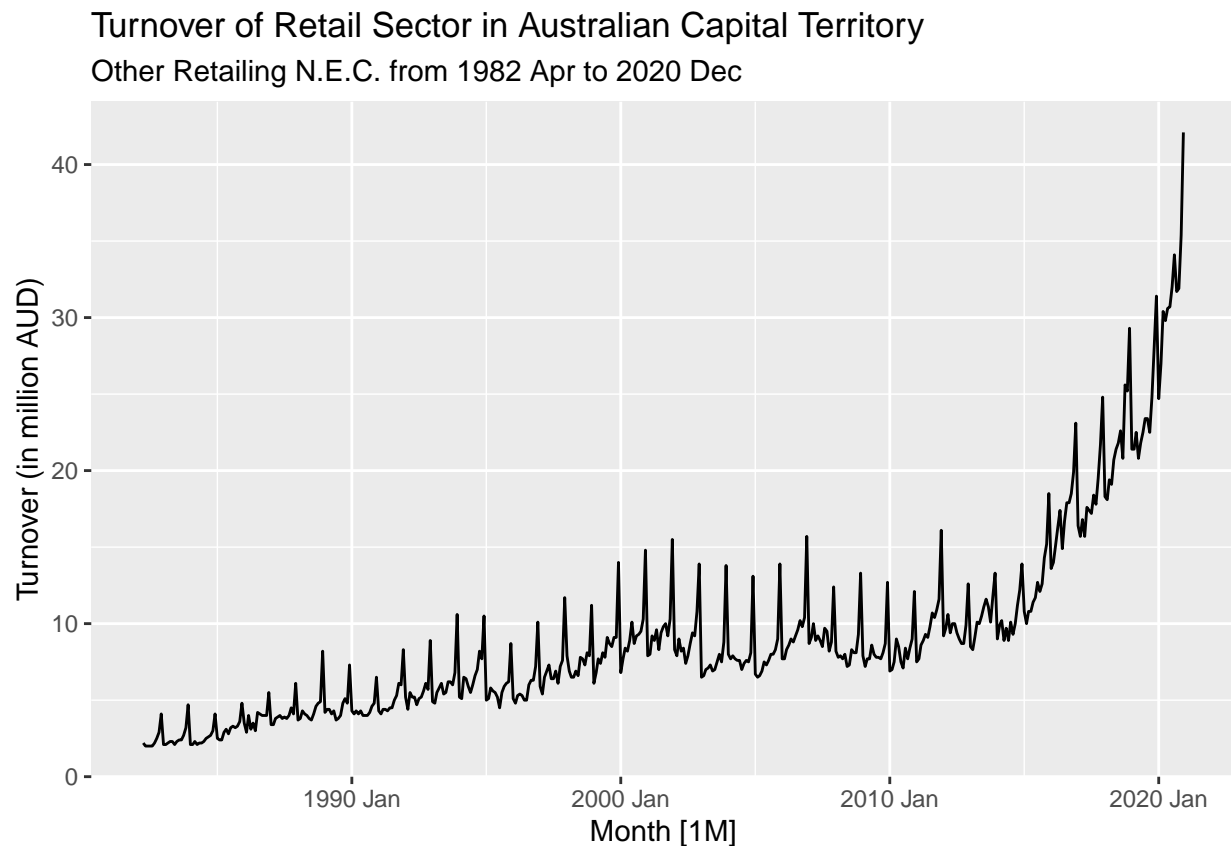


Forecasting Australia Other Retailing Sector in ACT

Masardi Rachman Rosyid

Data Inspection

```
my_series |>
  autoplot(Turnover) +
  labs(
    title = "Turnover of Retail Sector in Australian Capital Territory",
    subtitle = paste0(
      "Other Retailing N.E.C. from ",
      pull(my_series[1, 1]),
      " to ",
      pull(my_series[nrow(my_series), 1])
    ),
    y = "Turnover (in million AUD)"
  )
```



```
my_series_lambda <- my_series |>
  features(
    Turnover,
    guerrero
  ) |>
  pull()

round(my_series_lambda, 4)
```

```
## [1] 0.243
```

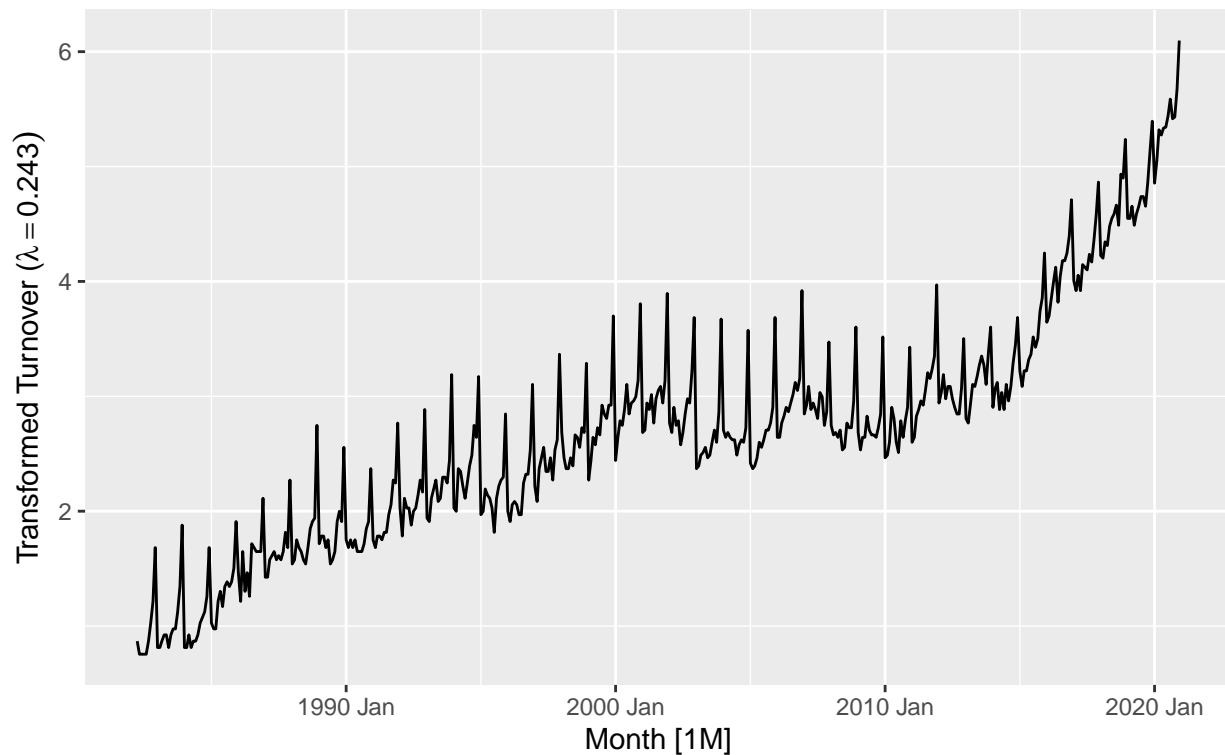
Observation

From the plot, it is observed a variation of seasonal components visually which a transformation may be necessary, previously the Guerrero method was chosen due to the ability to finely tweak the lambda $\lambda = 0.243$, the plot is also not stationary from the appearance of trend and seasonal components.

```
my_series |>
  autoplot(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    )
  ) +
  labs(
    title = "Transformed Turnover of Retail Sector in Australian Capital Territory",
    subtitle = paste0(
      "Box-Cox Transformed, Other Retailing N.E.C. from ",
      pull(my_series[1, 1]),
      " to ",
      pull(my_series[nrow(my_series), 1])
    ),
    y = TeX(
      paste0(
        "Transformed Turnover ( $\lambda = ",
        round(my_series_lambda, 3),
        ")"
      )
    )
  )
)$ 
```

Transformed Turnover of Retail Sector in Australian Capital Territory

Box-Cox Transformed, Other Retailing N.E.C. from 1982 Apr to 2020 Dec



Observation

The box-cox transformation with $\lambda = 0.243$ helps to stabilise the variation, however, this is one aspect that needs doing in ARIMA model. This has still not solved the stationary problem which is sorted on the next stage along with the KPSS test.

```
# Check if seasonal difference is required
my_series |>
  features(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    ),
    unitroot_nsdiffs
  )
```

```
## # A tibble: 1 x 1
##   nsdiffs
##   <int>
## 1     1
```

```
# Check if first difference is needed
my_series |>
  features(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    )
```

```

    ),
    unitroot_ndiffs
  )

```

```

## # A tibble: 1 x 1
##   ndiffs
##   <int>
## 1     1

```

```

# KPSS test
my_series |>
  features(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    ) |>
    difference(12) |>
    difference(),
    unitroot_kpss
  )

