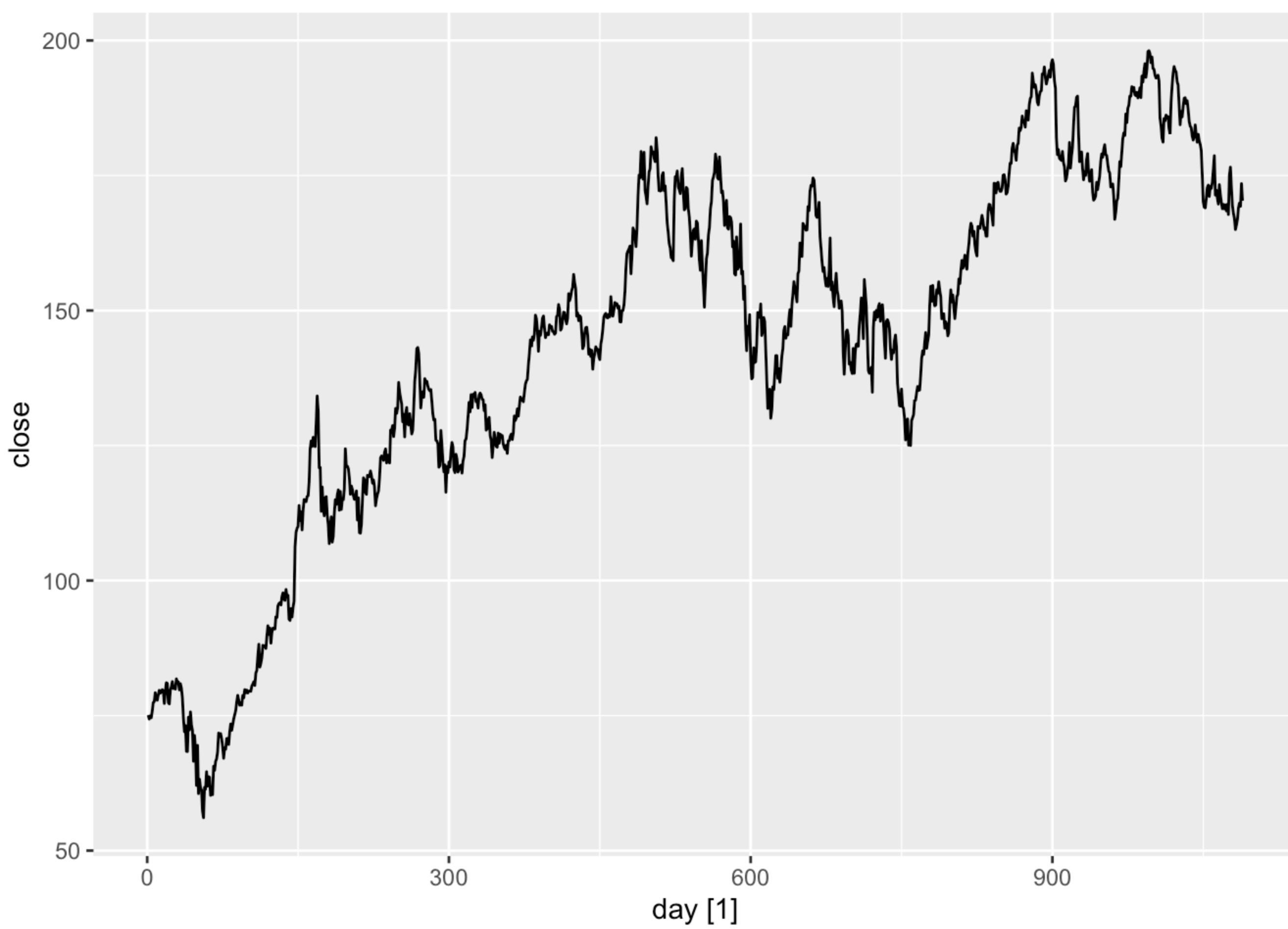
Stock Price Prediction

Shailesh Mahto(50540379), Abhiroop(______), Shivam(______), Kshitij(______)

2024-05-10

```
apple <- tq_get('AAPL', from = '2020-01-01', to = '2024-05-01')
apple <- apple|>
    select(date, close) |>
    mutate(day = seq.int(nrow(apple))) |>
    select(day, close) |>
    as_tsibble(
      index = day
    )
autoplot(apple)
```

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




```
# using all data except 1 year for training
train_apple <- apple |> filter(day <= 989)
```

ETS Model

1. Used Box-cox transformation to stabilize variance
2. Created ETS model and reported autoselected model: ETS(A,N,N)
3. Checked residuals visually and using Ljung-Box test to ensure it is indistinguishable from white noise.
4. Forecasred for the next 100 days and reported accuracy.

```
train_apple %>%
  features(close, features = guerrero)
```

