Lecture 10 Code Examples

1 Misspecified ARMA models

Let's generate data from an AR(2) model. What happens if we fit a misspecified model, i.e. if we choose the wrong order for p.

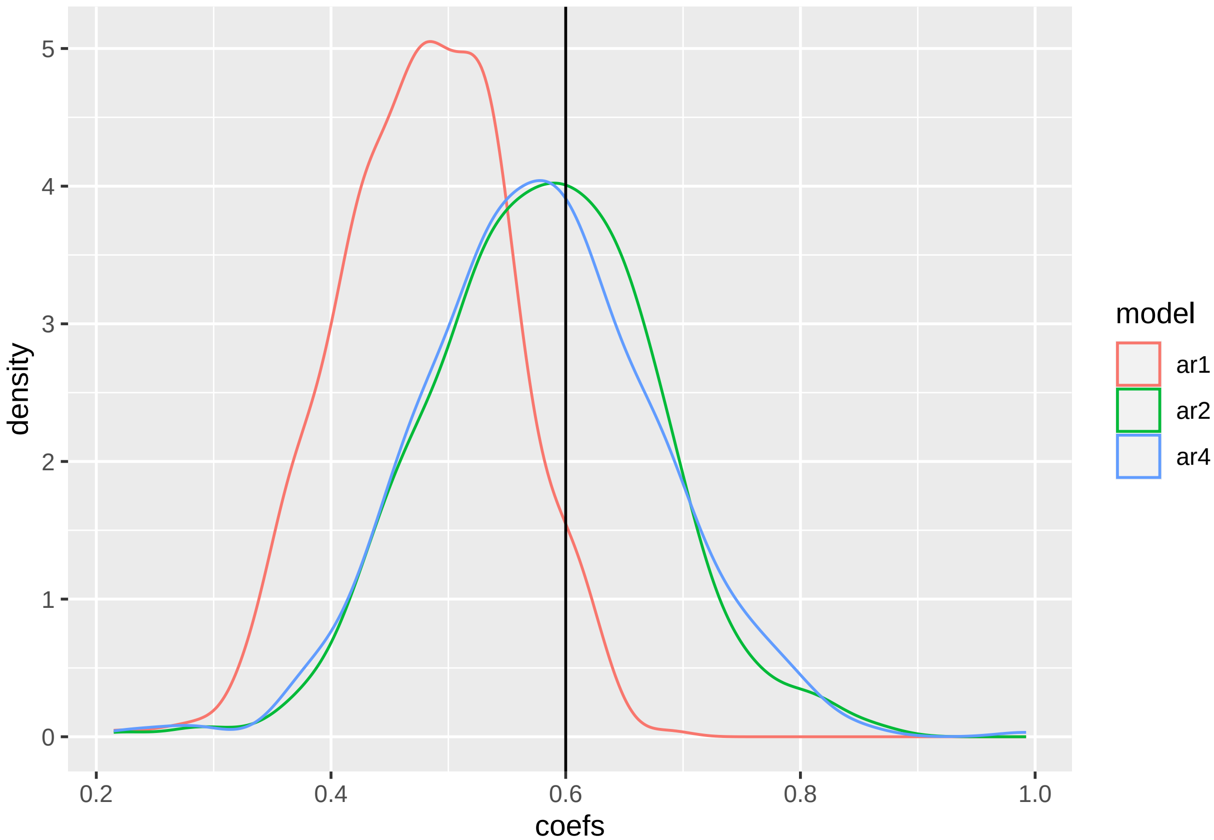
```
set.seed(5209)
ar2_data <- arima.sim(model = list(ar = c(0.6, -0.2)), n = 100)
ar1_model <- arima(ar2_data, order = c(1, 0, 0))
ar2_model <- arima(ar2_data, order = c(2, 0, 0))
ar4_model <- arima(ar2_data, order = c(4, 0, 0))
```

```
set.seed(5209)
B <- 500
ar2_data <- map(1:B, ~ arima.sim(model = list(ar = c(0.6, -0.2)), n = 100))
ar1_model_coefs_ <- map(ar2_data, ~ arima(., order = c(1, 0, 0))$coef) |>
  transpose() |>
  map(unlist) |>
  as.tibble()
```

