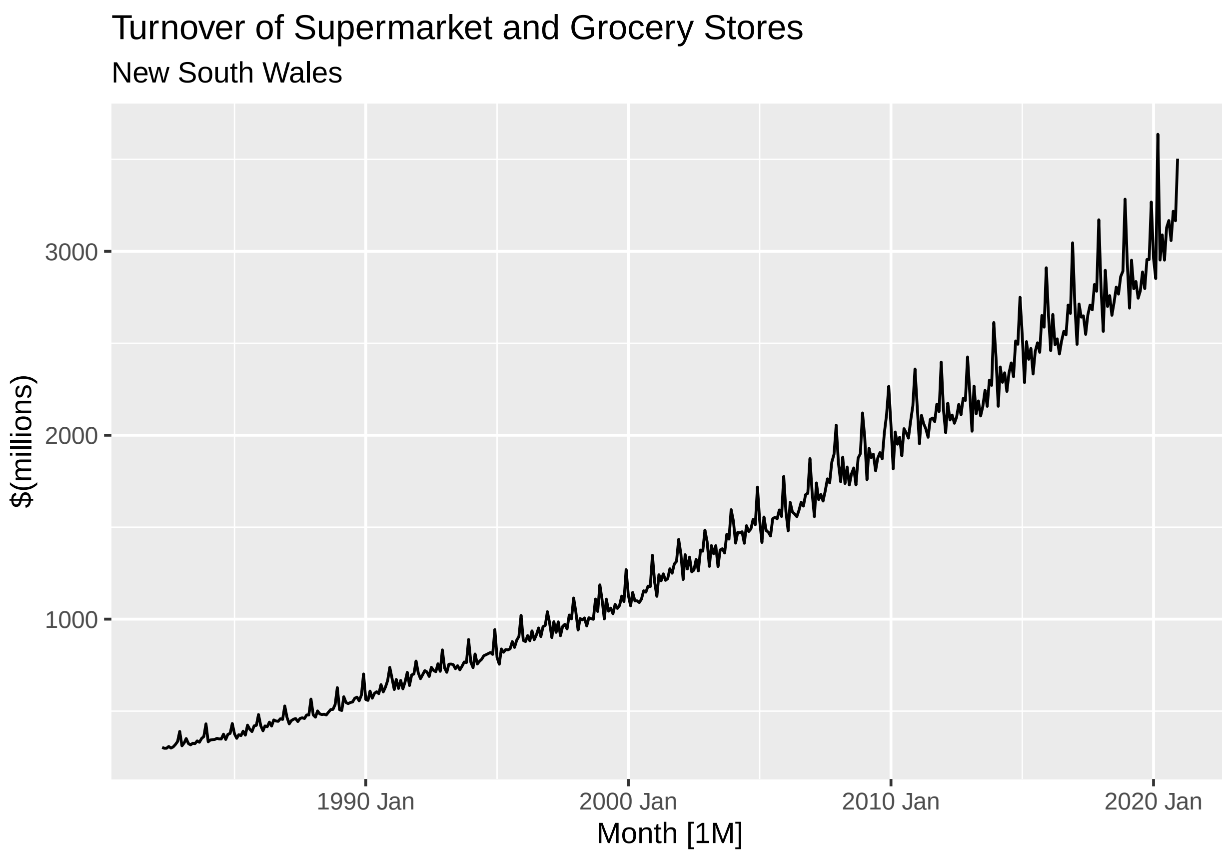
Forecasting Turnover of Markets in New South Wales

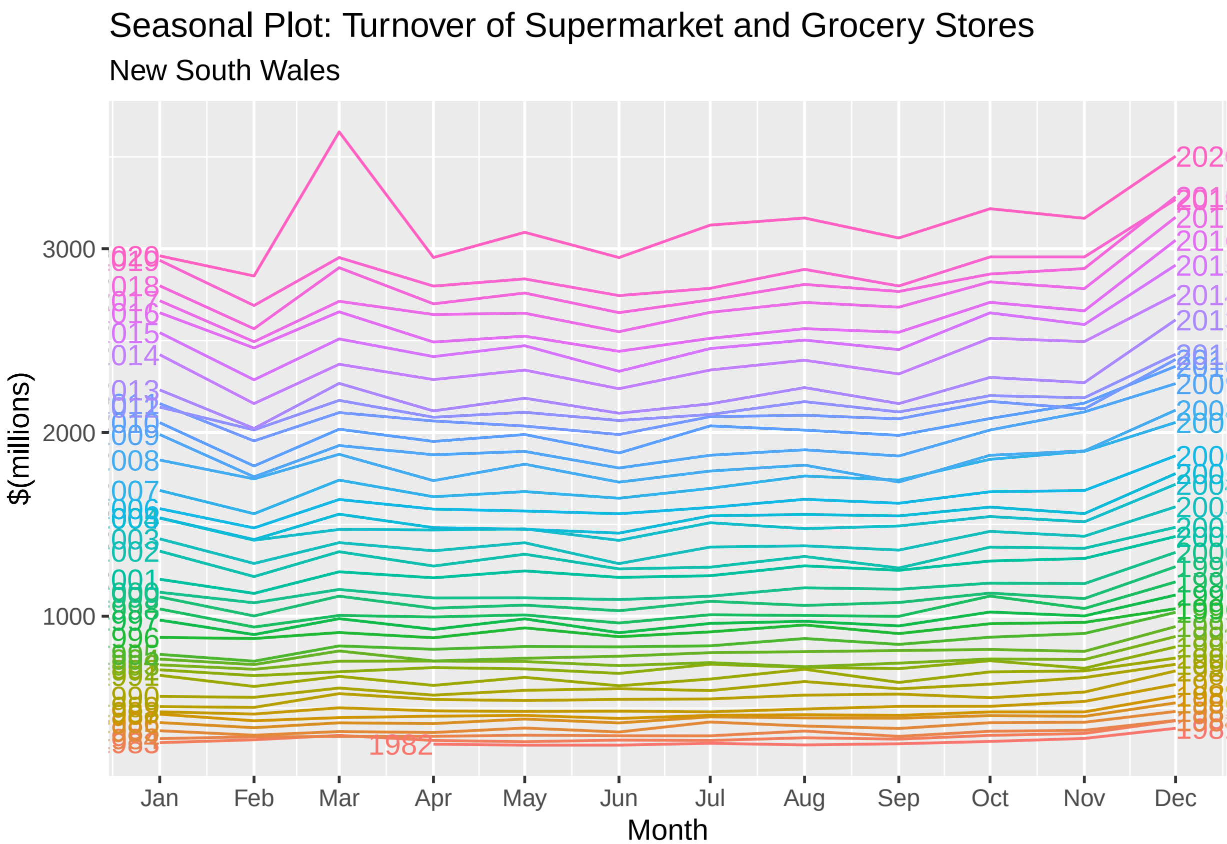
Changsoo Byun

```r
#Time plot of Turnover of Supermarket and Grocery Stores in New South Wales

timeplot <- my_series |>
  autoplot(y)+
  labs(title="Turnover of Supermarket and Grocery Stores",
       subtitle="New South Wales",
       y="$(millions)")

print(timeplot)
```