```

```

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1  0.0294     0.1

```

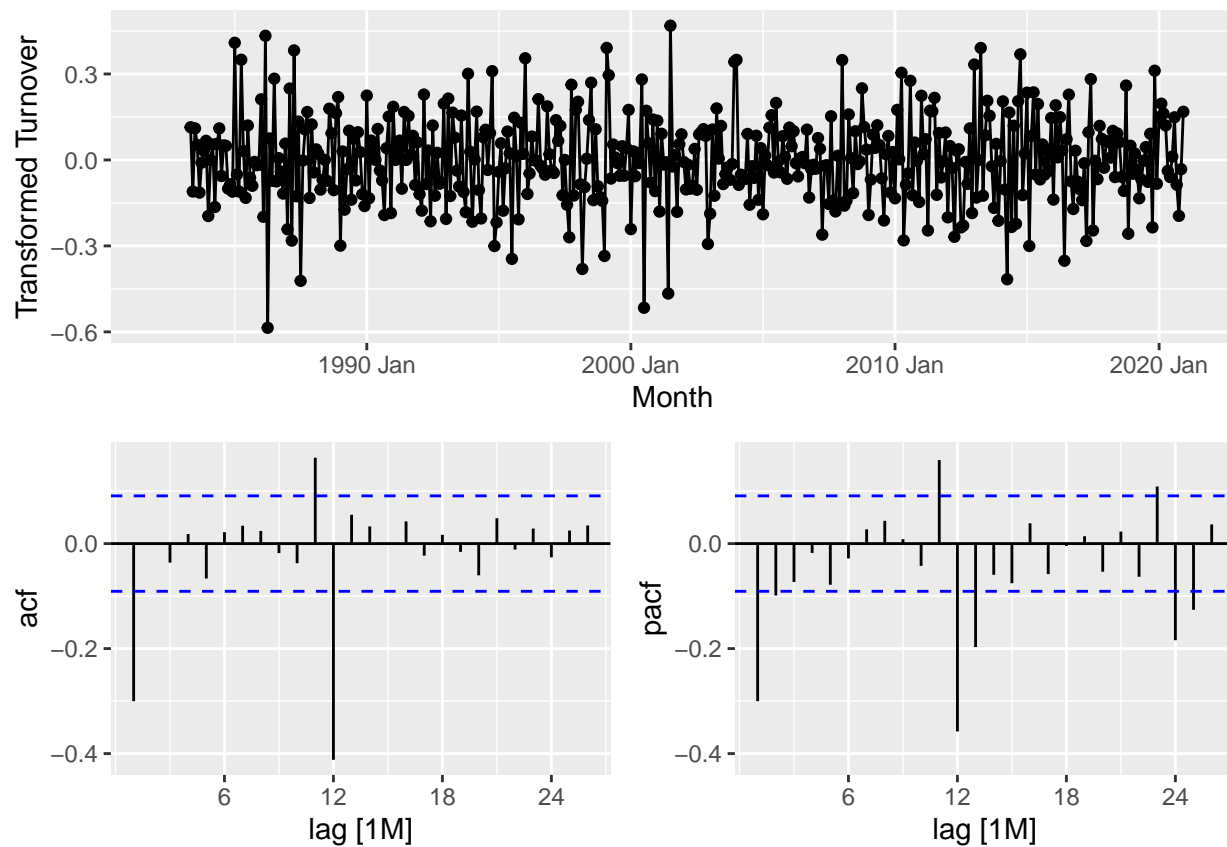
```

# Check stationary time-series
my_series |>
  gg_tsdisplay(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    ) |>
    difference(12) |>
    difference(),
    plot_type = "partial"
  ) +
  labs(
    y = "Transformed Turnover"
  )

```

```
## Warning: Removed 13 rows containing missing values ('geom_line()').
```

```
## Warning: Removed 13 rows containing missing values ('geom_point()').
```



Observation

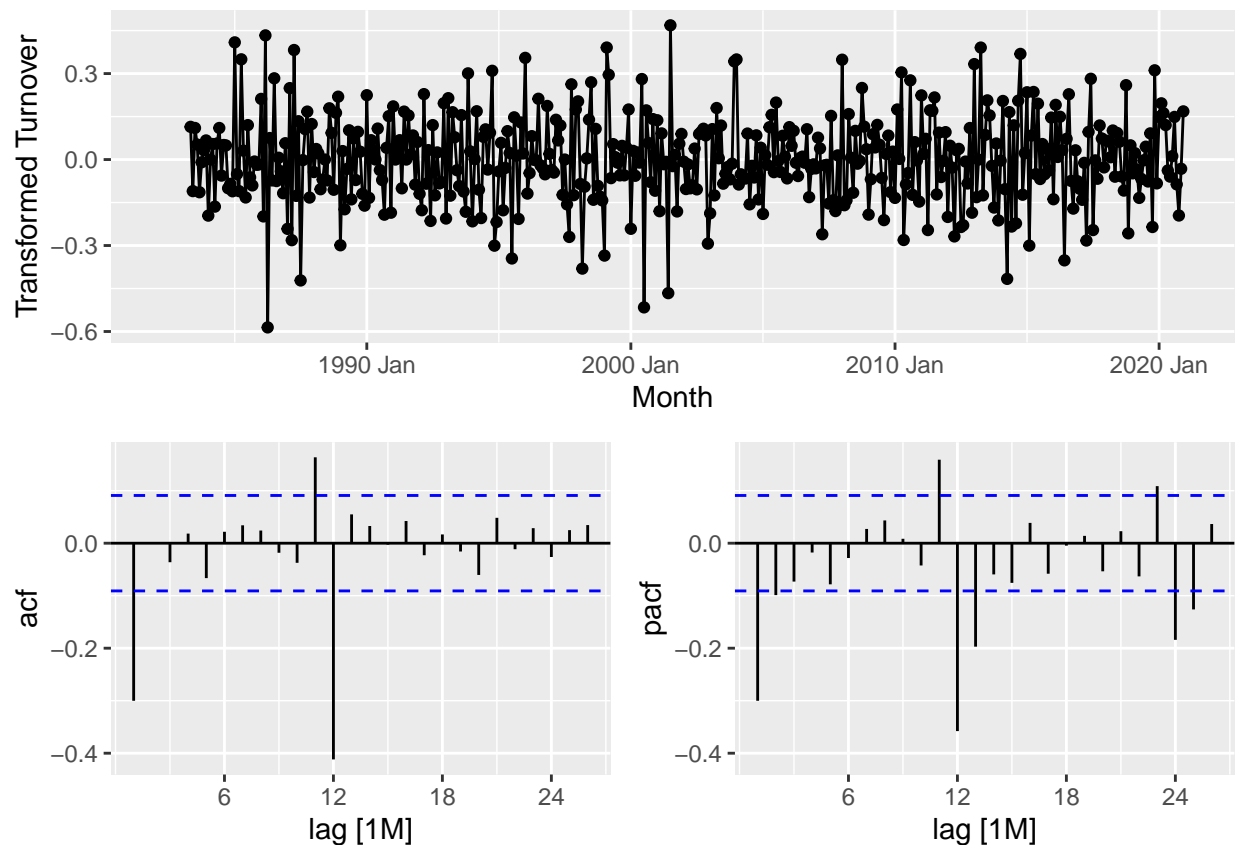
The appropriate sequence happens to be both first and seasonal difference from the unit root tests, this does look stationary confirmed with the *KPSS* test which returns a p -value $p = 0.1$ where the null hypothesis H_0 cannot be rejected that the data is stationary with 5% level of significance, the alternative H_1 is the data being not stationary.

ARIMA Model Specification from ACF and PACF plot

```
my_series |>
  gg_tsdisplay(
    box_cox(
      x = Turnover,
      lambda = my_series_lambda
    ) |>
    difference(12) |>
    difference(),
    plot_type = "partial"
  ) +
  labs(
    y = "Transformed Turnover"
  )
)
```

Warning: Removed 13 rows containing missing values ('geom_line()').

Warning: Removed 13 rows containing missing values ('geom_point()').