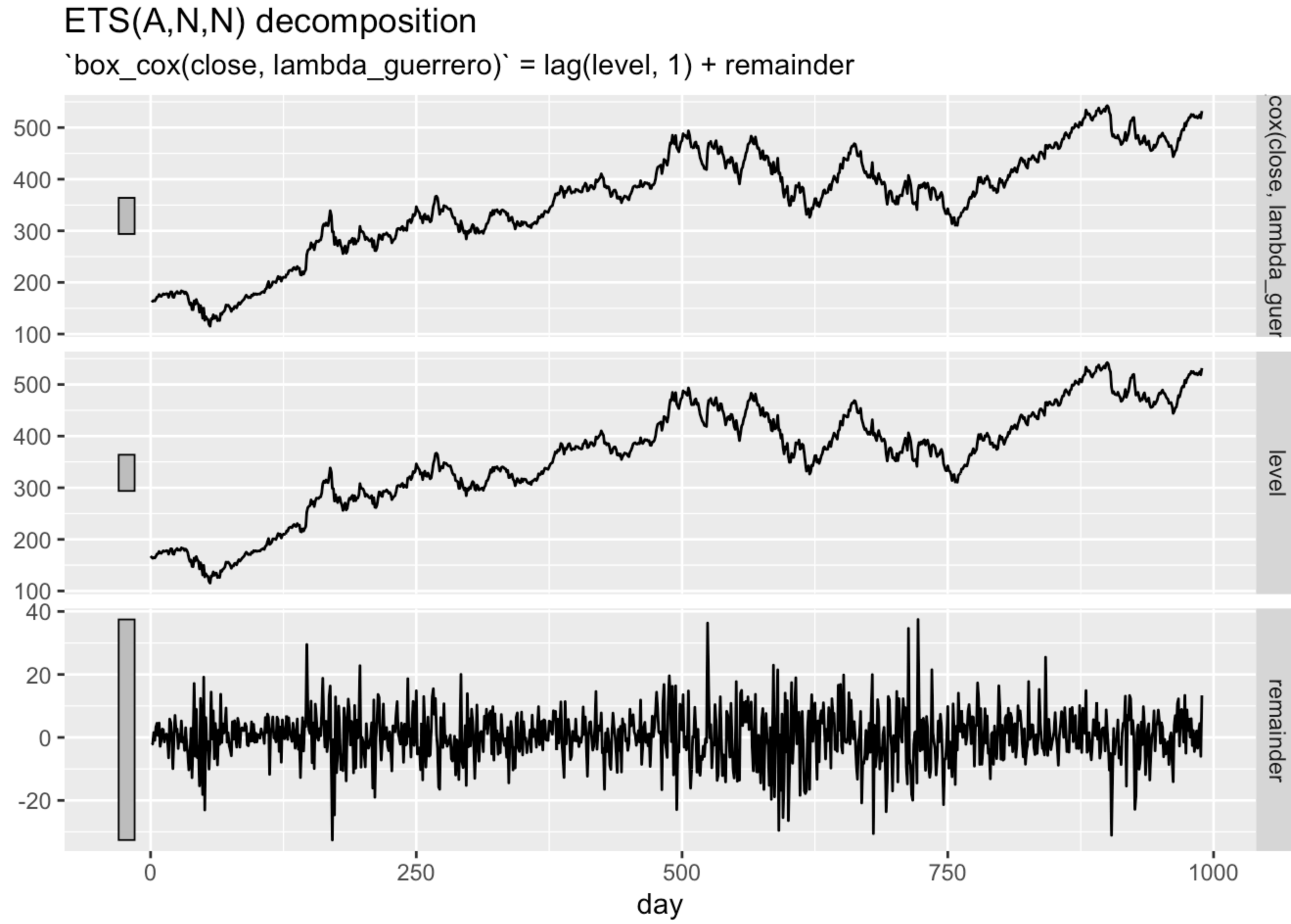
```
## # A tibble: 1 × 1
##   lambda_guerrero
##             <dbl>
## 1            1.23
```

```
lambda_guerrero = 1.232205

apple_ets <- train_apple |>
  model(ets = ETS(box_cox(close, lambda_guerrero)))

apple_ets |> report()
```