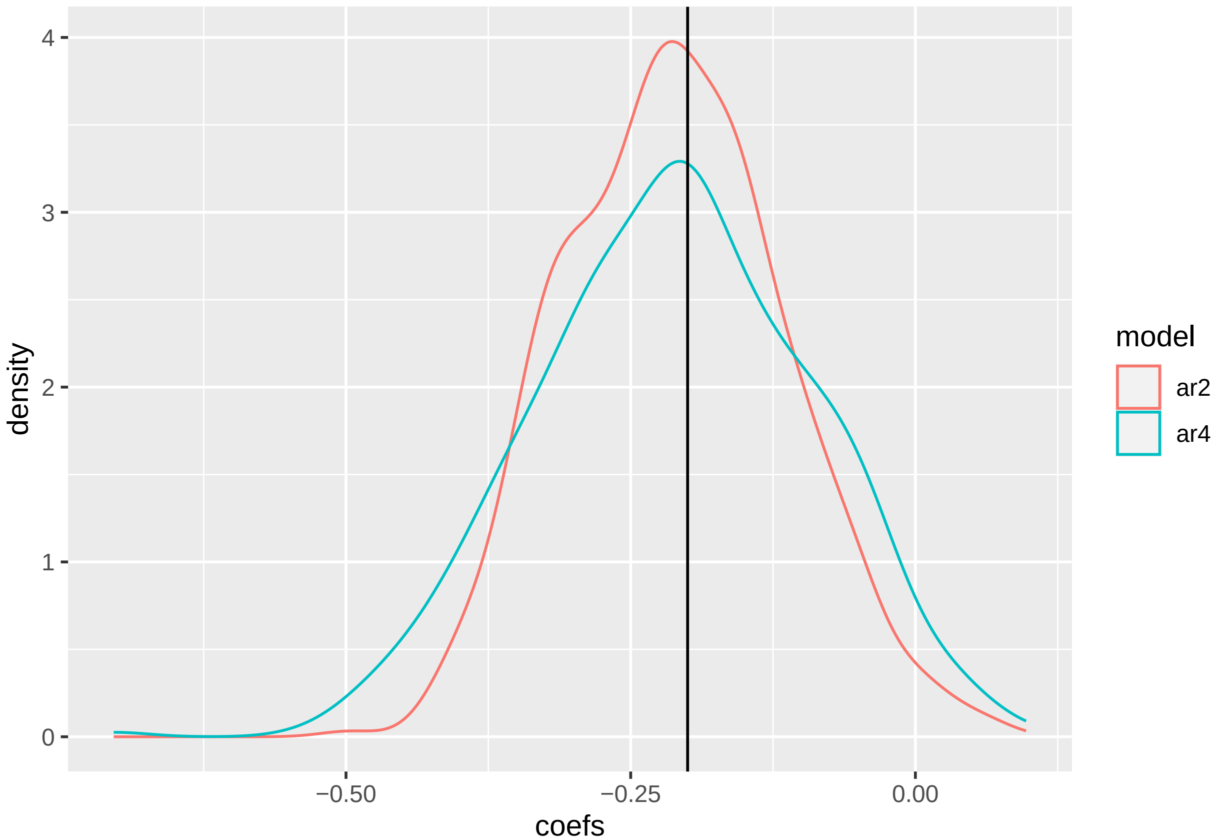
```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` instead.
## i The signature and semantics have changed, see `?as_tibble`.
```

```
ar2_model_coefs_ <- map(ar2_data, ~ arima(., order = c(2, 0, 0))$coef) |>
  transpose() |>
  map(unlist) |>
  as.tibble()
ar4_model_coefs_ <- map(ar2_data, ~ arima(., order = c(4, 0, 0))$coef) |>
  transpose() |>
  map(unlist) |>
  as.tibble()
```

```
phi1_coefs <- tibble(ar1 = ar1_model_coefs_$ar1,
                     ar2 = ar2_model_coefs_$ar1,
                     ar4 = ar4_model_coefs_$ar1)
phi1_coefs |>
  pivot_longer(cols = everything(),
               names_to = "model",
               values_to = "coefs") |>
  ggplot() +
  geom_density(aes(x = coefs, color = model)) +
  geom_vline(xintercept = 0.6)
```




```r
phi2_coefs <- tibble(ar2 = ar2_model_coefs_$ar2,
                     ar4 = ar4_model_coefs_$ar2)
phi2_coefs |>
  pivot_longer(cols = everything(),
               names_to = "model",
               values_to = "coefs") |>
  ggplot() +
  geom_density(aes(x = coefs, color = model)) +
  geom_vline(xintercept = -0.2)
```