Observation

There are a few ARIMA models that could be used, one of them:

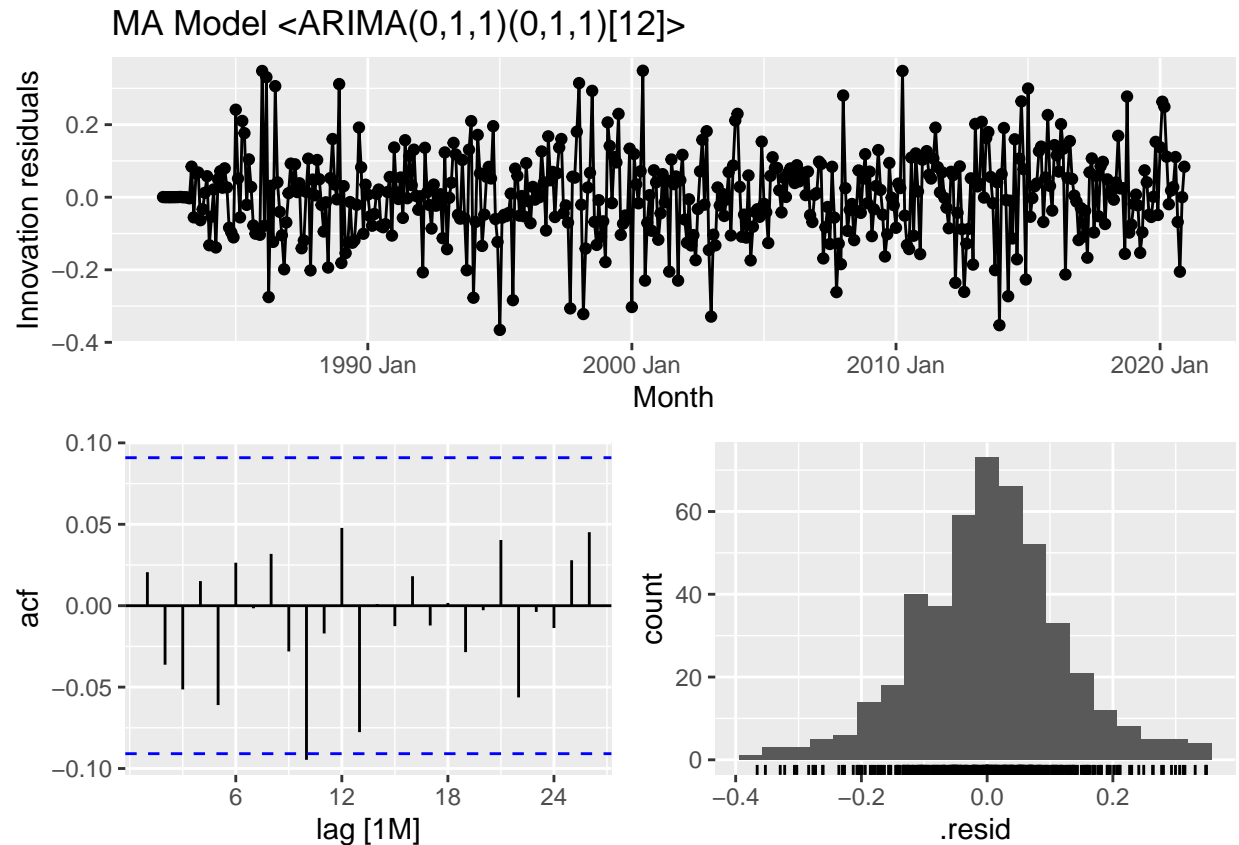
- $\text{ARIMA}(0,1,1)(0,1,1)_{12}$ from the ACF plot

The aim is to use a more simple model and the *MA* model shows one significant spike at the beginning (non-seasonal) and also one significant spike (seasonal at $lag = 12$). The rationale for the *MA* model is mainly from the ease of observation due to less significant spikes.

Residual White Noise Evaluation

```
observed_arima <- my_series |>
  model(
    arima_011_011 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 0, d = 1, q = 1) +
      PDQ(P = 0, D = 1, Q = 1)
    )
  )

# MA Model
observed_arima |>
  select(arima_011_011) |>
  gg_tsresiduals() +
  labs(
    title = paste0(
      "MA Model ",
      pull(
        observed_arima |>
          select(arima_011_011)
      )
    )
  )
```

```
observed_arima |>
  select(arima_011_011) |>
  augment() |>
  features(
    .innov,
    ljung_box,
    dof = 2,
    lag = 12 * 2
  )
```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>      <dbl>    <dbl>
## 1 arima_011_011  16.7      0.781
```

Observation

ARIMA(0,1,1)(0,1,1)₁₂

The residuals fluctuates around zero, visually homoscedastic, uncorrelated and is a normal distribution. Regardless of one significant spike, Ljung-Box test shows $p = 0.781$ where we accept H_0 where residuals are white noise at 5% significance level (H_1 is residual being auto-correlated). This suggests that it is a fit model for forecasting.

ARIMA Model Alternative Models

```
arma_alternatives <- my_series |>
  model(
    arima_210_210 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 2, d = 1, q = 0) +
      PDQ(P = 2, D = 1, Q = 0)
    ),
    arima_110_210 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 1, d = 1, q = 0) +
      PDQ(P = 2, D = 1, Q = 0)
    ),
    arima_210_110 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 2, d = 1, q = 0) +
      PDQ(P = 1, D = 1, Q = 0)
    ),
    arima_011_011 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 0, d = 1, q = 1) +
      PDQ(P = 0, D = 1, Q = 1)
    )
  )

arma_alternatives |>
  glance() |>
  arrange(AICc)
```

```
## # A tibble: 4 x 8
##   .model      sigma2 log_lik   AIC   AICc   BIC ar_roots  ma_roots
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
## 1 arima_011_011 0.0145   311.  -616. -616. -604. <cpl [0]> <cpl [13]>
## 2 arima_210_210 0.0166   285.  -559. -559. -539. <cpl [26]> <cpl [0]>
## 3 arima_110_210 0.0168   282.  -556. -556. -539. <cpl [25]> <cpl [0]>
## 4 arima_210_110 0.0176   271.  -535. -535. -518. <cpl [14]> <cpl [0]>
```

Observation

- ARIMA(0,1,1)(0,1,1)₁₂

- This is the *MA* model from previous question as a comparison to the alternatives, based on the significant spikes from the ACF plot, one significant spike for the non-seasonal $lag = 1$ and one significant spike from the seasonal $lag = 12$.
- $ARIMA(2,1,0)(2,1,0)_{12}$
 - The *AR* model based on the possibly two significant spikes on the non-seasonal part from the PACF plot and two significant spikes on the seasonal part (lag 12 and 24).
- $ARIMA(1,1,0)(2,1,0)_{12}$
 - This is based on $ARIMA(2,1,0)(2,1,0)_{12}$, however, since the second lag for the non-seasonal component is barely significant visually, assume that it might not be significant.
- $ARIMA(2,1,0)(1,1,0)_{12}$
 - This is based on the $ARIMA(2,1,0)(2,1,0)_{12}$, however, since the seasonal lag is slowly decaying, an attempt to use the first significant spike for the seasonal component is made.

As all models have the same order of differencing, comparing the AICc is appropriate with a target that is to minimise AICc. It can be observed that the first model that is $ARIMA(0,1,1)(0,1,1)_{12}$, is the most appropriate model with $AICc = -616.188$, therefore it is the chosen model for forecasting.

ARIMA Stepwise and Exhaustive Model

```
# If this variable exists, this WILL NOT run, it takes a long time
# from exhaustive search for the ARIMA model
if (!exists("arima_comparison")) {
  arima_comparison <- my_series |>
  model(
    arima_210_210 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 2, d = 1, q = 0) +
      PDQ(P = 2, D = 1, Q = 0)
    ),
    arima_110_210 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 1, d = 1, q = 0) +
      PDQ(P = 2, D = 1, Q = 0)
    ),
    arima_011_011 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 0, d = 1, q = 1) +
      PDQ(P = 0, D = 1, Q = 1)
    ),
    stepwise = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      )
    ),
    exhaustive = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ),
      stepwise = FALSE,
      greedy = FALSE,
      approximation = FALSE
    )
  )

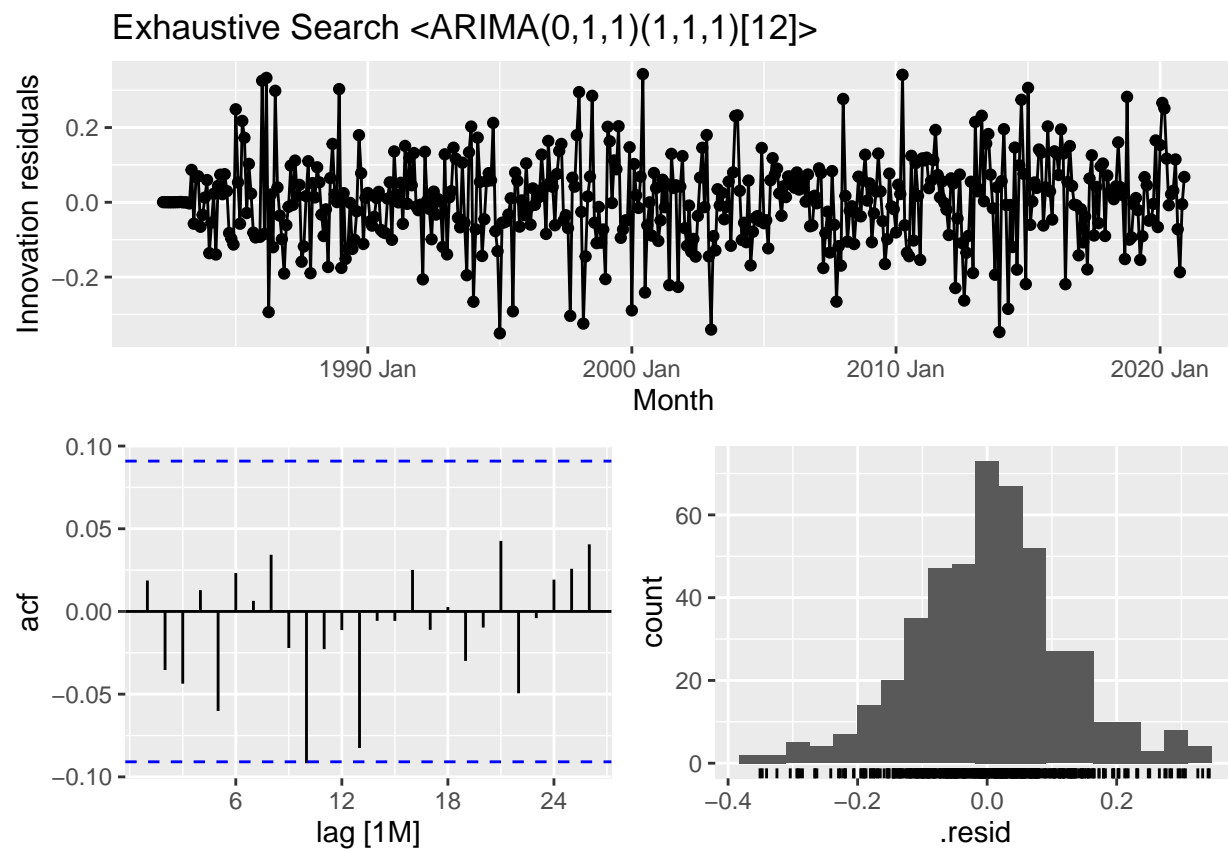
  # Save it to an RDS file so it can be loaded again next time
  # Delete file to run this again or manually run the code above
  saveRDS(
    object = arima_comparison,
    file = here("arima_comparison.RDS")
  )
}
```