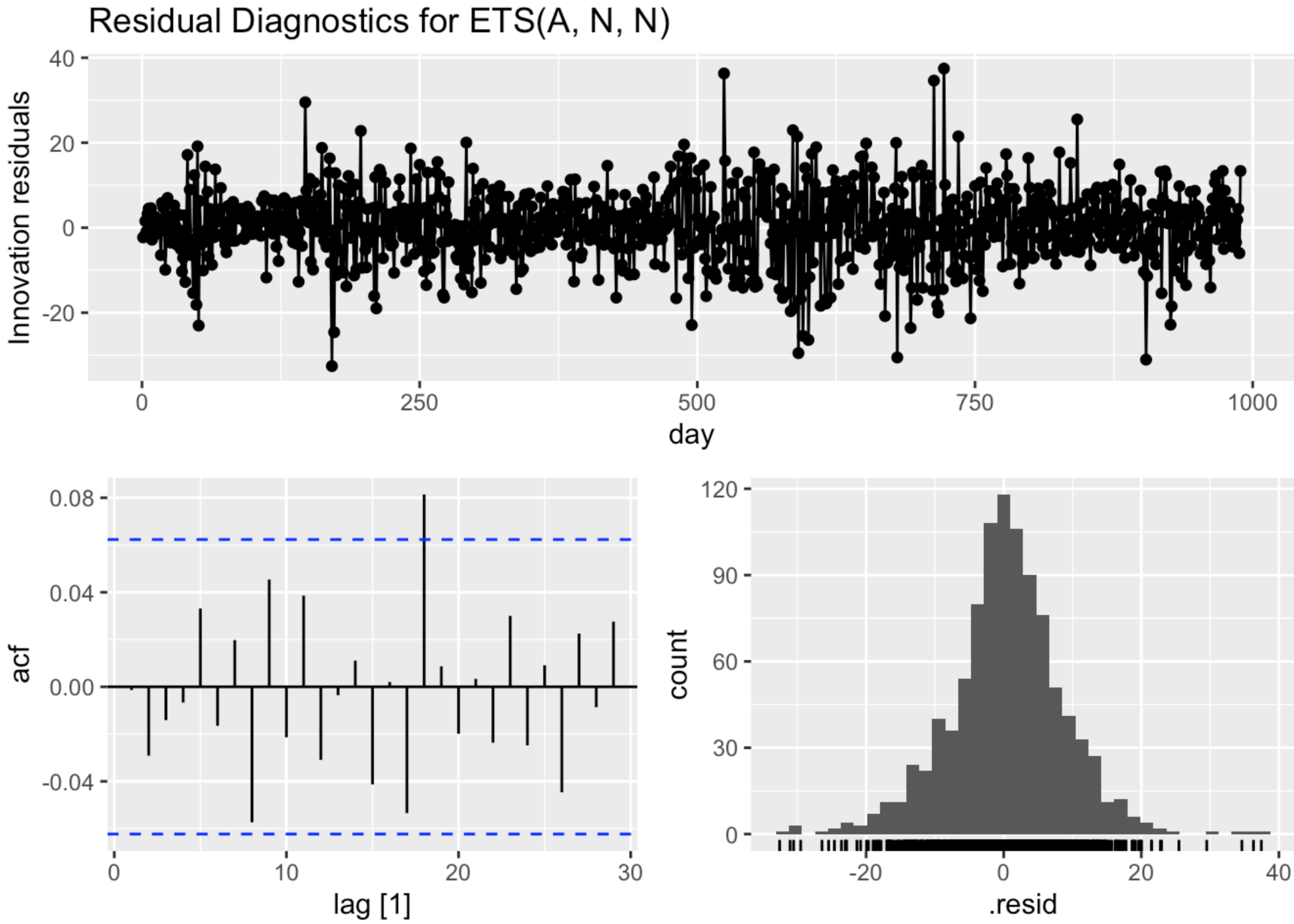
```
## Series: close
## Model: ETS(A,N,N)
## Transformation: box_cox(close, lambda_guerrero)
##   Smoothing parameters:
##     alpha = 0.9720357
##
##   Initial states:
##      l[0]
##  167.5884