2 Real data analysis

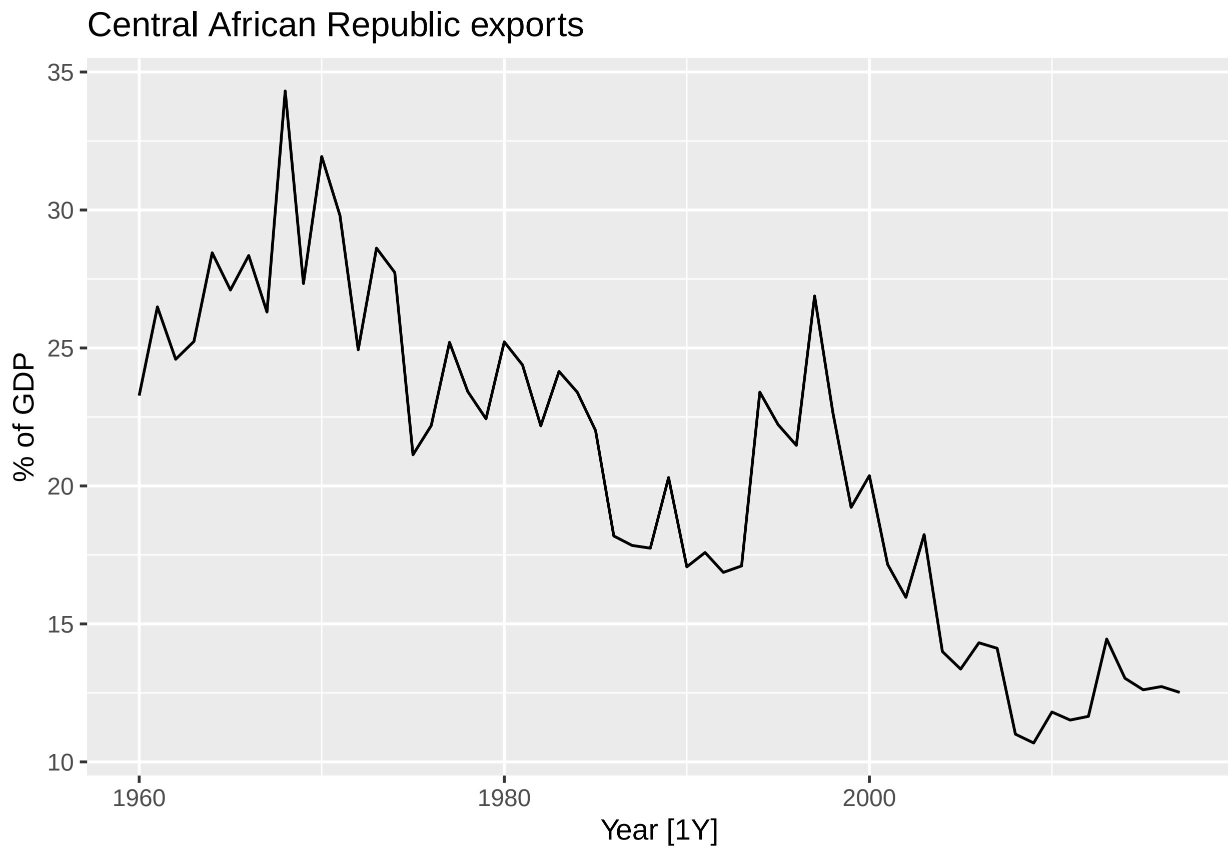
2.1 Working with tsibble

The `tsibble` package allows us to work with multiple time series in one data frame. For instance, consider the `global_economy` data frame, which contains economic indicators featured by the World Bank from 1960 to 2017. Each time series is identified by a `Key`. The time series may be multivariate, i.e. have multiple columns.

2.2 CAF exports

Let us try to model exports from the Central African Republic.

```
caf_economy <- global_economy |>
  filter(Code == "CAF")
caf_economy |>
  autoplot(Exports) +
  labs(title="Central African Republic exports",
       y="% of GDP")
```