```

)
}

arima_comparison |>
  select(exhaustive) |>
  gg_tsresiduals() +
  labs(
    title = paste0(
      "Exhaustive Search ",
      pull(
        arima_comparison |>
          select(exhaustive)
      )
    )
  )
)

```



```

arima_comparison |>
  select(exhaustive) |>
  augment() |>
  features(
    .innov,
    ljung_box,
    dof = 3,
    lag = 12 * 2
  )
)

```

```
## # A tibble: 1 x 3
##   .model      lb_stat lb_pvalue
##   <chr>      <dbl>    <dbl>
## 1 exhaustive    15.3      0.810
```

Observation

The chosen model is the exhaustive search that is ARIMA(0,1,1)(1,1,1)₁₂, with the parameter $P = 1$ instead of zero from the previous chosen model, it improves the model regarding the whiteness of the residuals overall, residuals fluctuate around zero, visually homoscedastic and normally distributed. The residuals are more uncorrelated with Ljung-Box showing $p = 0.810$ that is higher than 0.781 previously that means not rejecting H_0 that is residuals are white noise and reject H_1 that is residuals are auto-correlated. It has a higher degree of freedom $dof = 3$ that is more complex, but a better fitness overall, this could be more appropriate for forecasting.

ARIMA Model Selection

```
arima_comparison |>
  glance() |>
  arrange(AICc)
```

```
## # A tibble: 5 x 8
##   .model      sigma2 log_lik   AIC  AICc   BIC ar_roots  ma_roots
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
## 1 exhaustive  0.0144    313. -617. -617. -601. <cpl [12]> <cpl [13]>
## 2 arima_011_011 0.0145    311. -616. -616. -604. <cpl [0]>  <cpl [13]>
## 3 stepwise    0.0144    313. -616. -616. -596. <cpl [1]>  <cpl [25]>
## 4 arima_210_210 0.0166    285. -559. -559. -539. <cpl [26]> <cpl [0]>
## 5 arima_110_210 0.0168    282. -556. -556. -539. <cpl [25]> <cpl [0]>
```

Observation

Using *AICc* to measure the level of fitness as they have the same order of differencing, the $ARIMA(0,1,1)(1,1,1)_{12}$ is chosen from its lowest *AICc*, it is however has a worse *BIC* to $ARIMA(0,1,1)(0,1,1)_{12}$, but as *BIC* aims to find a true model and it's rarely the case, *AICc* is used.

ACT Forecast using ARIMA

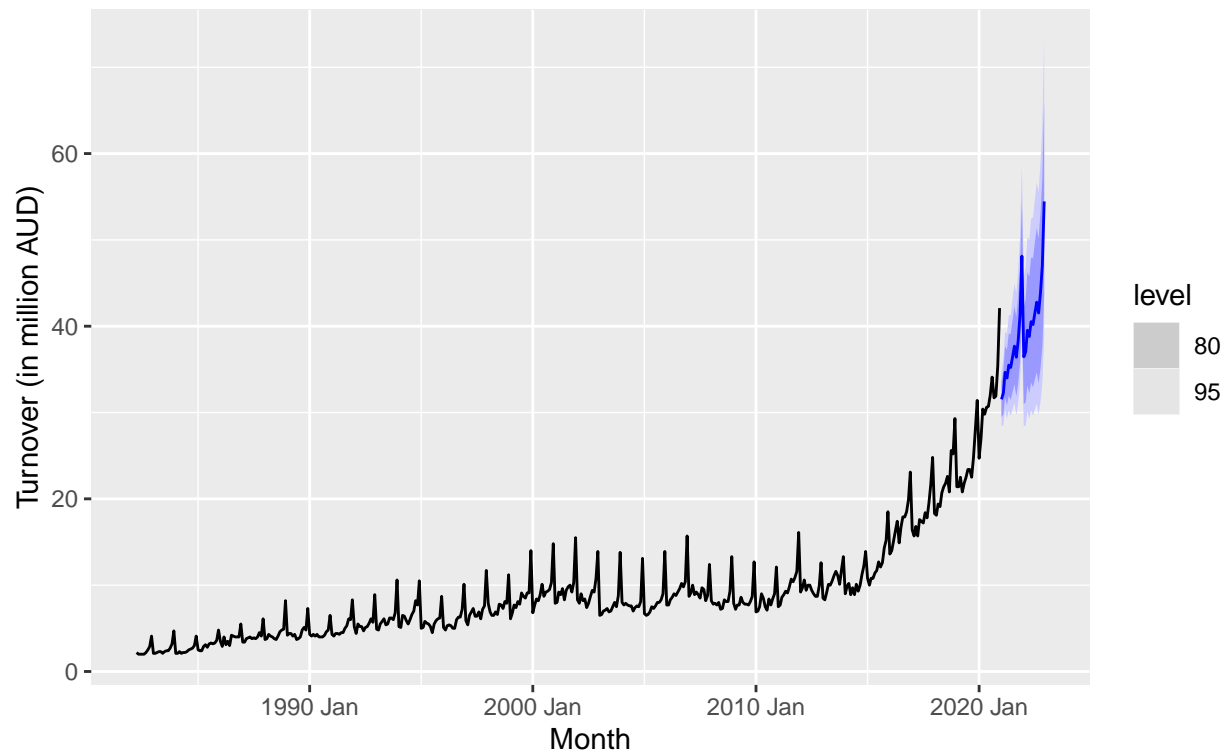
```
my_series_fit <- my_series |>
  model(
    arima_011_111 = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      pdq(p = 0, d = 1, q = 1) +
      PDQ(P = 1, D = 1, Q = 1)
    )
  )

my_series_fc <- my_series_fit |>
  forecast(h = "2 years")

my_series_fc |>
  autoplot(my_series) +
  labs(
    title = "2 Years Forecast Turnover in Australian Capital Territory",
    subtitle = paste0(
      "From ",
      pull(my_series[1, 1]),
      " to ",
      pull(my_series[nrow(my_series), 1]),
      ", forecasted to ",
      pull(my_series_fc[nrow(my_series_fc), 2])
    ),
    y = "Turnover (in million AUD)"
  )
```


2 Years Forecast Turnover in Australian Capital Territory

From 1982 Apr to 2020 Dec, forecasted to 2022 Dec



Observation

The point forecast follows recent observation, it is visually reasonable assuming the future follows the momentum of the trend. The prediction interval is widening for further forecast horizon, but it is relatively narrow to other models. This could be due to the fitness of the model from the residual tests.

Checking Forecast Accuracy

```
# Latest datasets as in 2023-05-09 (YYYY-MM-DD)
# Table 11
data_url <- "https://www.abs.gov.au/statistics/industry/retail-and-wholesale-trade/retail-trade-austral.

download.file(
  url = data_url,
  destfile = here("updated_datasets.csv"),
  mode = "wb"
)