##
##   sigma^2:  70.1303
##
##      AIC     AICc      BIC
## 11028.43 11028.45 11043.12
```

```
components(apple_ets) |> autoplot()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



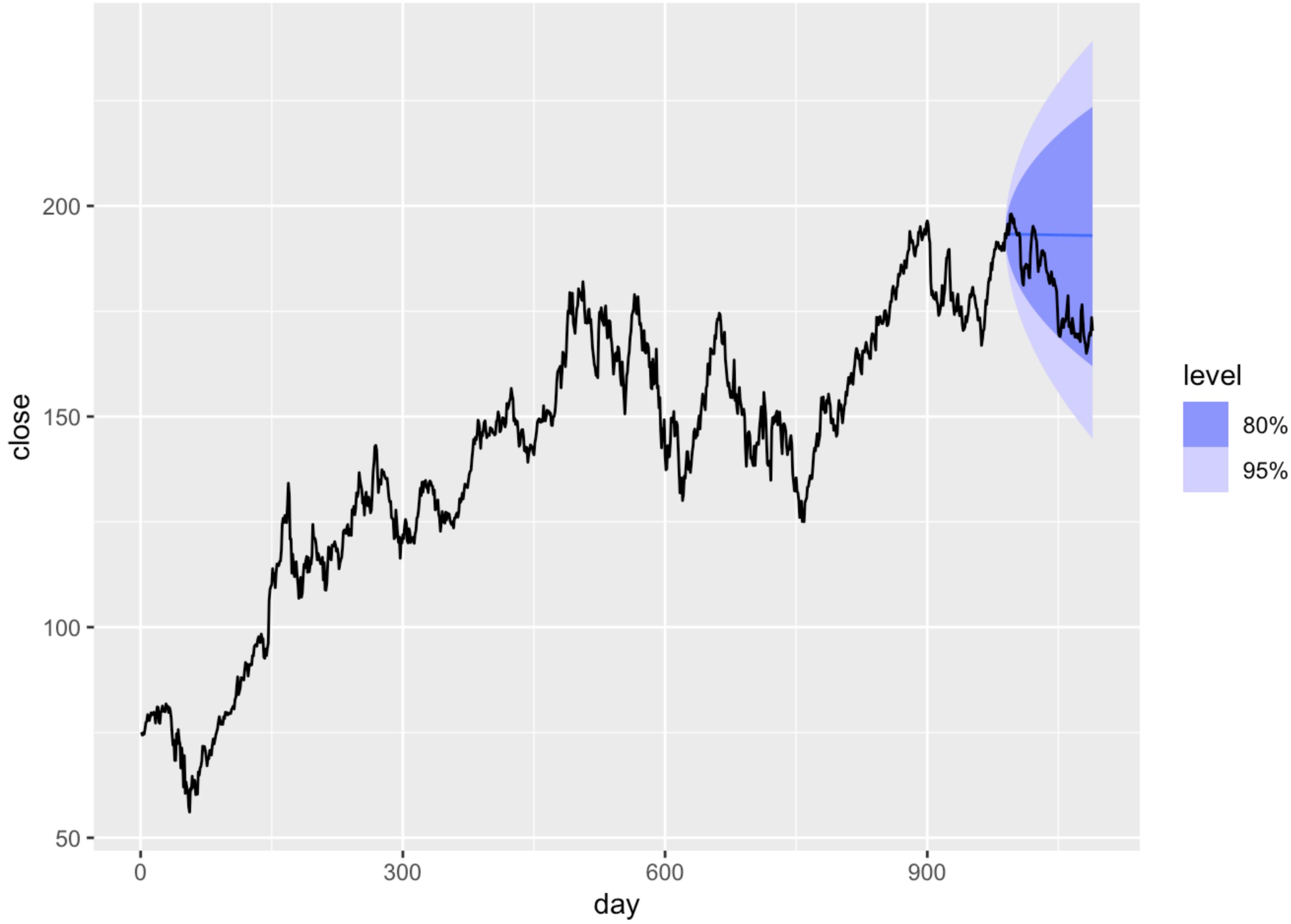

