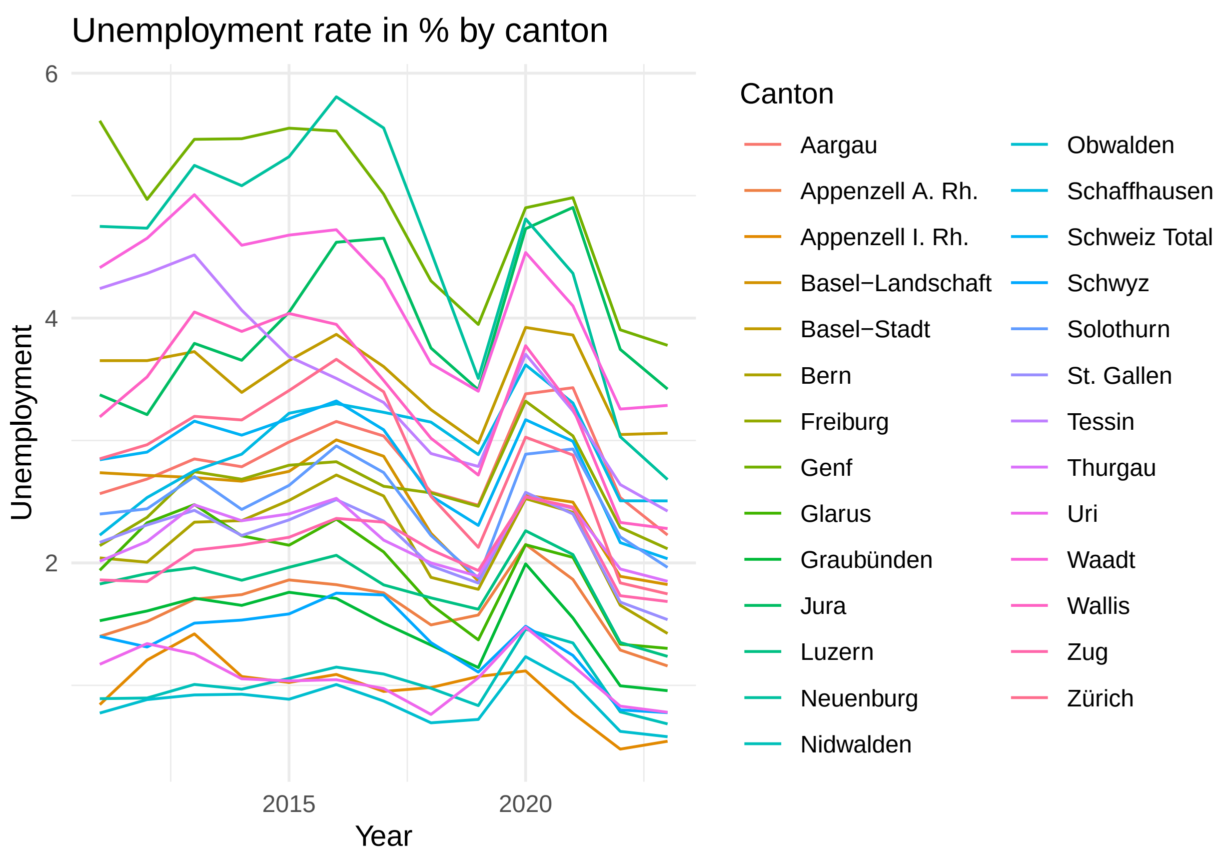
Homework2

Erdan Beka, Cyril Scheuermann, Roman Krass, Keijo Nierula

2024-05-21

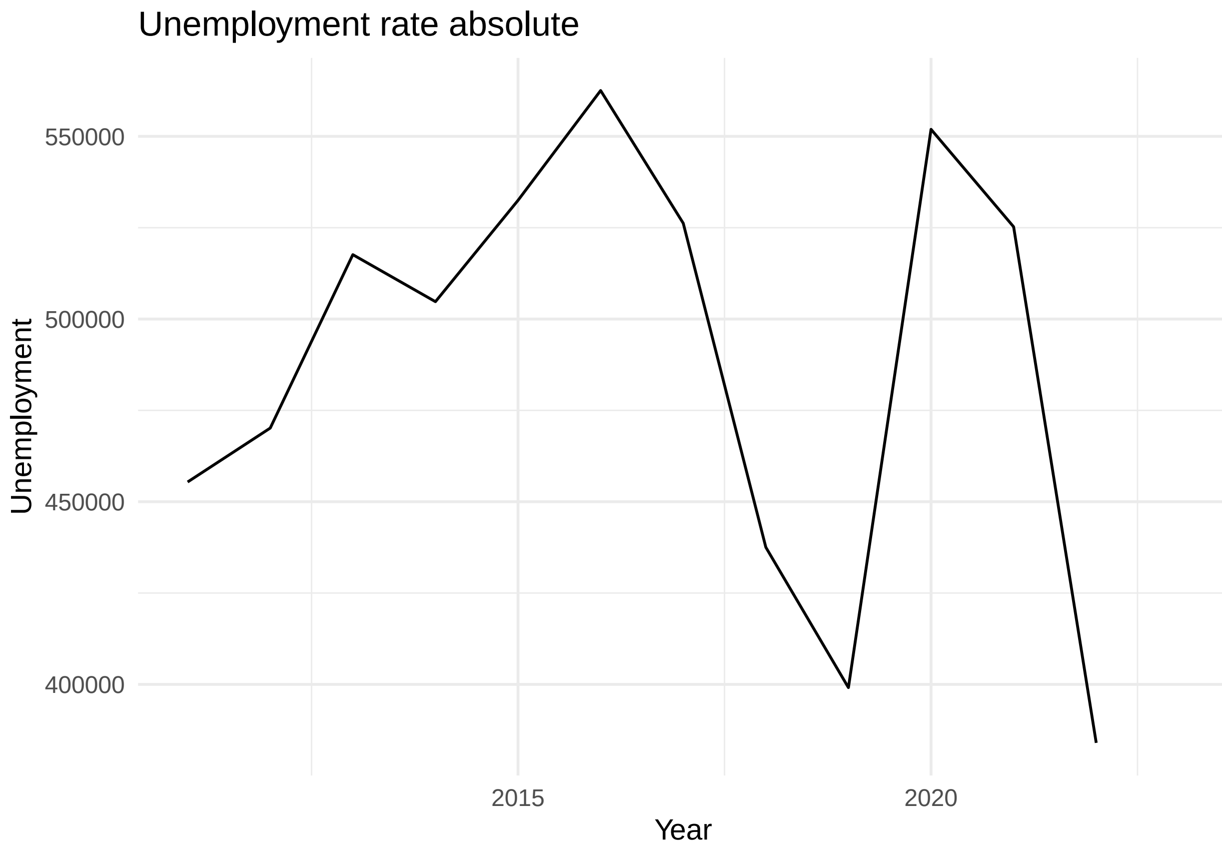
```r
# visualize the data for each canton
data |>
    ggplot(aes(x = Year, y = Unemployment, color = Kanton)) +
    geom_line() +
    theme_minimal() +
    labs(title = "Unemployment rate in % by canton", color = "Canton")
```




```r
# visualize the data summarized
data |>
    index_by(Year) |>
    summarise(Unemployment = sum(Unemployment_number, na.rm = FALSE)) |>
    ggplot(aes(x = Year, y = Unemployment)) +
    geom_line() +
```

```r
    theme_minimal() +
    labs(title = "Unemployment rate absolute")
```