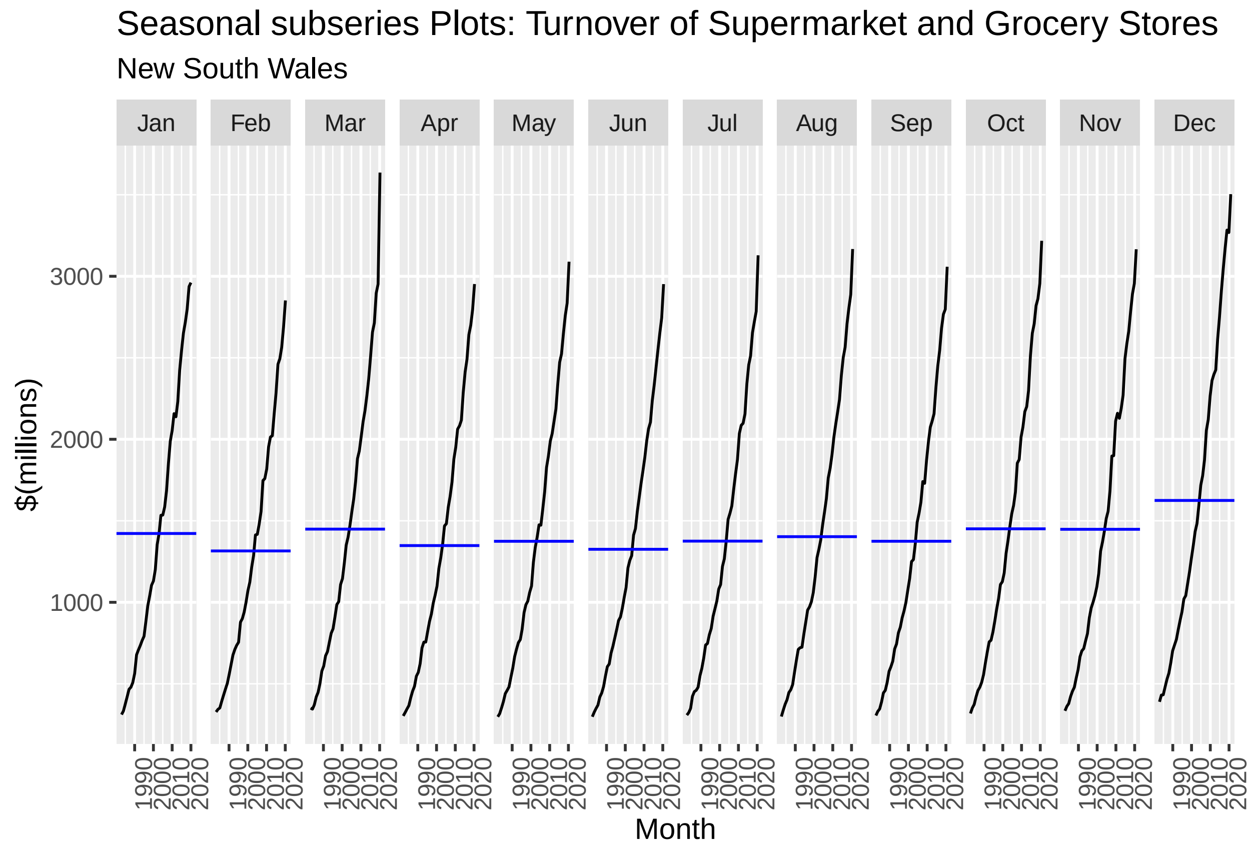
#The time plot depicts an increasing trend in the turnover data of supermarket and grocery stores in New South Wales. Moreover, there is a clear presence of seasonality in the data as the turnover is consistently higher during November and December. The pattern of rises and falls in the data occurs at fixed periods, indicating the absence of any significant cyclicity.

```
#Seasonal Plot of Turn of Supermarket and Grocery Stores in New South Wales

seasonal <- my_series |>
  gg_season(y, labels="both")+
  labs(title="Seasonal Plot: Turnover of Supermarket and Grocery Stores",
       subtitle="New South Wales",
       y="$(millions)")

seasonal
```



#The turnover of supermarket and grocery stores in New South Wales shows a sharp increase in March and December, while February shows a slight dip. Seasonal factors such as the Christmas or other holidays drive high turnover during March and December.