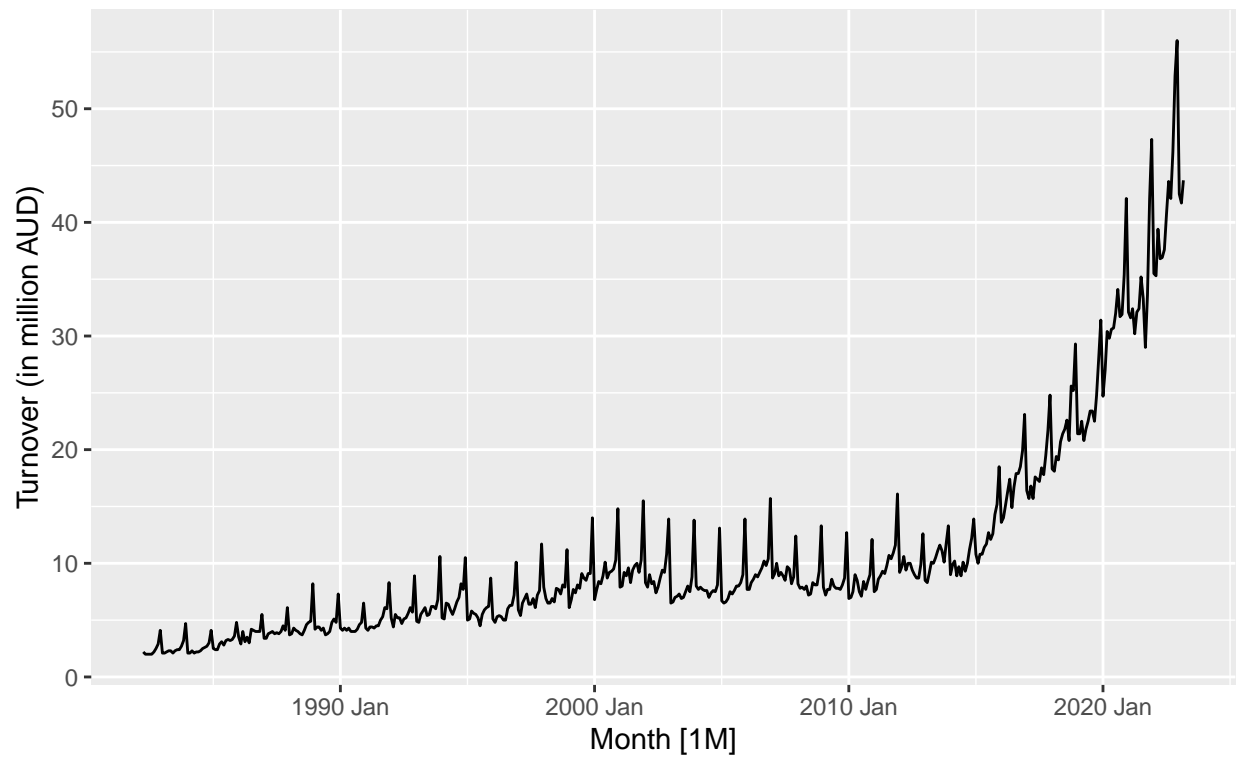
# Interested in Data1 sheets
readxl::excel_sheets(here("updated_datasets.csv"))
```

```
## [1] "Index"      "Data1"      "Enquiries"
```

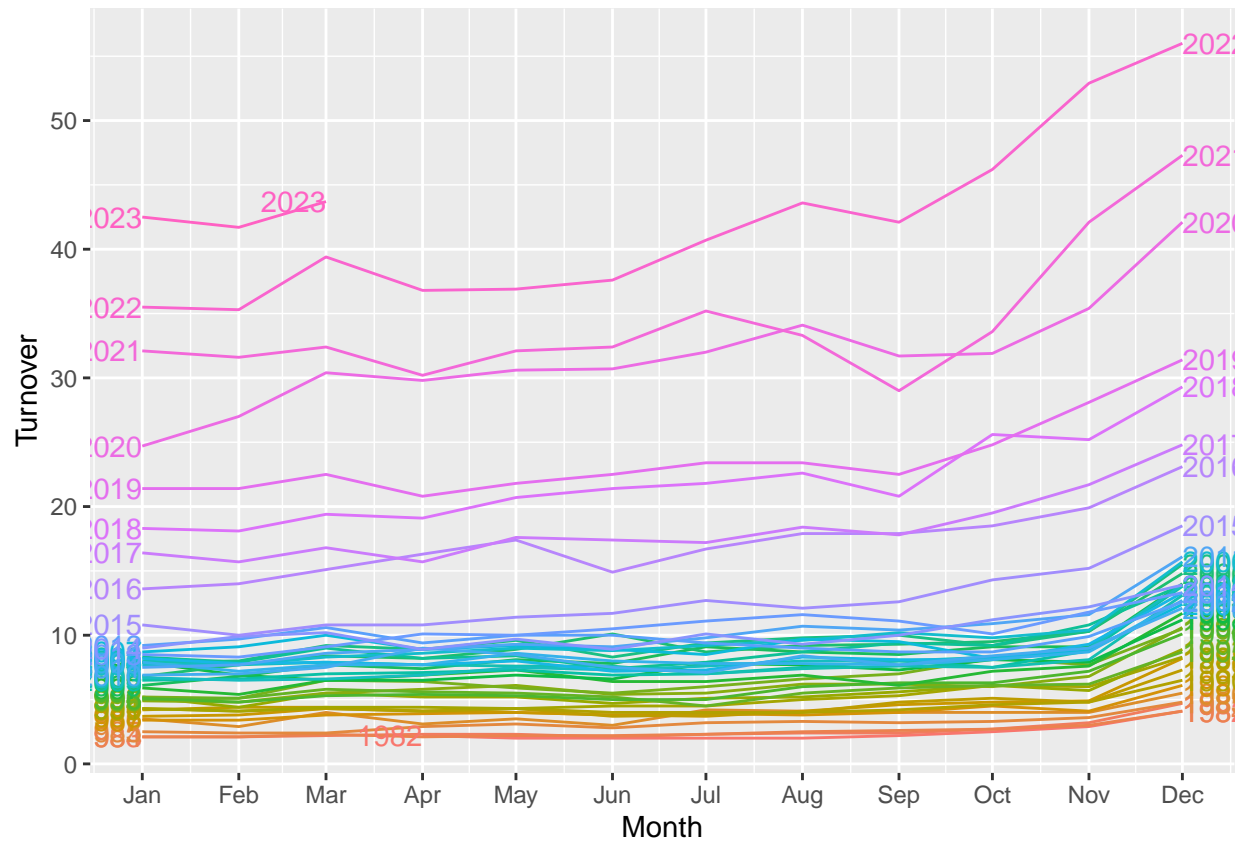
```
my_updated_series <- readxl::read_xlsx(
  here("updated_datasets.csv"),
  sheet = "Data1",
  skip = 9
) |>
  select(
    `Series ID`,
    any_of(my_meta_id)
  ) |>
  rename(
    "Month" = `Series ID`,
    "Turnover" = any_of(my_meta_id)
  ) |>
  mutate(
    Month = yearmonth(Month)
  ) |>
  as_tsibble(
    index = Month
  )

my_updated_series |>
  autoplot(Turnover) +
  labs(
    title = "Turnover of Retail Sector in Australian Capital Territory",
    subtitle = paste0(
      "Other Retailing N.E.C. from ",
      pull(my_updated_series[1, 1]),
      " to ",
      pull(my_updated_series[nrow(my_updated_series), 1])
    ),
    y = "Turnover (in million AUD)"
  )
```

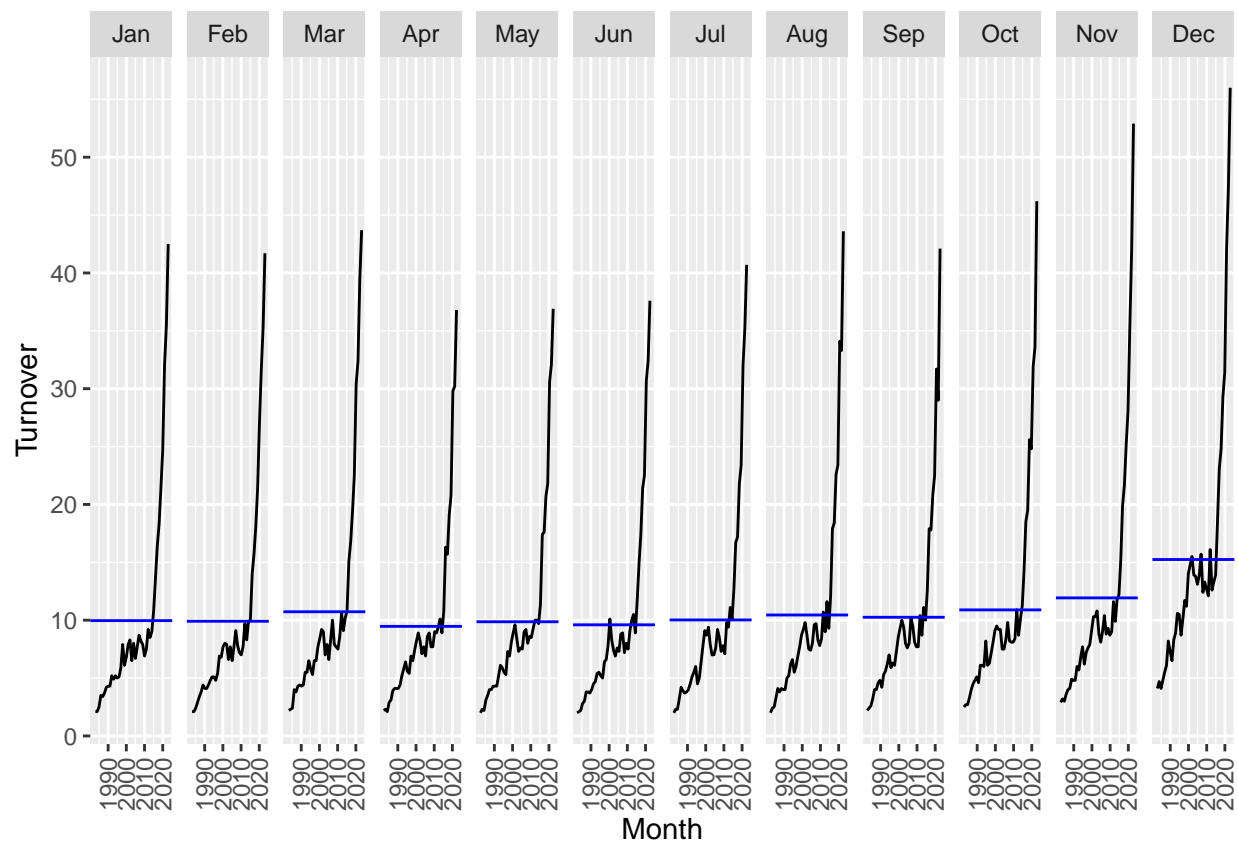
Turnover of Retail Sector in Australian Capital Territory Other Retailing N.E.C. from 1982 Apr to 2023 Mar