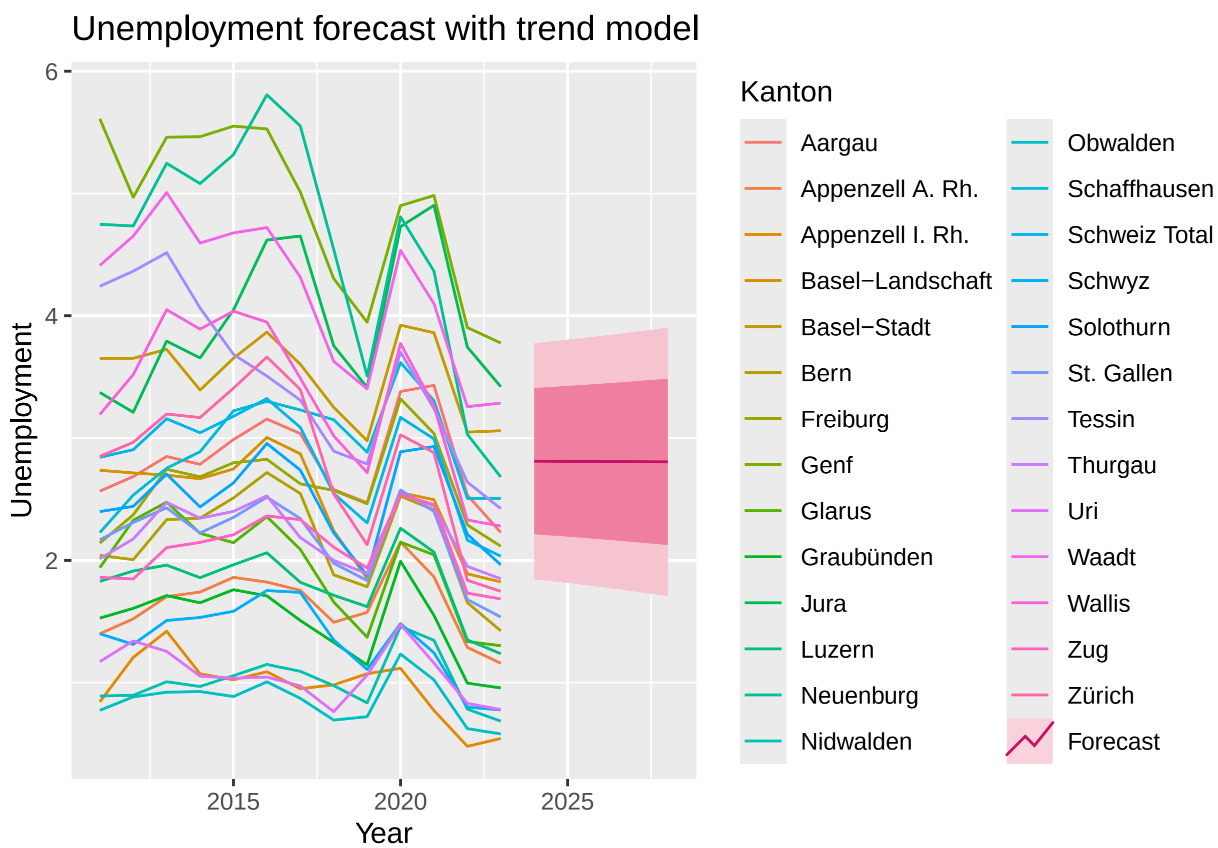
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



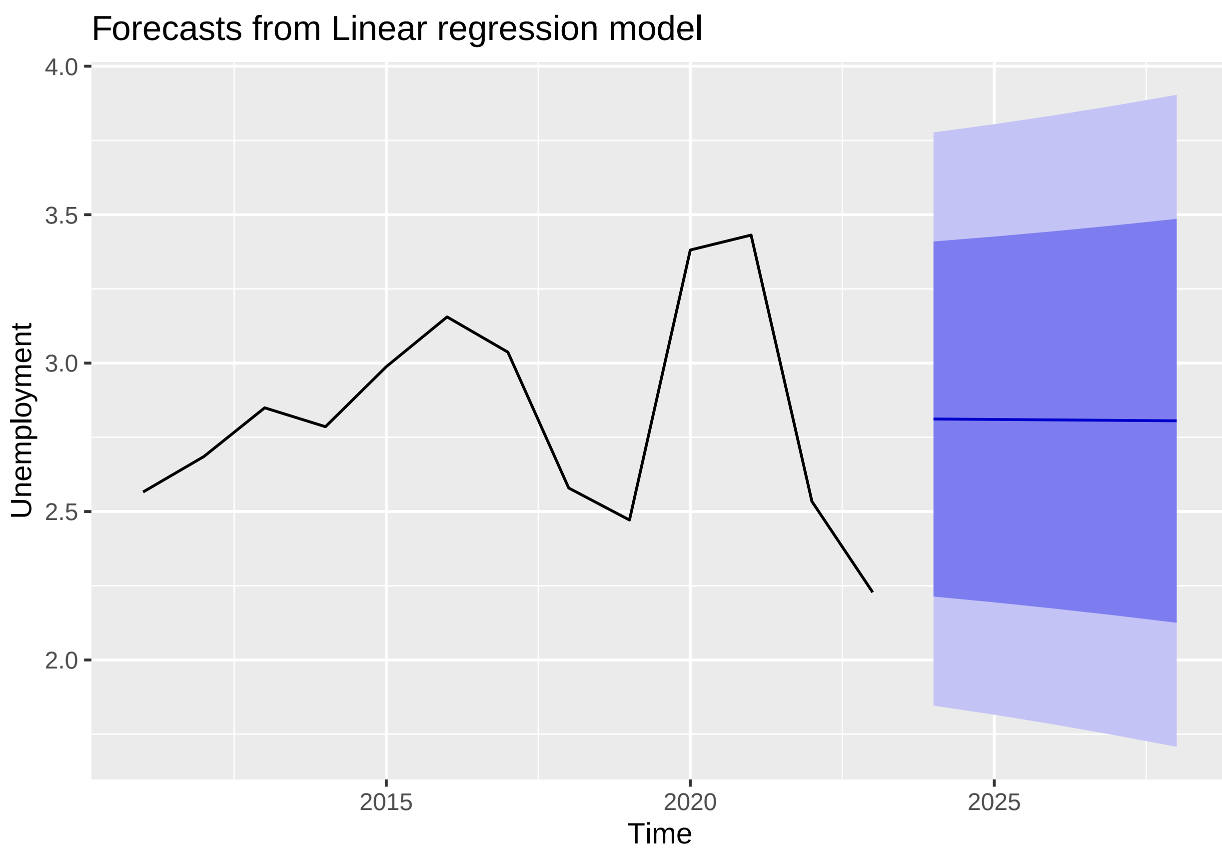

```r
# Define trend model for the Unemployment

fit <- tslm(Unemployment ~ trend, data = data_ts)

#
fit_fc <- fit |>
    forecast(h = 5)

autoplot(data_tsibble, series = "Data") +
    autolayer(fit_fc, series = "Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment forecast with trend model")
```