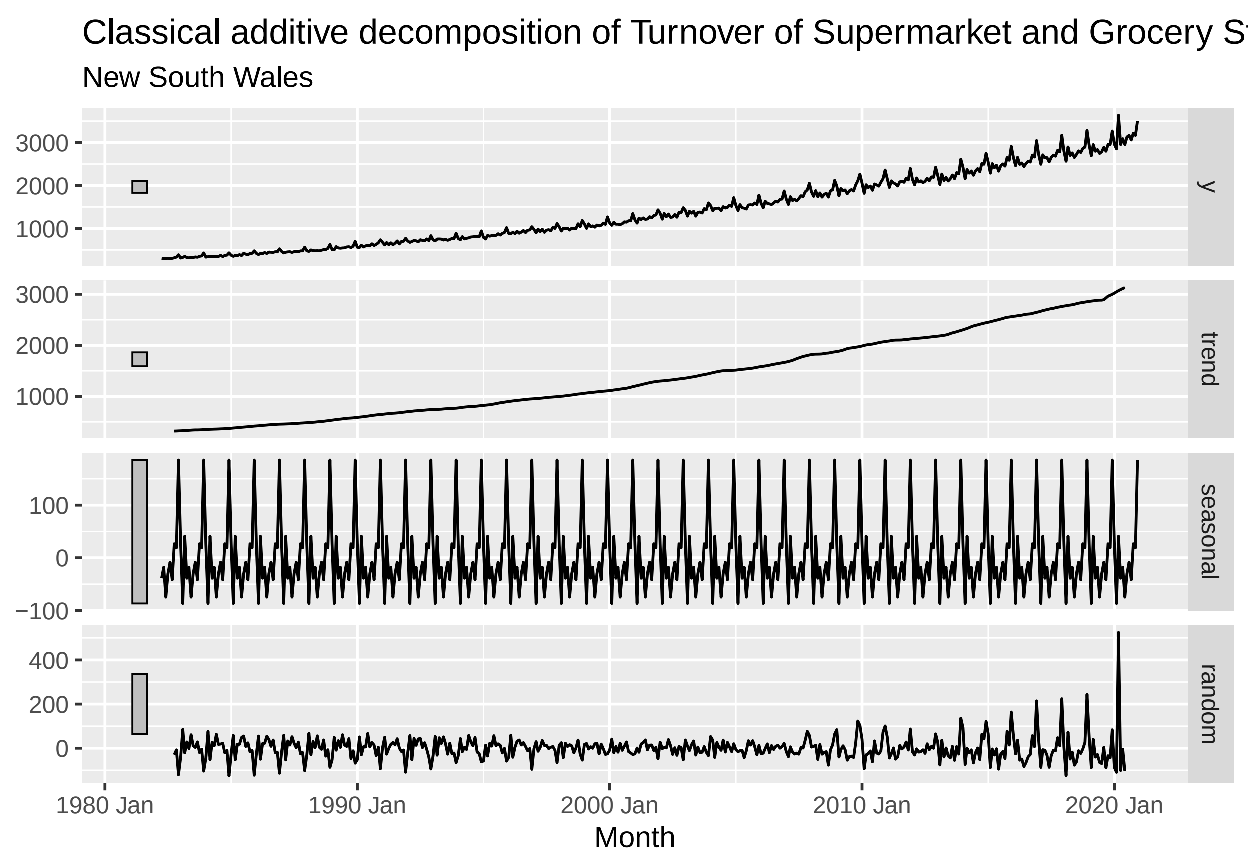
```
#Seasonal subseries plots

ssp <- my_series |>
  gg_subseries(y)+
  labs(
    y="$(millions)",
    title="Seasonal subseries Plots: Turnover of Supermarket and Grocery Stores",
    subtitle="New South Wales"
  )

ssp
```