```
apple_ets |>
  gg_tsresiduals() +
  labs(title = "Residual Diagnostics for ETS(A, N, N)")
```




```
# using lag=min(2m, T/5) for seasonal data and 10 for non-seasonal data
augment(apple_ets) %>%
  features(.innov, ljung_box, lag = 10)
```

```
## # A tibble: 1 × 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ets       8.65     0.565
```

```
apple_ets_fc <- apple_ets %>%
  forecast(h = 100)

apple_ets_fc%>%
  autoplot(apple)
```




```
apple_ets_fc |> accuracy(apple)
```

```
## # A tibble: 1 × 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ets    Test  -11.8  15.3  12.5 -6.81  7.15  6.27  5.74 0.960
```

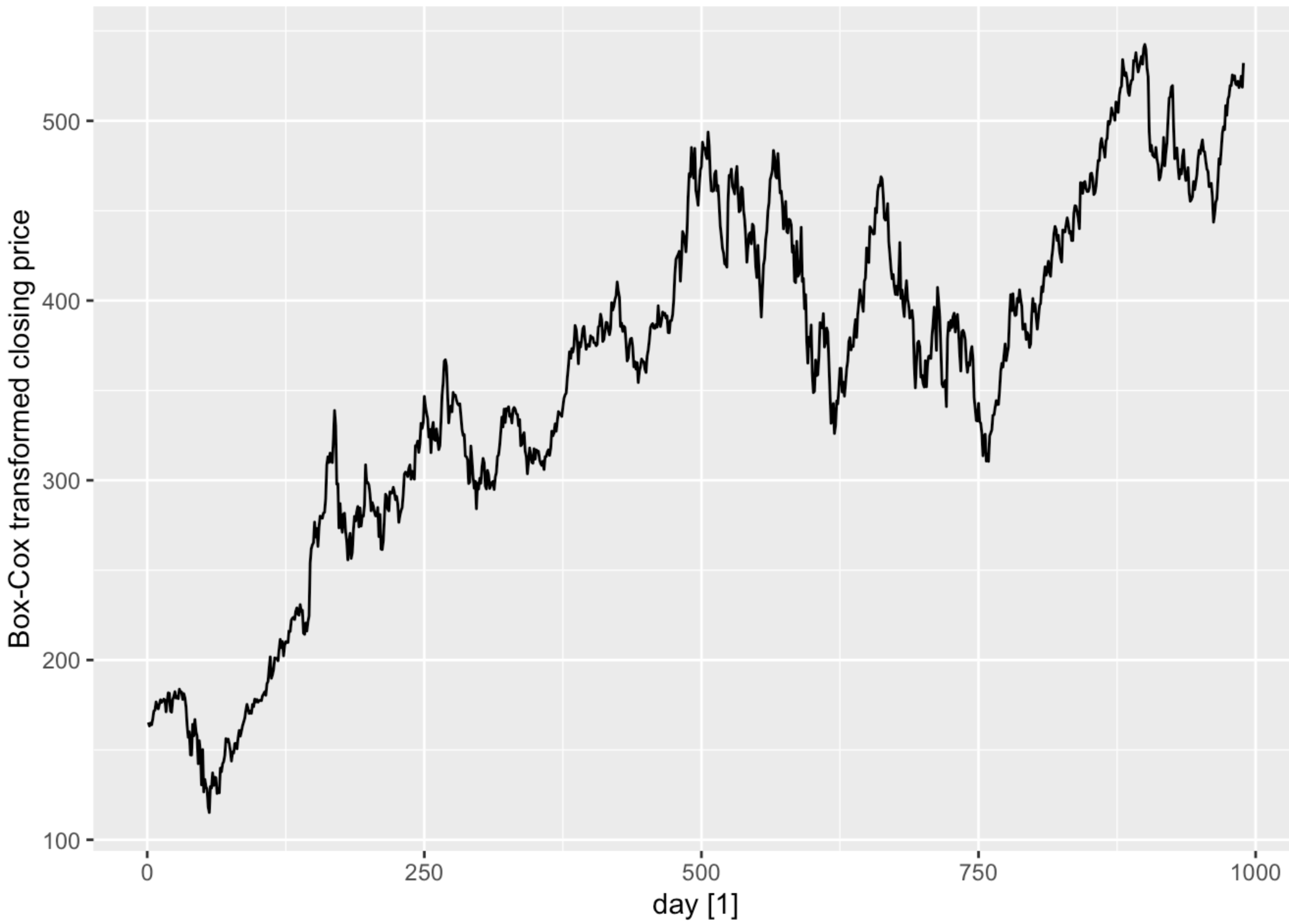
ARIMA Model

1. Used Box Cox transoformation to stabilize the variance
2. Converted the data to stationary for modelling
3. Created an ARIMA(2,1,4) model.
4. Ensured its residuals are indintinguishable from white noise-visually and using Ljung-Box test.
5. Reported the model’s performance.

```
train_apple %>%
  features(close, features = guerrero)
```

```
## # A tibble: 1 × 1
##   lambda_guerrero
##             <dbl>
## 1            1.23
```

```
lambda_guerrero = 1.232205

# transform data for constant variance
train_apple %>% autoplot(box_cox(close, lambda_guerrero)) +
  labs(y = "Box-Cox transformed closing price")
```




```
# check for number of seasonal differencing = 0
train_apple %>% features(box_cox(close, lambda_guerrero), unitroot_nsdiffs)
```