```
my_updated_series |>  
  gg_season(  
    y = Turnover,  
    labels = "both"  
  )
```



```
my_updated_series |>
  gg_subseries(y = Turnover) +
  theme(
    axis.text.x = element_text(vjust = 0.5)
  )
```



Observation

The movement of the time series was not expected, the COVID-19 pandemic did not impact the turnover of the “other” retailing sector in the Australian Capital Territory. The momentum of the trend has kept on going despite effect of the pandemic, but it did not grow exponentially, perhaps the width of the graph has skewed the perspective of trend visually as shown by the seasonal and subseries plots.

Another component to consider is the increasing variation of the seasonal components, the event of COVID-19 has apparently increased the variation of seasonal pattern in the time series. This suggests changing fluctuating variation throughout the time series, this is to be expected due to possibly other lurking variables e.g. events not taken into account since the only variable being observed is COVID-19.

Checking Forecast Accuracy with Different Models

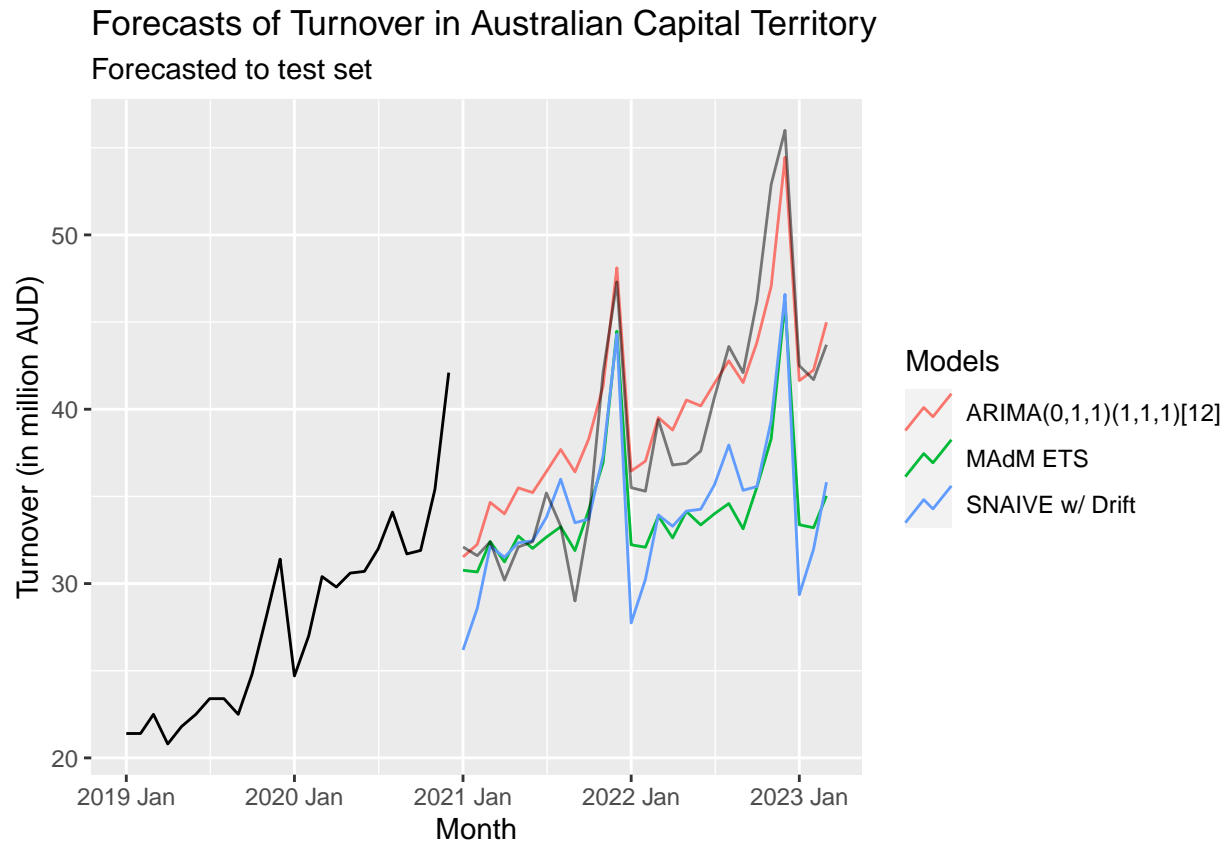
```
updated_train_set <- my_series
updated_test_set <- my_updated_series |>
  filter(
    year(Month) > (max(year(Month)) - 3)
  )

updated_test_fc_range <- nrow(updated_test_set)

updated_train_set_fit <- updated_train_set |>
  model(
    "SNAIVE w/ Drift" = SNAIVE(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
      drift()
    ),
    "MAdM ETS" = ETS(
      Turnover ~
        error("M") +
        trend("Ad") +
        season("M")
    ),
    "ARIMA(0,1,1)(1,1,1)[12]" = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_series_lambda
      ) ~
        pdq(p = 0, d = 1, q = 1) +
        PDQ(P = 1, D = 1, Q = 1)
    )
  )

# Point forecast comparison
updated_train_set_fit |>
  forecast(h = updated_test_fc_range) |>
  autoplot(
    my_series |>
      filter_index("2019 Jan" ~ .),
    level = NULL
  ) +
  autolayer(
    updated_test_set,
    alpha = 0.5
  ) +
  labs(
    title = "Forecasts of Turnover in Australian Capital Territory",
    subtitle = "Forecasted to test set",
    y = "Turnover (in million AUD)",
    colour = "Models"
  )
```

```
## Plot variable not specified, automatically selected '.vars = Turnover'
```

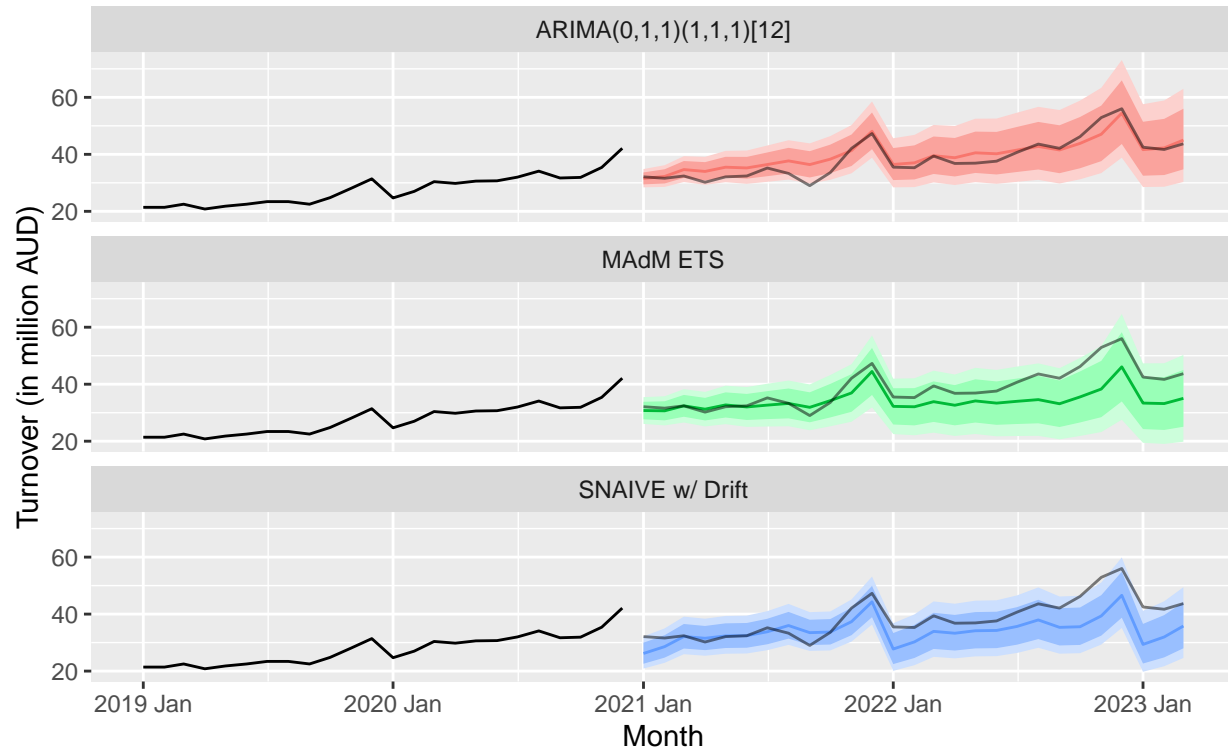