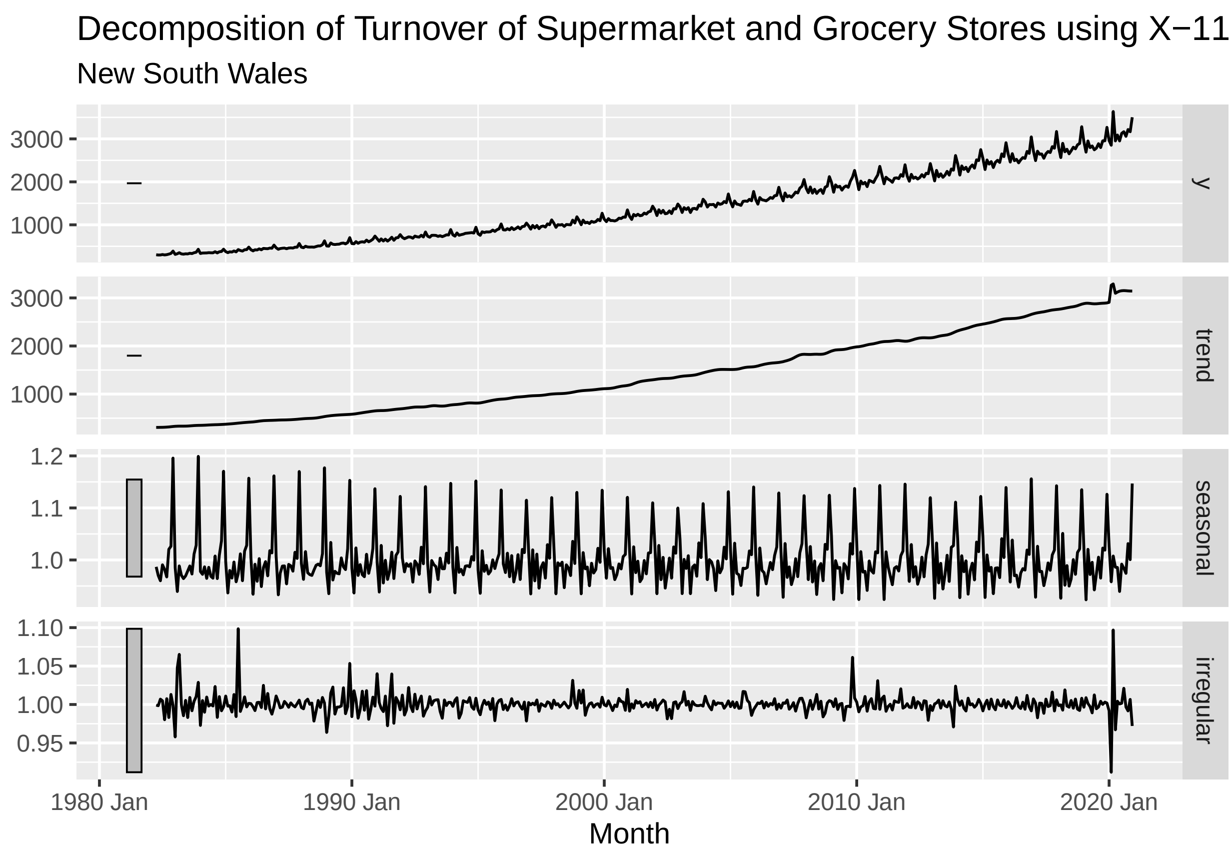
#The plot displays some seasonal patterns, with a notable rise in turnover during December and a decline in February, while other months remain consistent with few fluctuations.

```
#Additive decomposition

AD <- my_series |>
  model(
    classical_decomposition(y, type = "additive")
  ) |>
  components() |>
  autoplot() +
  labs(title = "Classical additive decomposition of Turnover of Supermarket and Grocery Stores",
       subtitle="New South Wales"
       )

AD
```

```
## Warning: Removed 6 rows containing missing values (`geom_line()`).
```




```
#X-11 method

x11_dcmp <- my_series |>
  model(x11 = X_13ARIMA_SEATS(y ~ x11())) |>
  components()
autoplot(x11_dcmp) +
  labs(title =
    "Decomposition of Turnover of Supermarket and Grocery Stores using X-11.",
    subtitle="New South Wales")
```