```
## # A tibble: 1 × 1
##   nsdiffs
##     <int>
## 1       0
```

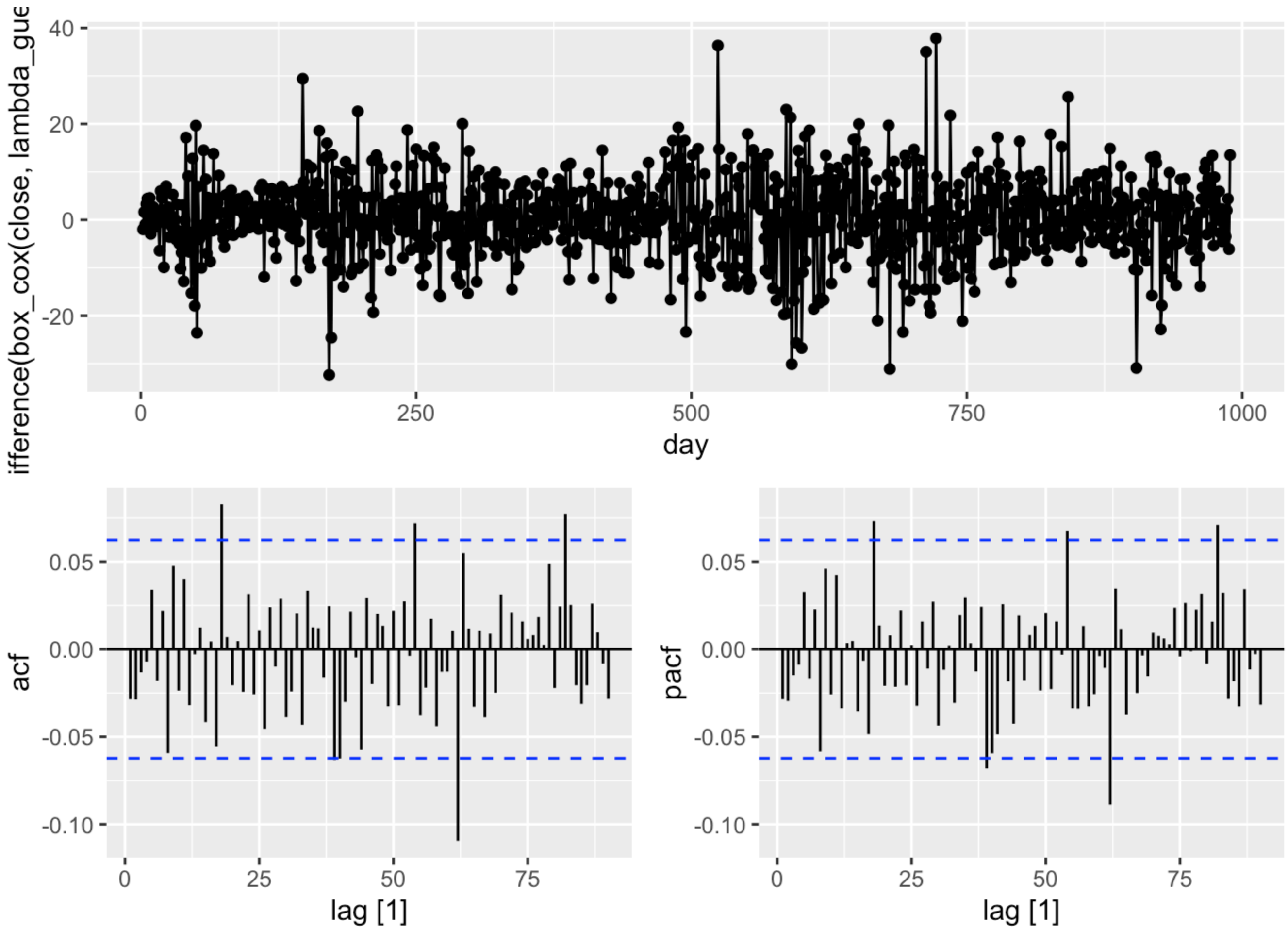
```
# check for first-order differencing = 1
train_apple |> features(box_cox(close, lambda_guerrero), unitroot_ndiffs)
```

```
## # A tibble: 1 × 1
##   ndiffs
##    <int>
## 1      1
```

```
gg_tsdisplay(train_apple, difference(box_cox(close, lambda_guerrero)), plot_type='partia
l', lag_max = 90)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Obs: ACF drops to 0 quickly, so we can say that the data is now stationary

train_apple |>
  features(difference(box_cox(close, lambda_guerrero)), unitroot_kpss)
```

```
## # A tibble: 1 × 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1    0.0483         0.1
```

```
# Obs: The data is stationary
```

```
arima_fit <- train_apple %>%
  model(
    arima_auto = ARIMA(box_cox(close, lambda_guerrero), stepwise = FALSE),
    # arima_auto = ARIMA(box_cox(sale, lambda_guerrero), stepwise = False, approx = Fals
e)
  )