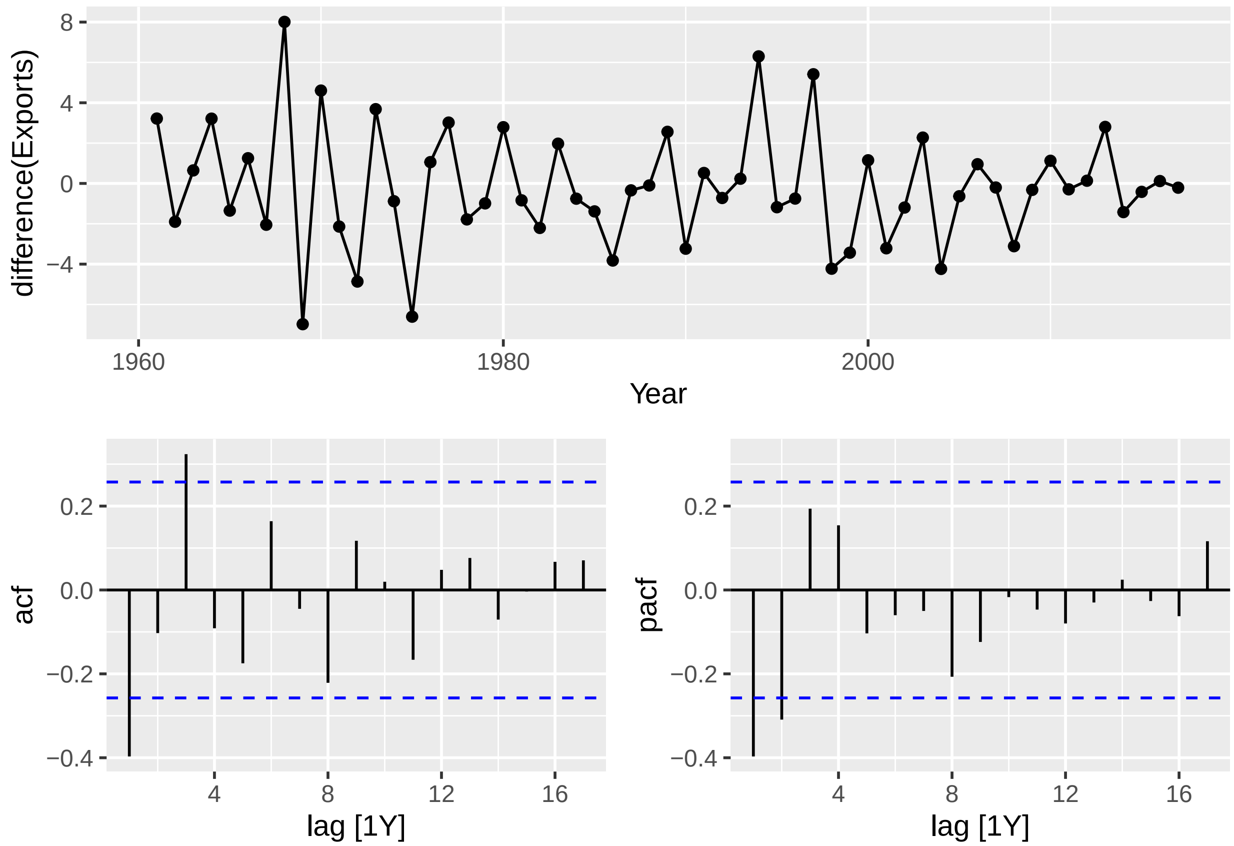


This is non-stationary, so we can take a first difference.

```
caf_economy |>
  gg_tsdisplay(difference(Exports), plot_type='partial')
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



From the ACF and PACF plots, it seems that either an AR(2) or MA(3) model is appropriate for the residuals. We hence fit these two models, and also try automatic model search (we will discuss this more next week). We also fit an AR(5) model for comparison. The `fable` package makes fitting all 3 models at the same time extremely easy. The result is a `mable`, i.e. a dataframe of models.

```
caf_fit <- caf_economy |>
  model(arima210 = ARIMA(Exports ~ pdq(2,1,0)),
        arima013 = ARIMA(Exports ~ pdq(0,1,3)),
        arima510 = ARIMA(Exports ~ pdq(5,1,0)),
        auto = ARIMA(Exports))

caf_fit |> glance()
```

```
## # A tibble: 4 x 9
##   Country                  .model sigma2 log_lik   AIC  AICc   BIC ar_ro~1 ma_ro~2
##   <fct>                    <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>  <list>
## 1 Central African Repub~ arima~   6.71   -134.  275.  275.  281. <cpl>   <cpl>
## 2 Central African Repub~ arima~   6.54   -133.  274.  275.  282. <cpl>   <cpl>
## 3 Central African Repub~ arima~   6.52   -132.  276.  278.  288. <cpl>   <cpl>
## 4 Central African Repub~ auto     6.42   -132.  274.  275.  284. <cpl>   <cpl>
## # ... with abbreviated variable names 1: ar_roots, 2: ma_roots
```

We see that while ARIMA(5,1,0) has the largest log likelihood, it has the largest AIC and AICc (smaller is better). The AIC and AICc of the other 3 models are comparable. Finally, we check the order of the model found by automatic model search: We got an ARIMA(2,1,2) model.

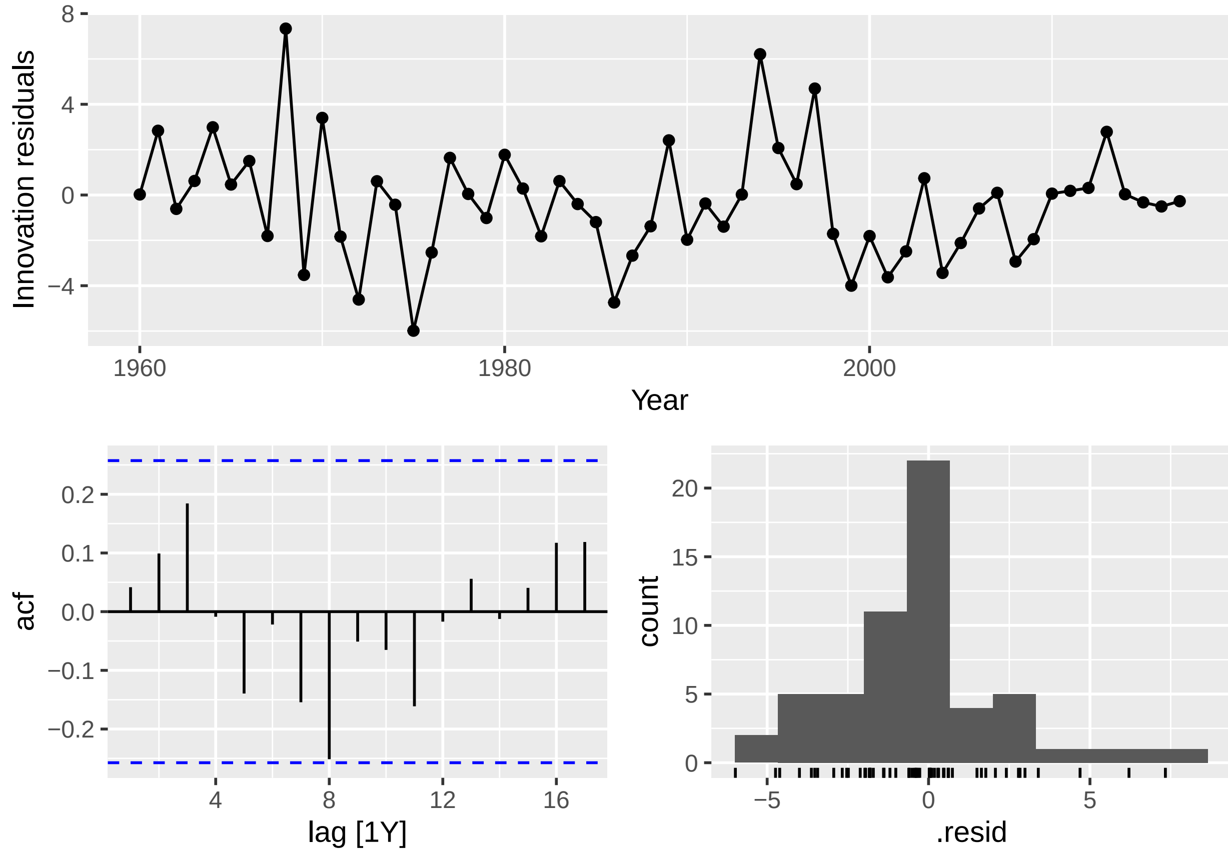
```
caf_fit["auto"]
```