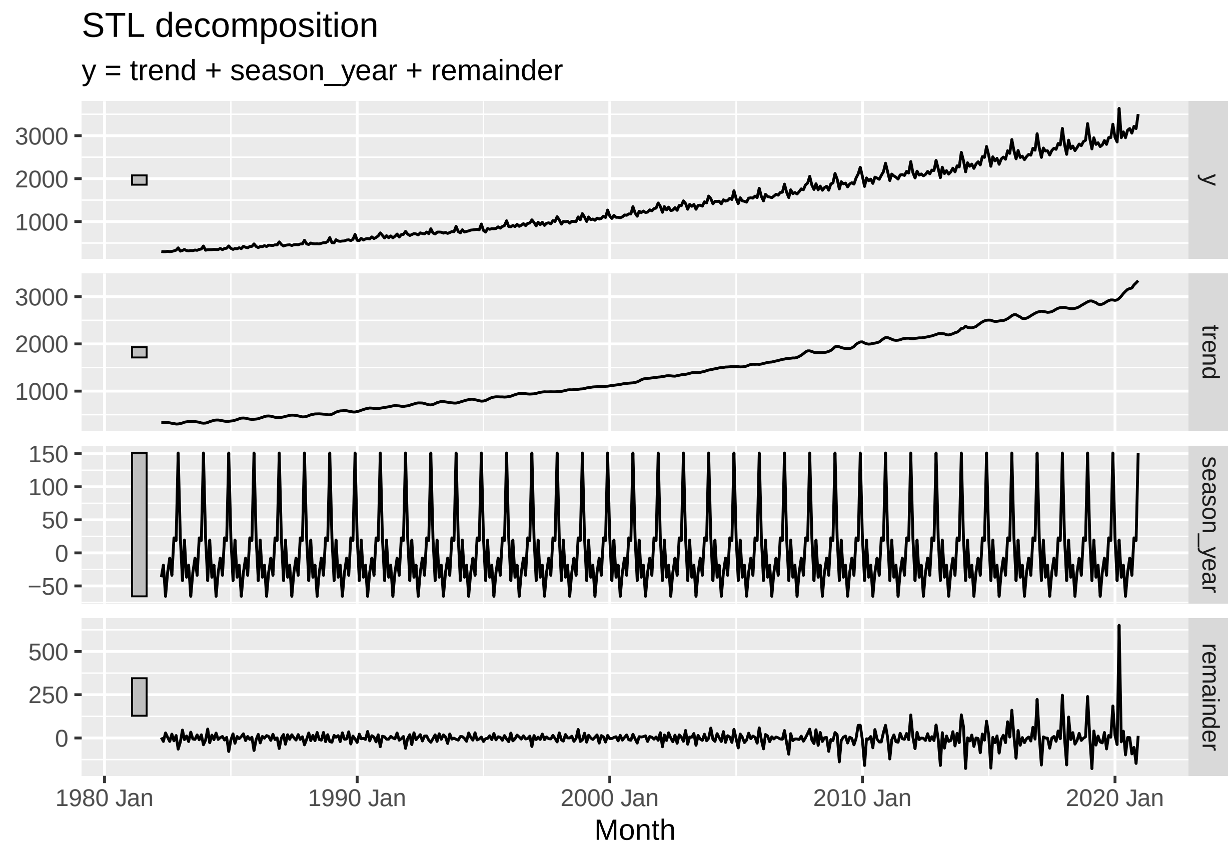



```
x11_dcmp
```

```
## # A dable: 465 x 7 [1M]
## # Key:      .model [1]
## # :         y = trend * seasonal * irregular
##    .model    Month     y trend seasonal irregular season_adjust
##    <chr>     <mth> <dbl> <dbl>    <dbl>     <dbl>         <dbl>
##  1 x11    1982 Apr  303.  307.    0.987     0.999          307.
##  2 x11    1982 May  298.  307.    0.970     0.998          307.
##  3 x11    1982 Jun  298   308.    0.960     1.01           310.
##  4 x11    1982 Jul  308.  309.    0.991     1.00           311.
##  5 x11    1982 Aug  299.  311.    0.982     0.980          305.
##  6 x11    1982 Sep  305.  313.    0.967     1.01           316.
##  7 x11    1982 Oct  318   317.    1.02      0.983          312.
##  8 x11    1982 Nov  334.  322.    1.03      1.01           326.
##  9 x11    1982 Dec  390.  326.    1.20      0.999          326.
## 10 x11    1983 Jan  311.  330.    0.985     0.958          316.
## # i 455 more rows
```

```
#STL decomposition

STL <- my_series |>
   model(
    STL(y ~ trend(window = 7) +
                   season(window = "periodic"),
```

```
    robust = TRUE)) |>
  components() |>
  autoplot()

STL
```



#Transformations may not be necessary when analyzing time series data, and simple transformations are often sufficient. The use of transformations can significantly affect the PI. If the data contains zeros or negative values, a lamda value greater than zero or the log1p() can be used. Using log transformations is a straightforward way to ensure that forecasts remain positive. It is important to reverse any transformations applied to the data to obtain forecasts in the original scale.

#STL is a more suitable method as it offers advantages over classical decomposition and X-11. It has the ability to handle any type of seasonality, and the seasonal component can change over time with a user-defined rate of change. Users can also control the smoothness of the trend-cycle. Additionally, it doesn't involve trading day or calendar adjustments and only utilizes additive methods

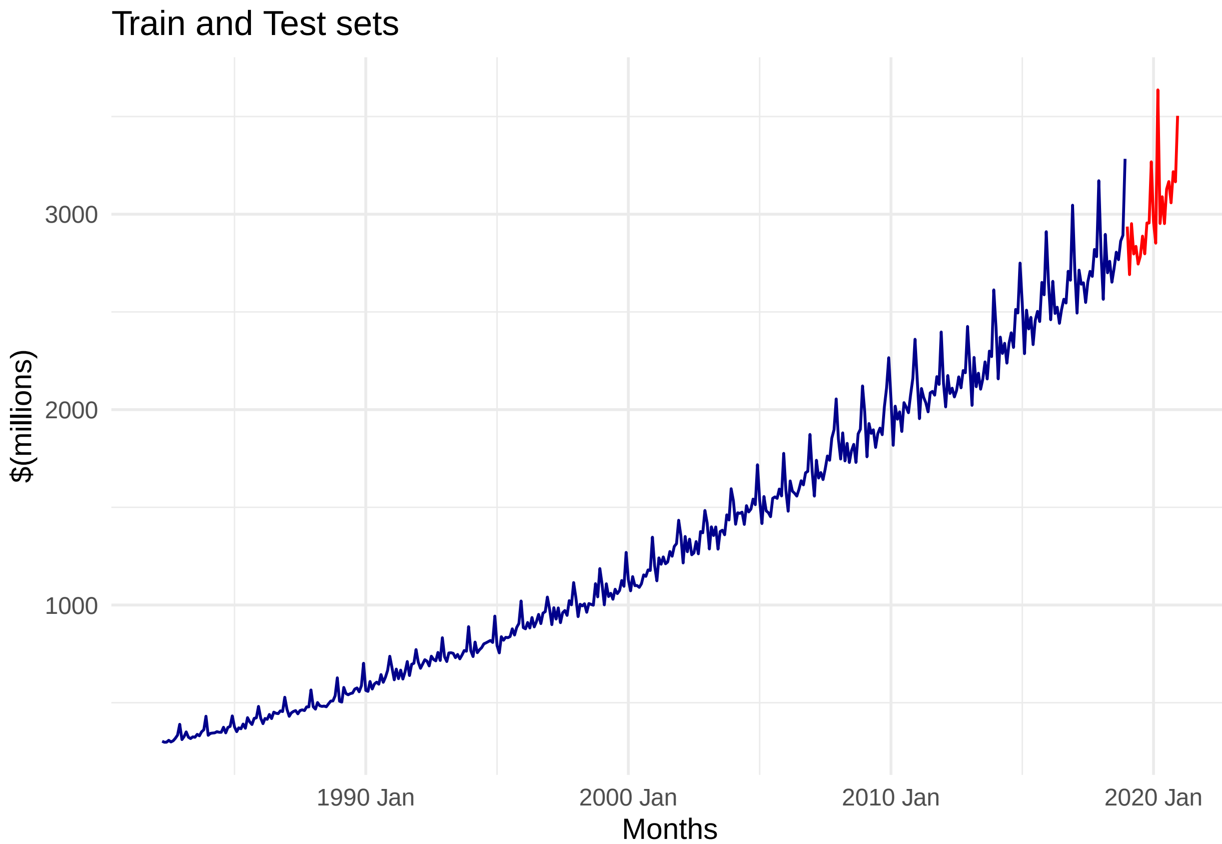
```r
#Training sets

train <- my_series |>
  slice(1:441)

train
```

```
## # A tsibble: 441 x 2 [1M]
##       Month     y
##       <mth> <dbl>
##  1 1982 Apr  303.
##  2 1982 May  298.
##  3 1982 Jun  298
##  4 1982 Jul  308.
##  5 1982 Aug  299.
##  6 1982 Sep  305.
##  7 1982 Oct  318
##  8 1982 Nov  334.
##  9 1982 Dec  390.
## 10 1983 Jan  311.
## # i 431 more rows
```