arima_fit |> pivot_longer(everything(), names_to = "Model name",
                     values_to = "Orders")
```

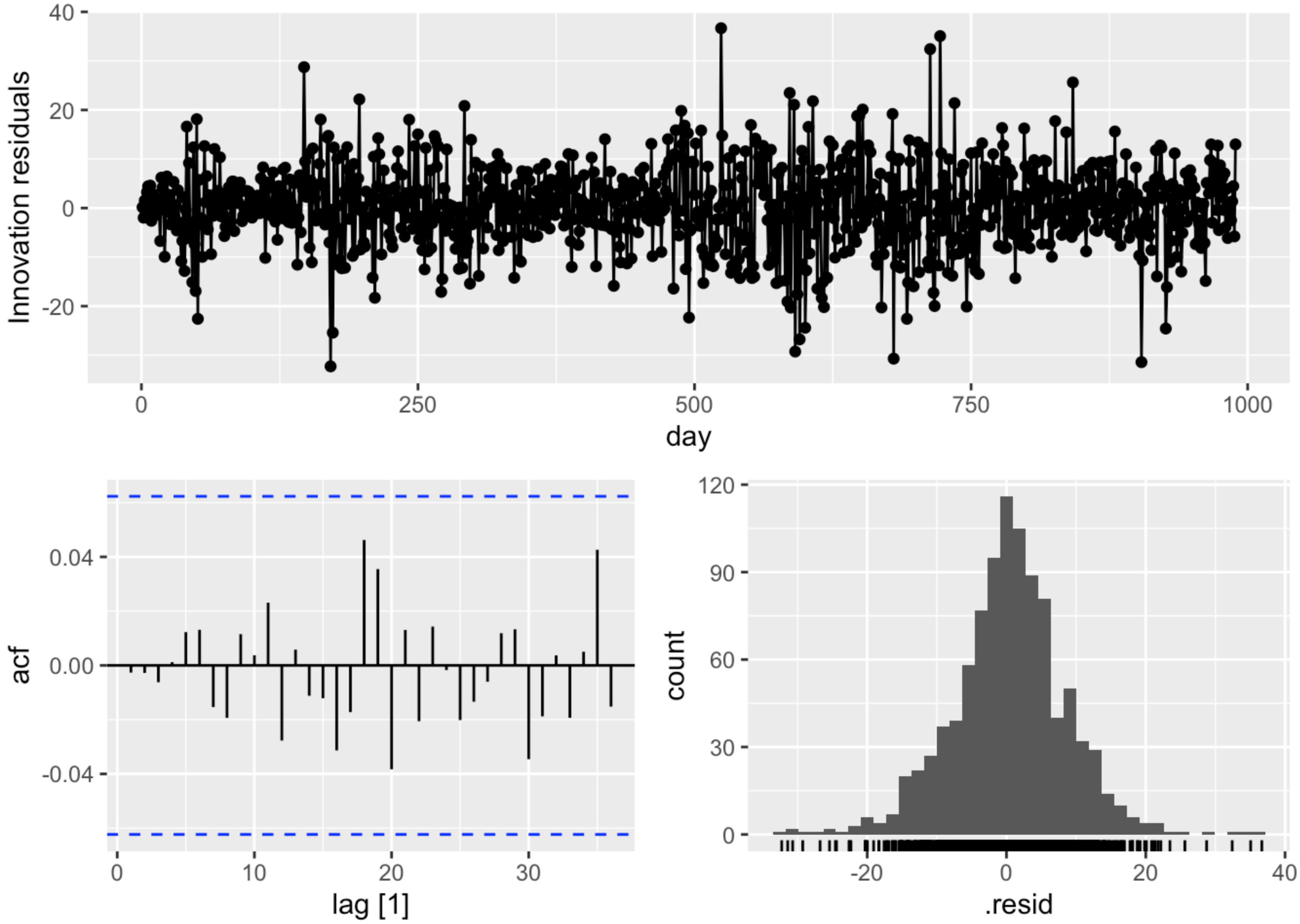
```
## # A mable: 1 x 2
## # Key:     Model name [1]
##   `Model name`         Orders
##   <chr>               <model>
## 1 arima_auto   <ARIMA(2,1,4)>
```

Autoselected model : ARIMA(2,1,4)

```
glance(arima_fit)
```

```
## # A tibble: 1 × 8
##   .model     sigma2 log_lik   AIC  AICc   BIC ar_roots  ma_roots
##   <chr>       <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
## 1 arima_auto   69.1  -3492. 6997. 6997. 7032. <cpl [2]> <cpl [4]>
```