```
## # A tibble: 1 x 1
```

```
##             auto
##          <model>
## 1 <ARIMA(2,1,2)>
```

We now do a residual diagnosis. To see if

```
caf_fit |>
  select(arima210) |>
  gg_tsresiduals()
```



The `augment` method produces the fitted and residual values for each model.

```
augment(caf_fit)
```

```
## # A tsibble: 232 x 7 [1Y]
## # Key:       Country, .model [4]
##    Country                  .model    Year Exports .fitted  .resid  .innov
##    <fct>                    <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Central African Republic arima210  1960    23.3    23.2  0.0233  0.0233
##  2 Central African Republic arima210  1961    26.5    23.7  2.83    2.83
##  3 Central African Republic arima210  1962    24.6    25.2 -0.613  -0.613
##  4 Central African Republic arima210  1963    25.2    24.6  0.619   0.619
##  5 Central African Republic arima210  1964    28.4    25.5  2.99    2.99
##  6 Central African Republic arima210  1965    27.1    26.6  0.461   0.461
##  7 Central African Republic arima210  1966    28.4    26.9  1.50    1.50
##  8 Central African Republic arima210  1967    26.3    28.1 -1.81   -1.81
##  9 Central African Republic arima210  1968    34.3    27.0  7.34    7.34
## 10 Central African Republic arima210  1969    27.3    30.9 -3.53   -3.53
## # ... with 222 more rows
```

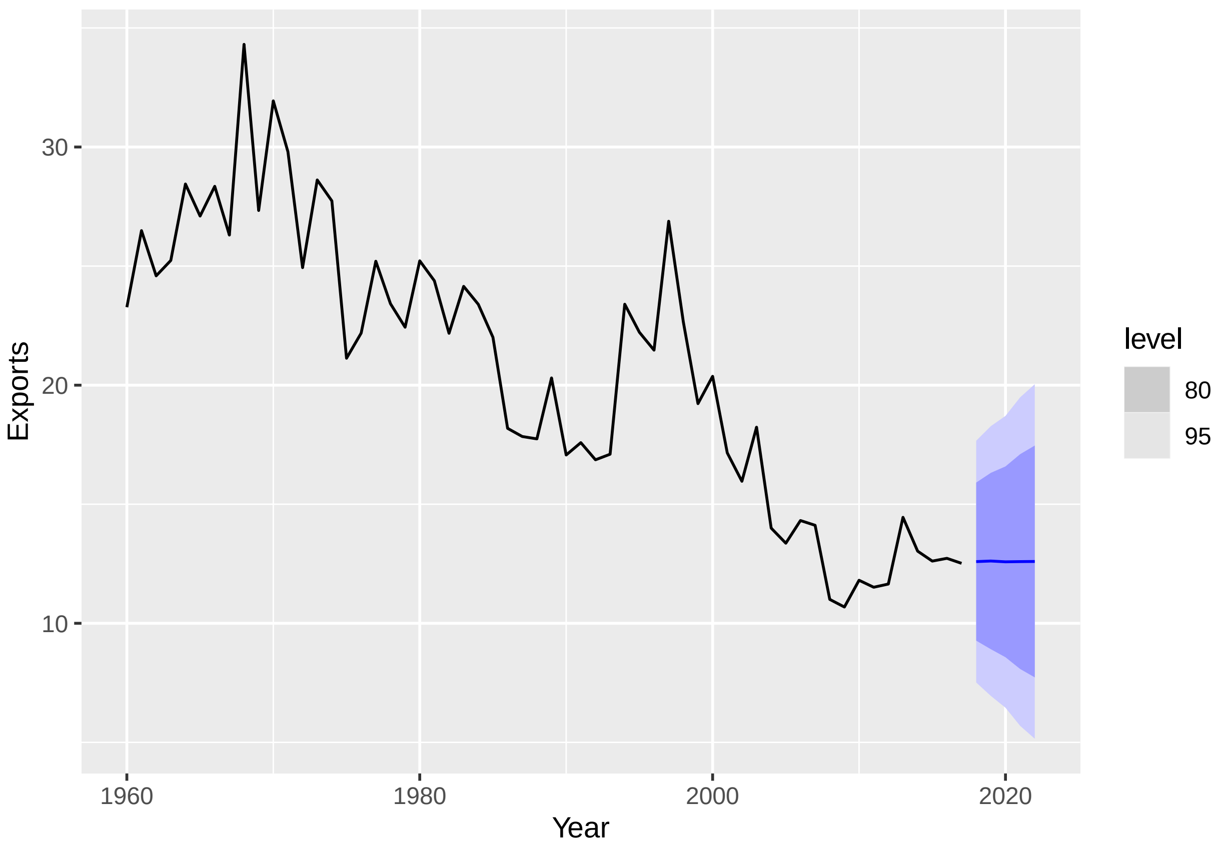
We can now use the residuals to compute a Ljung-Box test statistic for each model. We see that the p-values are large, so in each case, the residuals are well-approximated by a white noise sequence.