```
# Interval comparison
updated_train_set_fit |>
  forecast(h = updated_test_fc_range) |>
  autoplot(
    my_series |>
      filter_index("2019 Jan" ~ .)
  ) +
  autolayer(
    updated_test_set,
    alpha = 0.5
  ) +
  facet_wrap(
    ~ .model,
    ncol = 1
  ) +
  labs(
    title = "Forecasts of Turnover in Australian Capital Territory",
    subtitle = "Forecasted to test set",
    y = "Turnover (in million AUD)",
    colour = "Models"
  ) +
  theme(
    legend.position = "none"
  )
```

```
## Plot variable not specified, automatically selected '.vars = Turnover'
```

Forecasts of Turnover in Australian Capital Territory

Forecasted to test set



```
# Prediction interval accuracy
updated_train_set_fit |>
  forecast(h = updated_test_fc_range) |>
  accuracy(
    my_updated_series,
    list(
      winkler = winkler_score
    ),
    level = 80
  )
```

```
## # A tibble: 3 x 3
##   .model                .type winkler
##   <chr>                 <chr>   <dbl>
## 1 ARIMA(0,1,1)(1,1,1)[12] Test     14.1
## 2 MADm ETS             Test     16.7
## 3 SNAIVE w/ Drift      Test     20.2
```

Observation

The models' forecasts compared to the observation is similar in performance with the exception of the ARIMA model, from 2020 to around 2022, all models tend to overestimate the forecast but post 2022 the *Holt-Winters' Damped* and SNAIVE with Drift tend to underestimate the forecast whilst the ARIMA overestimates the forecast the bottom half from 2022 to 2023, the top half being so close to observation or

else underestimate the forecast as well. But it is visually observed that ARIMA is more accurate overall compared to the other model, supported by the fitness from the residual analysis.

The 95% prediction interval shows that the ETS model has a wider interval whilst the SNAIVE with Drift model shows a narrower interval, with the ARIMA model being the narrowest. The SNAIVE with Drift model is fascinating as it has narrower interval despite the residuals being auto-correlated meaning trend/seasonal components are leaking to the residuals. The ETS model has a wide interval from the residuals being auto-correlated with the addition of a multiplicative error. Using the *Winkler Score*, using the 80% prediction interval as the expectation is that the observations would land within the 80% interval, 80% of the time and outside the 95%, 5% of the time. With the *Winkler Score* measuring 80% interval would give greater penalty for further interval misses relative to the actual observation. It is observed that the ARIMA model provides a greater accuracy for the 80% prediction interval.

Point Forecast Accuracy Measures

```
updated_train_set_fit |>
  forecast(h = updated_test_fc_range) |>
  accuracy(my_updated_series) |>
  arrange(RMSSE)
```

```
## # A tibble: 3 x 10
##   .model                .type    ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>                <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ARIMA(0,1,1)(1,1,1)[12] Test  -1.18  2.81  2.17 -3.97  6.03  1.74  1.40  0.645
## 2 MAdM ETS             Test   4.35  6.12  4.73  9.91 11.2   3.80  3.05  0.841
## 3 SNAIVE w/ Drift      Test   4.41  6.30  5.07 10.2  12.4   4.07  3.14  0.807
```

Observation

From the point forecast accuracy, the previous analysis is supported by this table, the ARIMA model is the most accurate compared to the other models in all metrics being minimised in value such as *RMSE*, *MASE* and *RMSSE* whilst the *ME* and *MPE* are closer to zero.

Forecasting with Best Chosen Models

```
my_updated_series_lambda <- my_updated_series |>
  features(
    Turnover,
    guerrero
  ) |>
  pull()

my_updated_series_fit <- my_updated_series |>
  model(
    "SNAIVE w/ Drift" = SNAIVE(
      box_cox(
        x = Turnover,
        lambda = my_updated_series_lambda
      ) ~
      drift()
    ),
    "MAdM ETS" = ETS(
      Turnover ~
      error("M") +
      trend("Ad") +
      season("M")
    ),
    "ARIMA(0,1,1)(1,1,1)[12]" = ARIMA(
      box_cox(
        x = Turnover,
        lambda = my_updated_series_lambda
      ) ~
      pdq(p = 0, d = 1, q = 1) +
      PDQ(P = 1, D = 1, Q = 1)
    )
  )

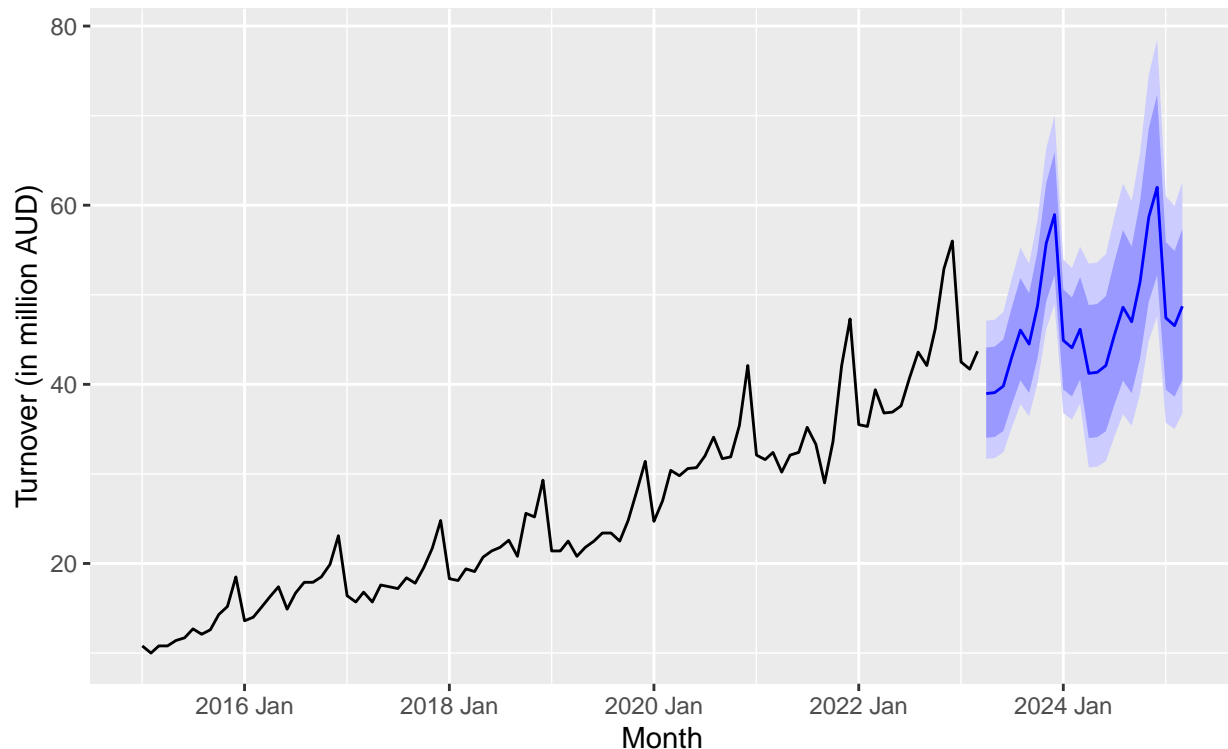
my_updated_series_fc <- my_updated_series_fit |>
  forecast(h = "24 months")
```

```
# SNAIVE w/ Drift
my_updated_series_fc |>
  filter(.model == "SNAIVE w/ Drift") |>
  autoplot(
    my_updated_series |> filter(year(Month) >= 2015)
  ) +
  labs(
    title = "24 Months Forecasts of Turnover in Australian Capital Territory",
    subtitle = paste0(
      "SNAIVE, from ",
      pull(my_updated_series_fc[1, 2]),
      " to ",
      pull(my_updated_series_fc[nrow(my_updated_series_fc), 2])
    ),
    y = "Turnover (in million AUD)"
  ) +
```

```
theme(
  legend.position = "none"
)
```

24 Months Forecasts of Turnover in Australian Capital Territory

SNAIVE, from 2023 Apr to 2025 Mar



Observation

SNAIVE with Drift fares mediocrely, it is based on the last observed value with the drift setting the trend. Another issue with the SNAIVE method is that since it only weights the most recent observation, it does not weight the previous observations like the ETS nor it uses stationary data thus using previous value/error like ARIMA. With the trend, the drift is similar to setting the trend from the first observation to the last observation in a straight line. It does not take into account the change in trend nor the trend in recent observation.