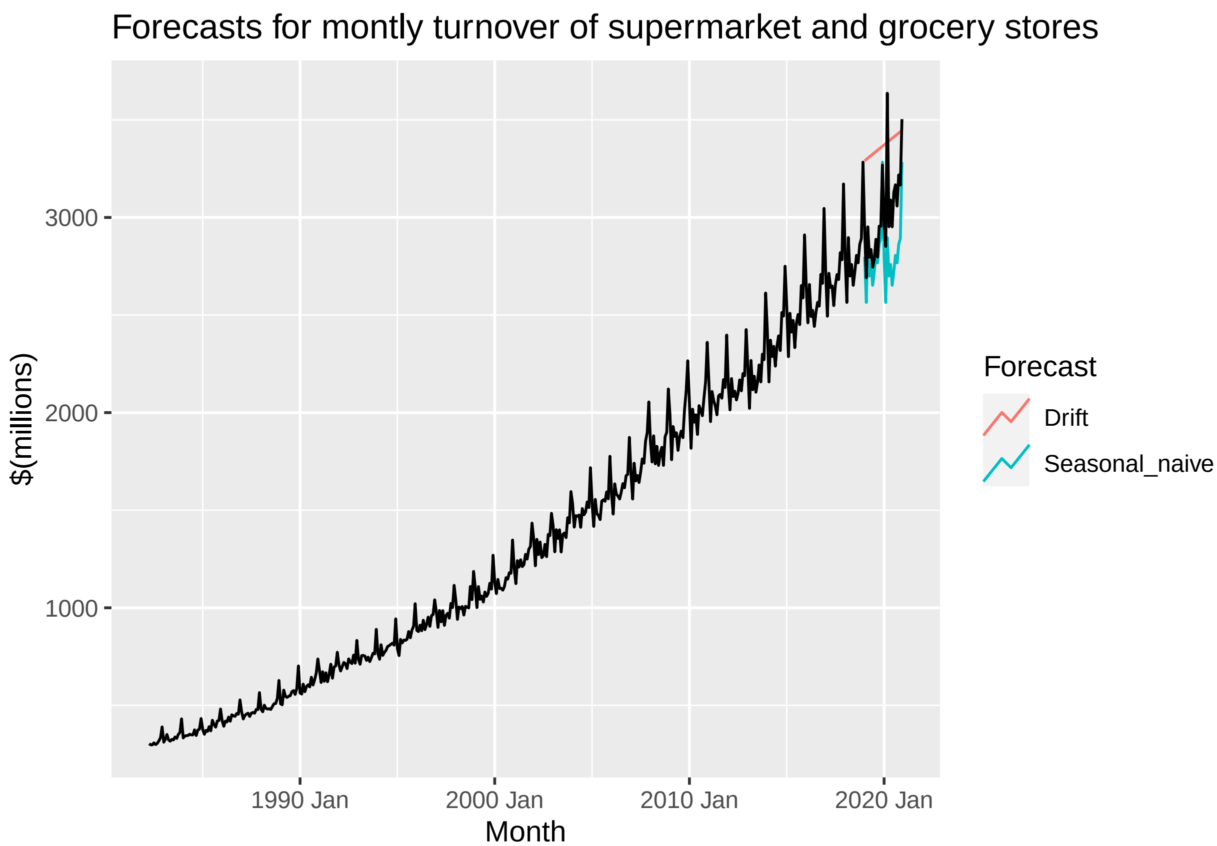
```r
#Test Sets

test <- my_series |>
  slice(442:nrow(my_series))

test
```

```
## # A tsibble: 24 x 2 [1M]
##       Month     y
##       <mth> <dbl>
##  1 2019 Jan 2937.
##  2 2019 Feb 2691.
##  3 2019 Mar 2952.
##  4 2019 Apr 2796.
##  5 2019 May 2836.
##  6 2019 Jun 2745.
##  7 2019 Jul 2784.
##  8 2019 Aug 2888
##  9 2019 Sep 2797.
## 10 2019 Oct 2955.
## # i 14 more rows
```

```r
#Plot to check

q3p <- ggplot() +
  geom_line(data = train, aes(x = Month, y = y), color = "darkblue") +
  geom_line(data = test, aes(x = Month, y = y), color = "red") +
  labs(x = "Months", y = "$(millions)", title = "Train and Test sets") +
  theme_minimal()

q3p
```