```
augment(caf_fit) |>
  # filter(.model=='arima210') |>
  features(.innov, ljung_box, lag = 10, dof = 3)
```

```
## # A tibble: 4 x 4
##   Country                  .model   lb_stat lb_pvalue
##   <fct>                    <chr>      <dbl>     <dbl>
## 1 Central African Republic arima013    5.64     0.582
## 2 Central African Republic arima210   10.7      0.152
## 3 Central African Republic arima510    3.58     0.827
## 4 Central African Republic auto        4.12     0.766
```

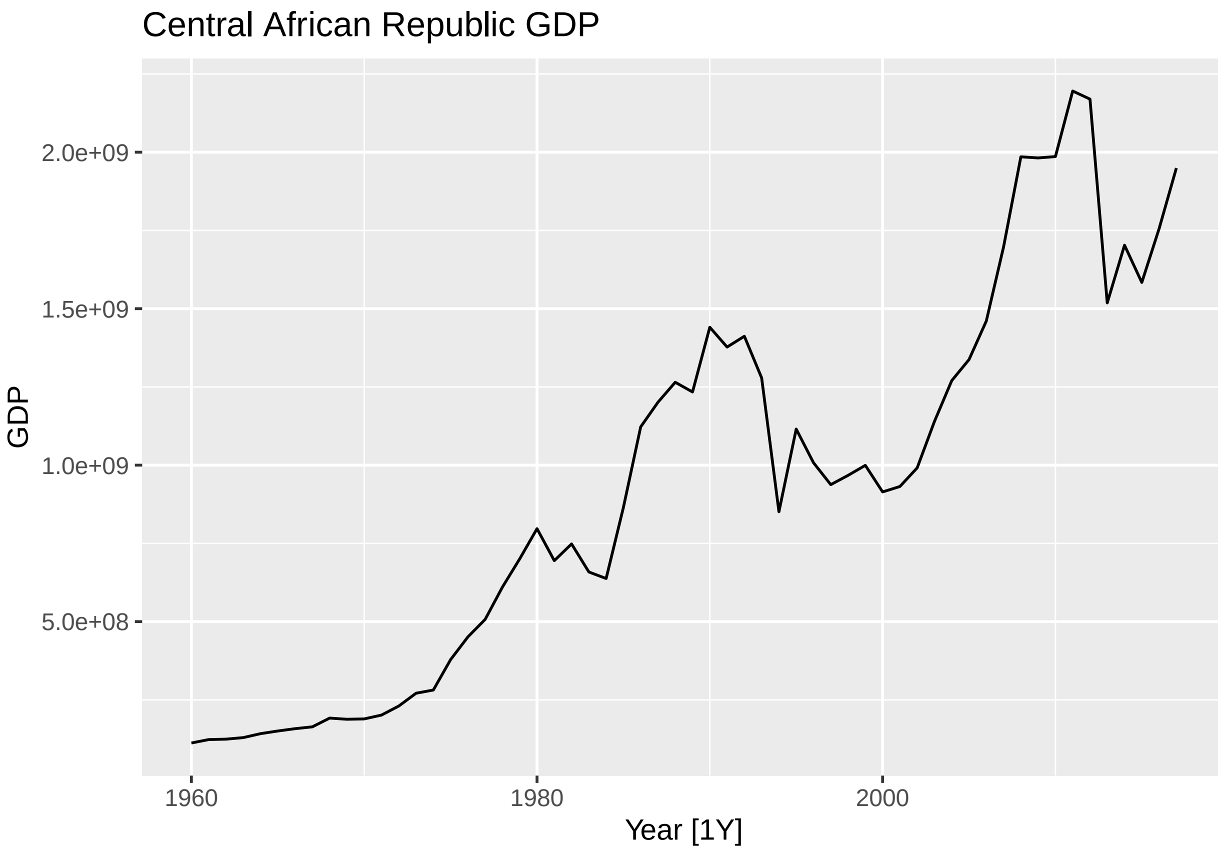
Finally, we can forecast using our model.