```
arima_fit |> select(arima_auto) |> gg_tsresiduals(lag=36)
```

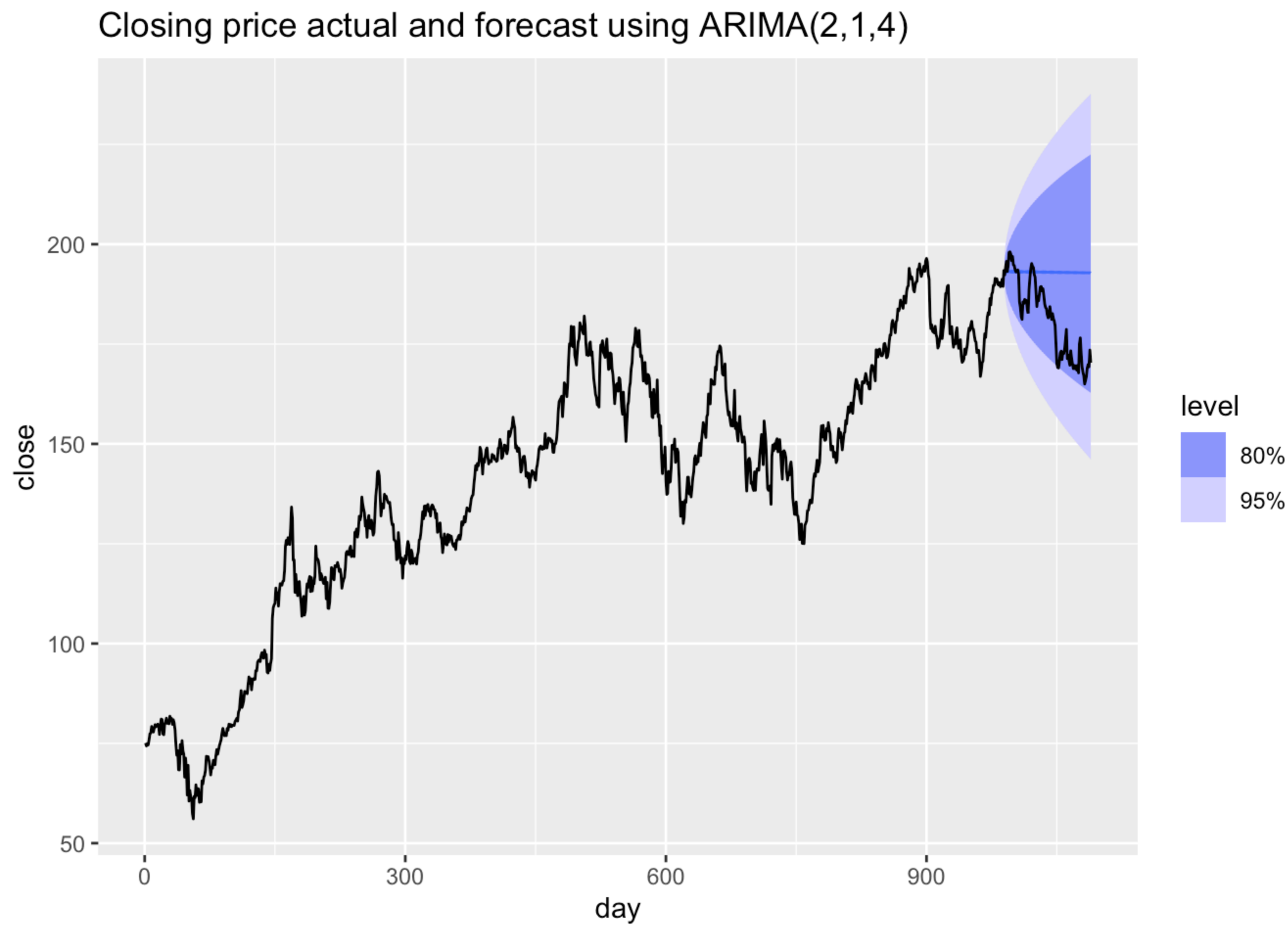



```
augment(arima_fit) |>
  filter(.model == "arima_auto") |>
  features(.innov, ljung_box, lag=24, dof=6)
```

```
## # A tibble: 1 × 3
##   .model     lb_stat lb_pvalue
##   <chr>        <dbl>     <dbl>
## 1 arima_auto    9.76     0.939
```

Observation: The residuals look like white noise, so the model is good. Hence its ready for forecasting.

```
forecast(arima_fit, h=100) |>
  filter(.model=='arima_auto') |>
  autoplot(apple) +
  labs(title = "Closing price actual and forecast using ARIMA(2,1,4)",
       )
```




```
forecast(arima_fit, h=100) |>
  filter(.model=='arima_auto') |>
  accuracy(apple)
```

```
## # A tibble: 1 × 10
##   .model     .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>      <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima_auto Test  -11.7  15.2  12.4 -6.74  7.10  6.22  5.70 0.960
```

Comparing models

We observe that ARIMA model performs better than the ETS model but their performance is close.

```
train_apple |>
   model(
   arima = ARIMA(box_cox(close, lambda_guerrero) ~ pdq(2,1,4)),
   ets = ETS(box_cox(close, lambda_guerrero) ~ error("A") + trend("N") + season("N"))
   ) |>
   forecast(h = 100) |>
   accuracy(apple)
```

```
## # A tibble: 2 × 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  -11.7  15.2  12.4 -6.74  7.10  6.22  5.70 0.960
## 2 ets    Test  -11.8  15.3  12.5 -6.81  7.15  6.27  5.74 0.960
```