Train and Test sets
$(millions)
3000
2000
1000
1990 Jan
2000 Jan
2010 Jan
2020 Jan
Months

```r
fit <- train |>
  model(
     Seasonal_naive = SNAIVE(y),
     Drift = RW(y ~ drift())
  )

fc <- fit |>
  forecast(h= "2 years")

fc |>
  autoplot(
    my_series, level = NULL
  )+
  labs(y= "$(millions)",
       title= "Forecasts for montly turnover of supermarket and grocery stores")+
  guides(colour = guide_legend(title = "Forecast"))
```



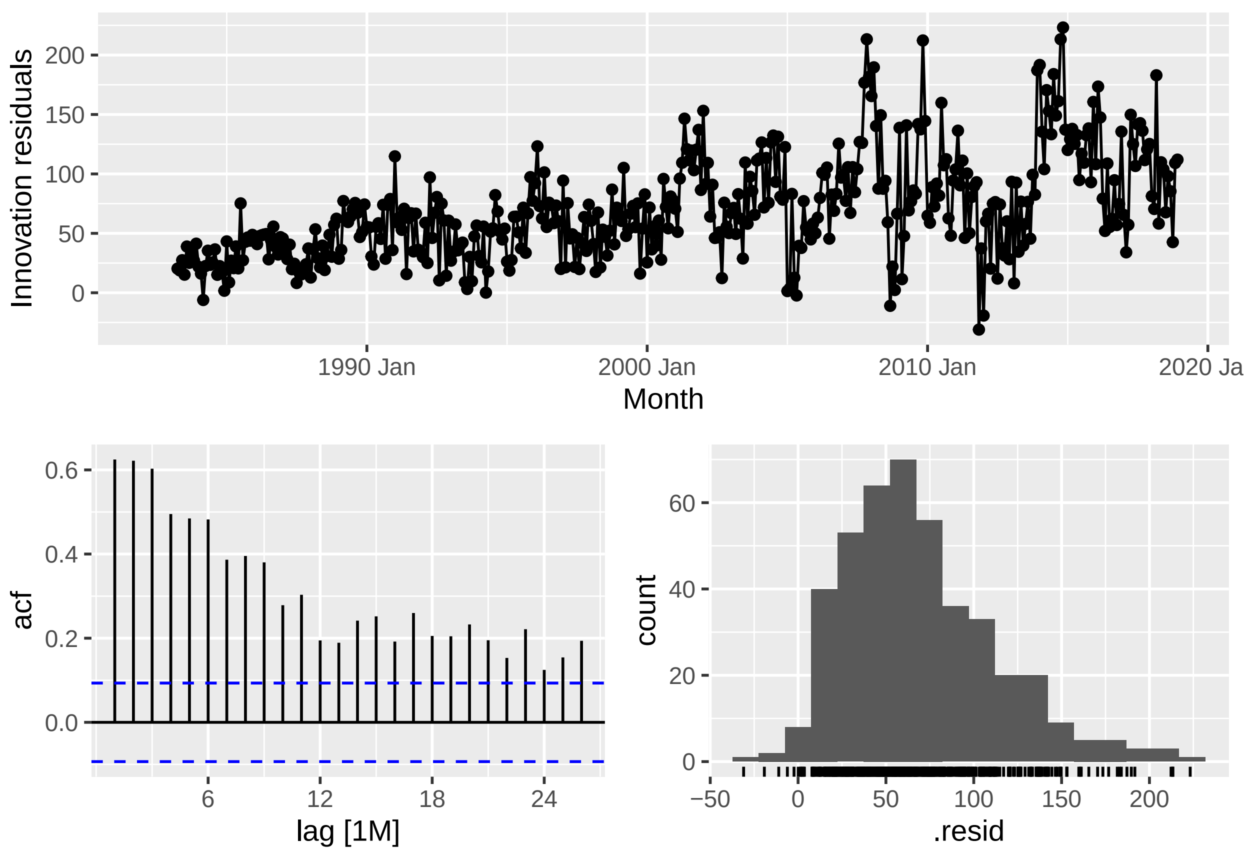

```r
accuracy(fc, my_series)
```

```
## # A tibble: 2 x 10
##   .model         .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE     ACF1
##   <chr>          <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 Drift          Test  -356.  407.  382. -12.3  13.0   5.46  4.97 -0.00309
## 2 Seasonal_naive Test   203.  261.  205.   6.54  6.58  2.93  3.19  0.568
```