```
caf_fit |>
  forecast(h=5) |>
  filter(.model=='arima210') |>
  autoplot(global_economy)
```



CAF GDP: Understanding ARIMA models

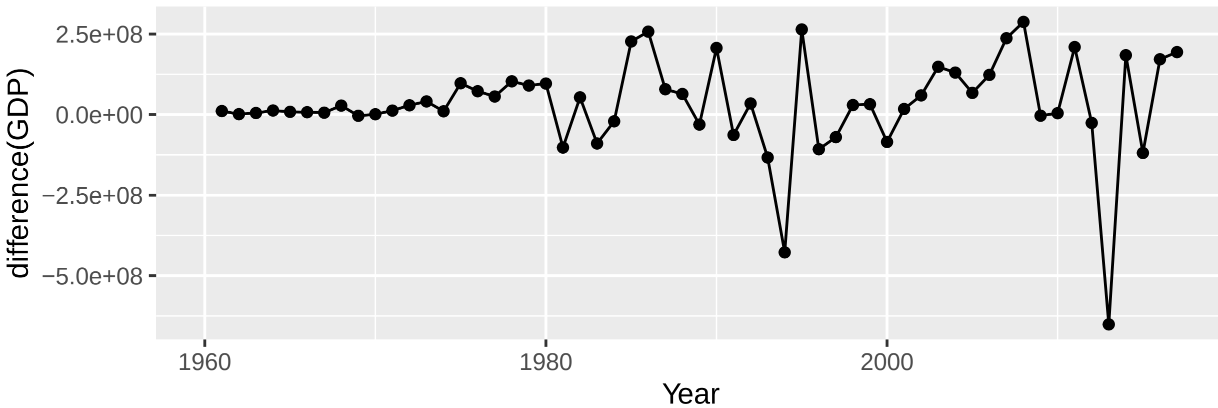
```
caf_economy |>
  autoplot(GDP) +
  labs(title="Central African Republic GDP")
```

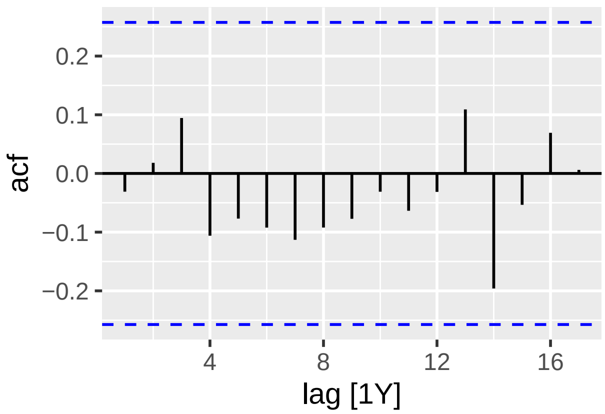


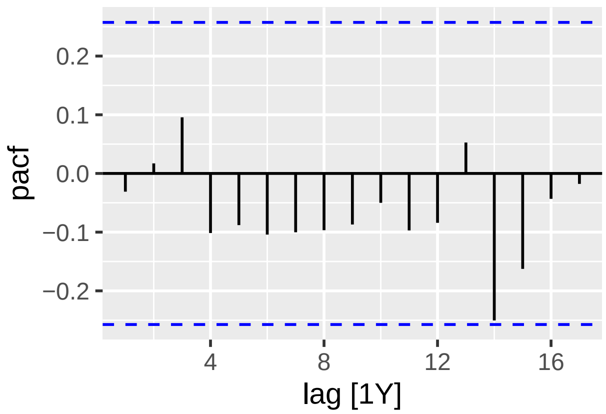

```
caf_economy |>
  gg_tsdisplay(difference(GDP), plot_type='partial')
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).

## Warning: Removed 1 rows containing missing values (`geom_point()`).
```








```
caf_fit <- caf_economy |>
  model(arima200 = ARIMA(GDP ~ pdq(2,0,0)),
        arima210 = ARIMA(GDP ~ pdq(2,1,0)),
        arima220 = ARIMA(GDP ~ pdq(2,2,0)),
        auto = ARIMA(GDP))
```

```
## Warning: It looks like you're trying to fully specify your ARIMA model but have not said if a constan
## You can include a constant using `ARIMA(y~1)` to the formula or exclude it by adding `ARIMA(y~0)`.