```
# MAdM ETS
my_updated_series_fc |>
  filter(.model == "MAdM ETS") |>
  autoplot(
    my_updated_series |> filter(year(Month) >= 2015)
  ) +
  labs(
    title = "24 Months Forecasts of Turnover in Australian Capital Territory",
    subtitle = paste0(
      "MAdM ETS, from ",
      pull(my_updated_series_fc[1, 2]),
      " to ",
      pull(my_updated_series_fc[nrow(my_updated_series_fc), 2])
    ),
  ),
```

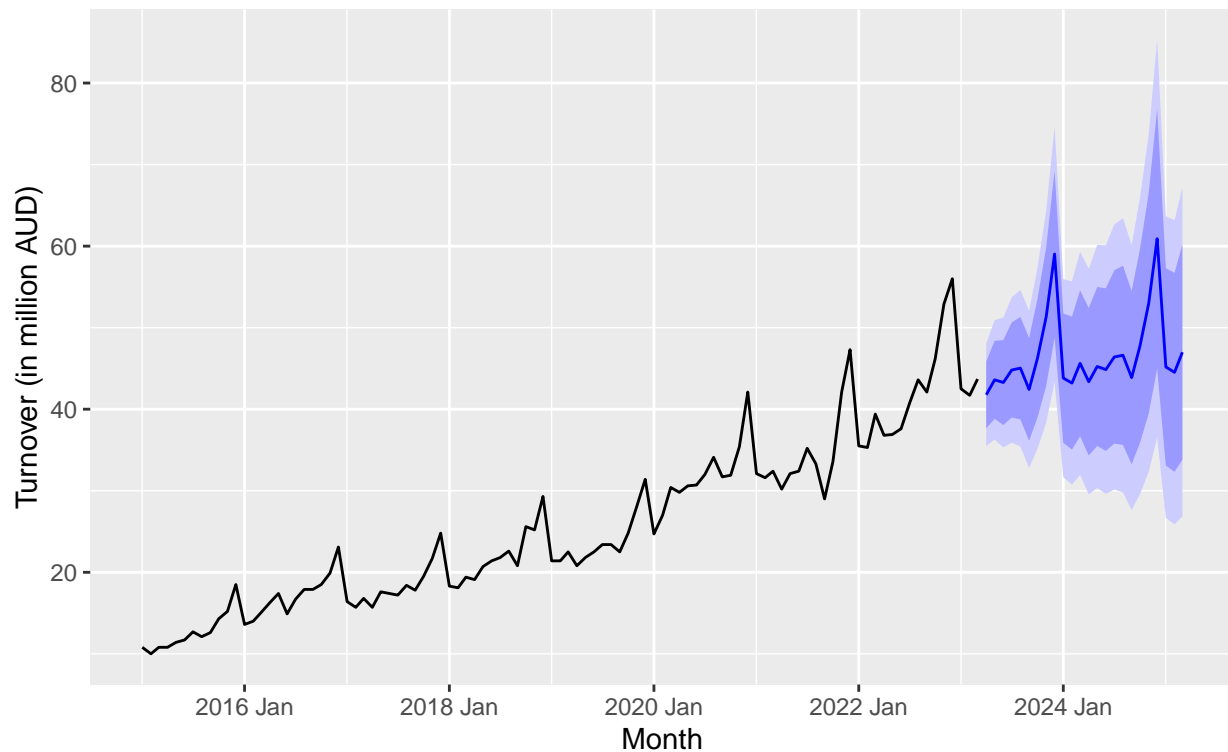
```

y = "Turnover (in million AUD)"
) +
theme(
  legend.position = "none"
)

```

24 Months Forecasts of Turnover in Australian Capital Territory

MAdM ETS, from 2023 Apr to 2025 Mar



Observation

Holt-Winters' Damped (MAdM) ETS fares okay, however, it is personally argued that it could be worse than SNAIVE with Drift in longer horizon. It is due to the ϕ component that is the dampening component, it will slowly dampen the trend and eventually flattens the trend the further the forecast horizon, the early horizon may be adequate, but the errors will add up the further the forecast horizon is and will significantly underestimate the point forecast. The wide 95% prediction interval is from the combination of auto-correlated residuals and multiplicative errors.

```

# ARIMA(0,1,1)(1,1,1)[12]
my_updated_series_fc |>
  filter(.model == "ARIMA(0,1,1)(1,1,1)[12]") |>
  autoplot(
    my_updated_series |> filter(year(Month) >= 2015)
  ) +
  labs(
    title = "24 Months Forecasts of Turnover in Australian Capital Territory",
    subtitle = paste0(
      "ARIMA(0,1,1)(1,1,1)[12], from ",
      pull(my_updated_series_fc[1, 2]),
      " to ",

```

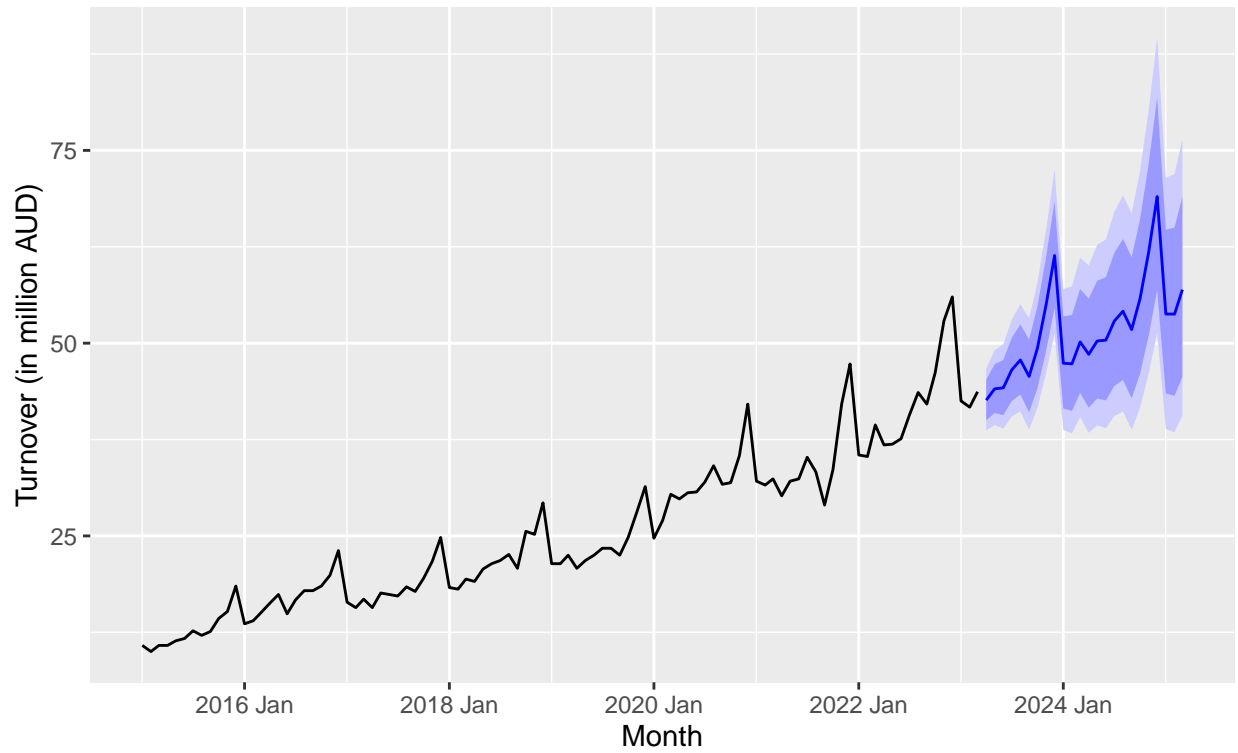
```

pull(my_updated_series_fc[nrow(my_updated_series_fc), 2])
),
y = "Turnover (in million AUD)"
) +
theme(
  legend.position = "none"
)

```

24 Months Forecasts of Turnover in Australian Capital Territory

ARIMA(0,1,1)(1,1,1)[12], from 2023 Apr to 2025 Mar



Observation

The ARIMA(0,1,1)(1,1,1)[12] model seems to be fairing well with the addition of COVID-19 effect, the point forecast seems reasonable following the increasing trend from COVID-19 and the prediction interval seems narrow, but it does get wider as the forecast horizon gets further from uncertainty in further forecast horizon. The seasonal pattern is visually similar to the recent observations, the trend however is slowly getting steeper the further the forecast horizon which is observed at past observations. Which makes the ARIMA model possibly the appropriate model to forecast the turnover datasets for the time being that is March 2023.