#By looking at the error measures provided in the tibble table, it appears that the seasonal naive method is the most appropriate benchmark method for forecasting Turnover of Supermarket and Grocery Stores in New South Wales. The fact that the seasonal naive method outperforms the Drift method in terms of RMSE, MAE, MAPE, MASE, and RMSSE, indicating that it has a better overall accuracy. Moreover, the forecast plot seems that the seasonal naive method produces similar results to the actual turnover recorded.

```
fit |>
  select(Seasonal_naive) |>
  gg_tsresiduals()
```

```
## Warning: Removed 12 rows containing missing values (‘geom_line()‘).

## Warning: Removed 12 rows containing missing values (‘geom_point()‘).

## Warning: Removed 12 rows containing non-finite values (‘stat_bin()‘).
```

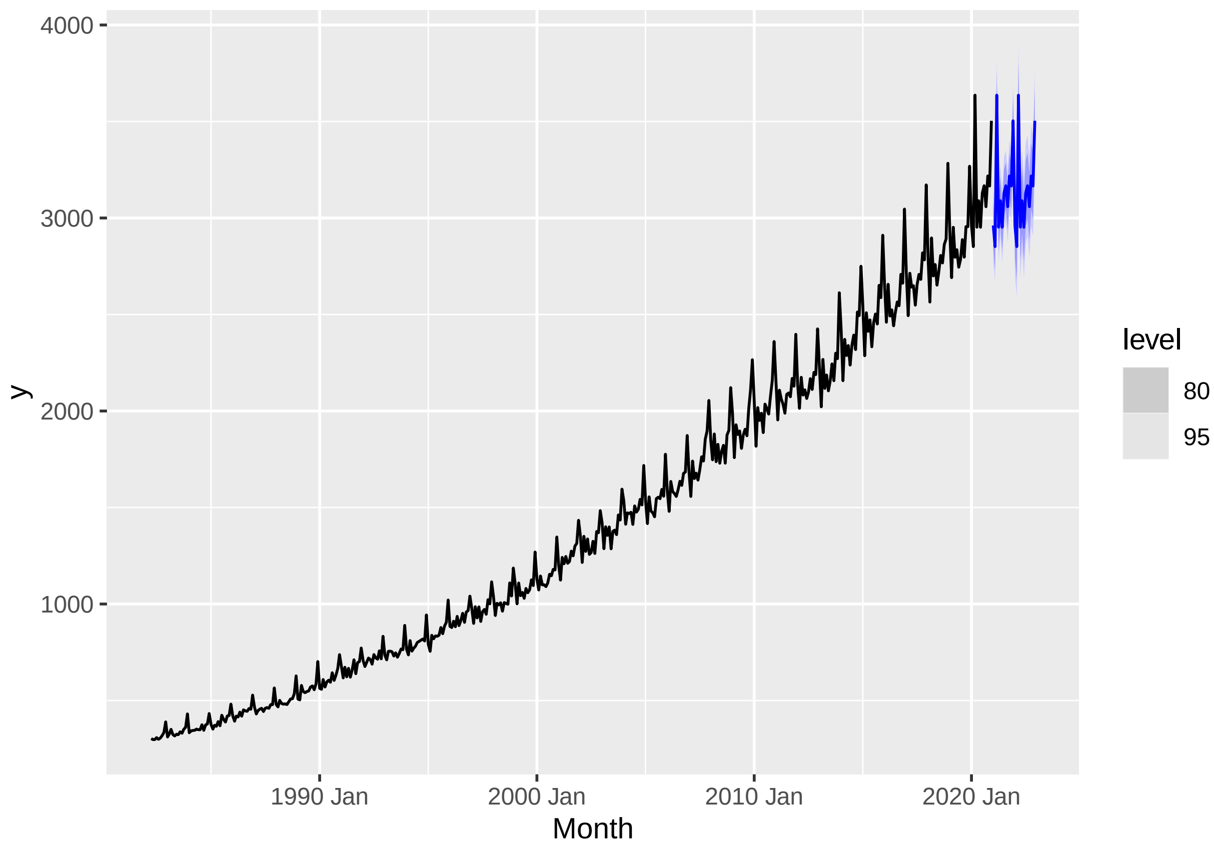



```
augment(fit) |>
  filter(.model=="Seasonal_naive") |>
  features(.innov, ljung_box, lag=24)
```

```
## # A tibble: 1 x 3
##   .model         lb_stat lb_pvalue
##   <chr>            <dbl>     <dbl>
## 1 Seasonal_naive   1331.         0
```

#Based on the diagnostic plots, it appears that the selected model for the time series is inadequate. The autocorrelation plot reveals significant spikes at every lag, indicating that the model is not capturing all the relevant information in the data. Furthermore, the Ljung-Box test suggests strong evidence of residual autocorrelation in the model, which supports this conclusion. Additionally, the histogram of the residuals is slightly right-skewed and not centered around zero, indicating that the forecasts from the model may be biased. Taken together, these findings suggest that the chosen model may not provide accurate predictions. Therefore, it would be advisable to explore alternative models to obtain more reliable forecasting results.

```
my_series |>
  model(SNAIVE(y)) |>
  forecast(h="2 years") |>
  autoplot(my_series)
```



#The plot shows the 80% and 95% prediction intervals and the point forecasts for the turnover of supermarket and grocery stores in New South Wales based on seasonal naive method.The point forecasts look reasonable, but the intervals are wide.This is due to the method relying solely on historical data and not capturing external variables,