## Warning: 1 error encountered for arima200
## [1] Could not find an appropriate ARIMA model.
## This is likely because automatic selection does not select models with characteristic roots that may
## For more details, refer to https://otexts.com/fpp3/arima-r.html#plotting-the-characteristic-roots
```

```
caf_fit |> glance()
```

```
## # A tibble: 3 x 9
##   Country                .model  sigma2 log_lik   AIC  AICc   BIC ar_ro~1 ma_ro~2
##   <fct>                  <chr>    <dbl>   <dbl> <dbl> <dbl> <dbl> <list>  <list>
## 1 Central African Repu~ arima~ 2.41e16  -1155. 2316. 2316. 2322. <cpl>   <cpl>
## 2 Central African Repu~ arima~ 2.94e16  -1140. 2287. 2287. 2293. <cpl>   <cpl>
## 3 Central African Repu~ auto   2.27e16  -1154. 2311. 2312. 2316. <cpl>   <cpl>
## # ... with abbreviated variable names 1: ar_roots, 2: ma_roots
```

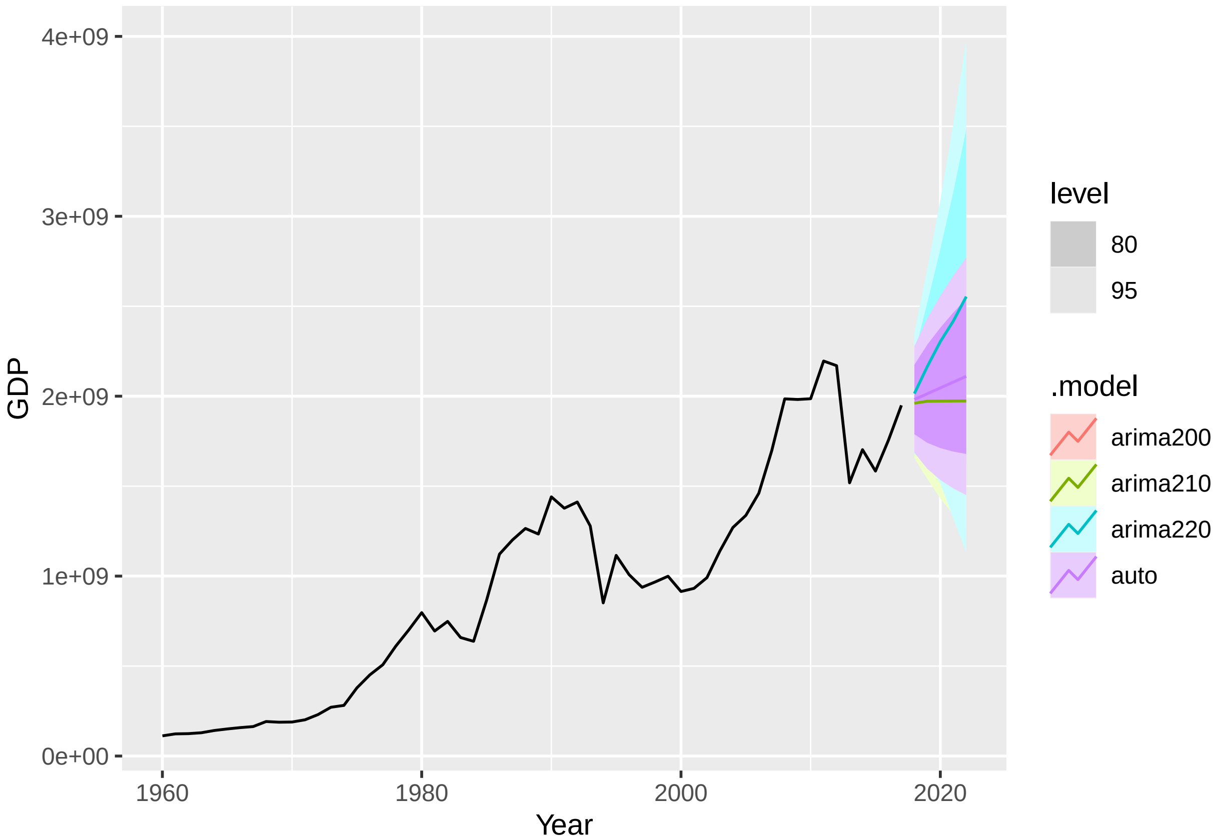
```
caf_fit |>
  forecast(h=5) |>
  autoplot(global_economy)
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
```

```
## -Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf

## Warning: Removed 5 rows containing missing values (`()`).
```



Understanding ARIMA models: - If $c = 0$, $d = 0$, long-term forecasts will tend to 0 - If $c = 0$, $d = 1$, long-term forecasts will tend to a nonzero constant - If $c = 0$, $d = 2$, long-term forecasts will follow a straight line -