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

```
## Warning in geom_line(...): Ignoring unknown parameters: `series`
```

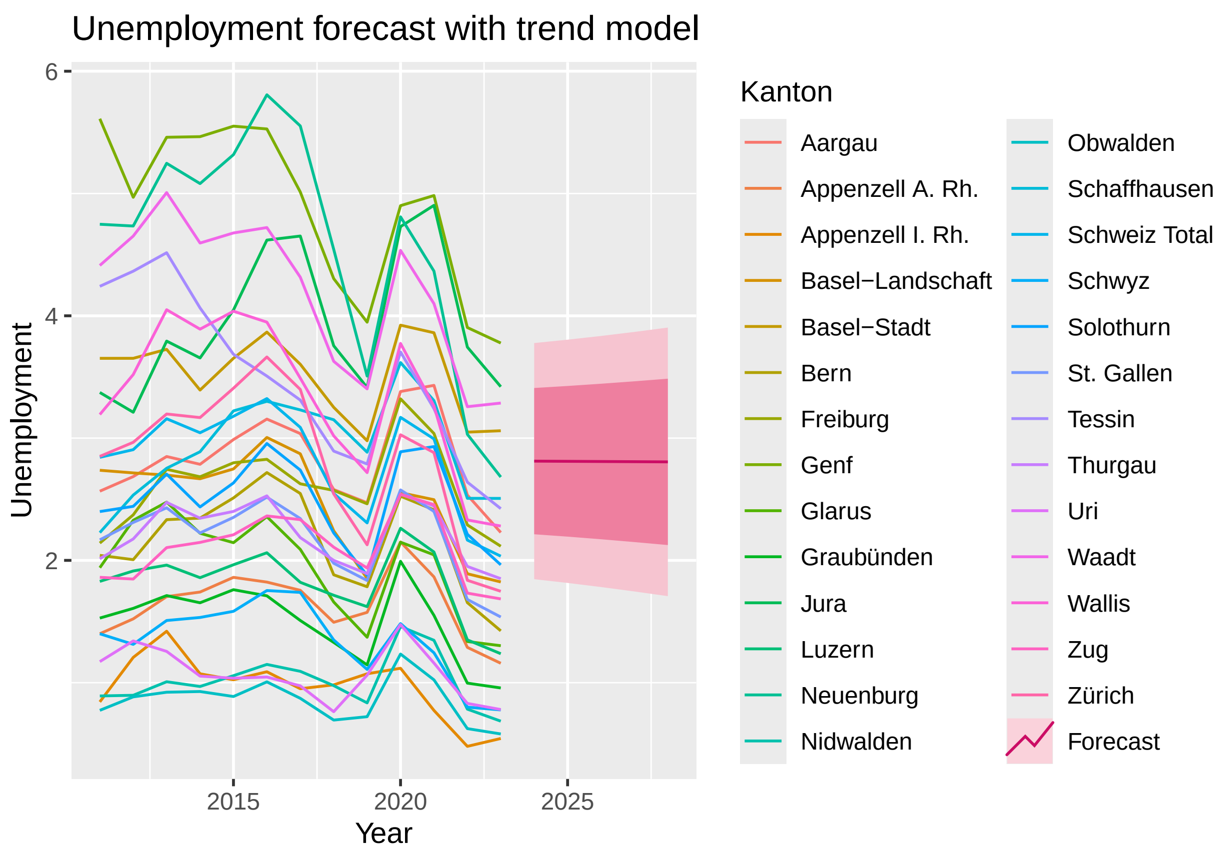



```r
fit_fc |>
    autoplot()
```



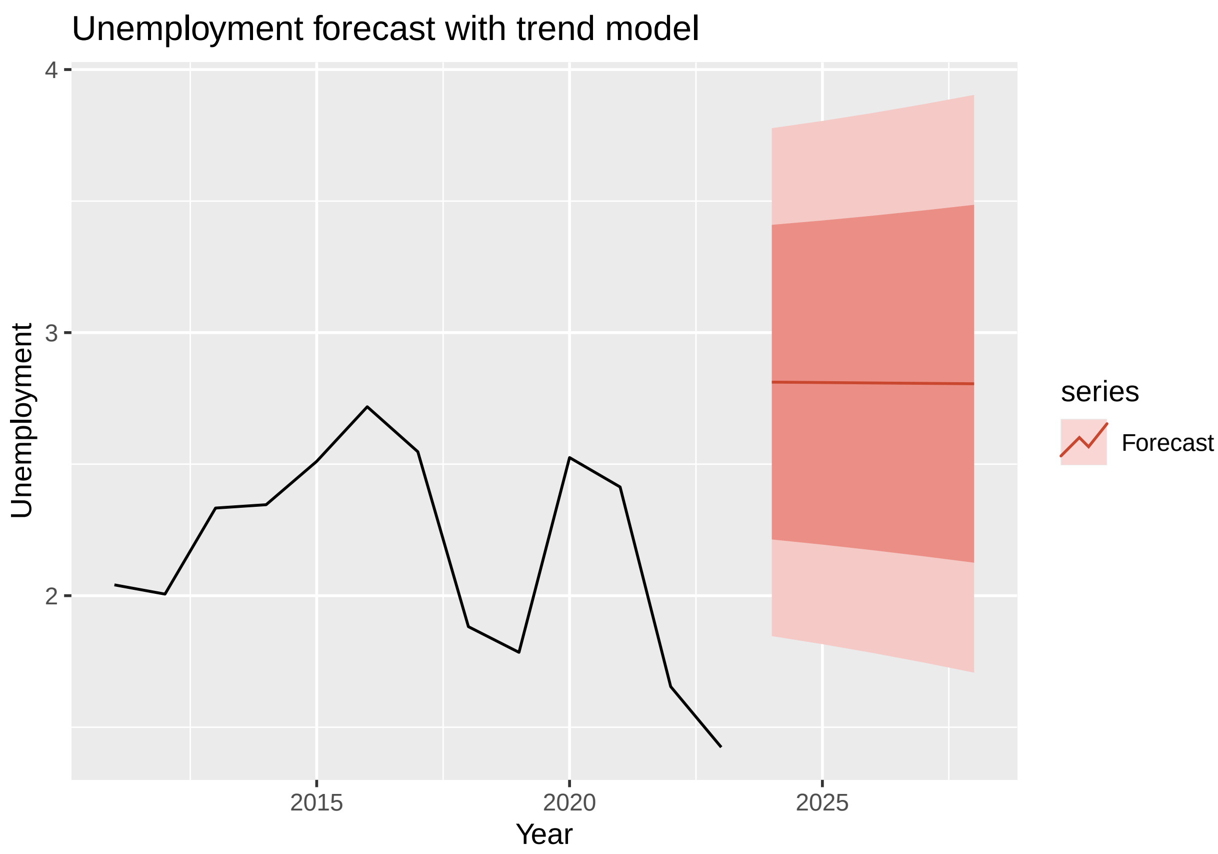

```r
autoplot(data_tsibble) +
    autolayer(fit_fc, series = "Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment forecast with trend model")
```

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




```
data_tsibble |>
    filter(Kanton == "Bern") |>
    autoplot() +
    autolayer(fit_fc, series = "Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment forecast with trend model")
```

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

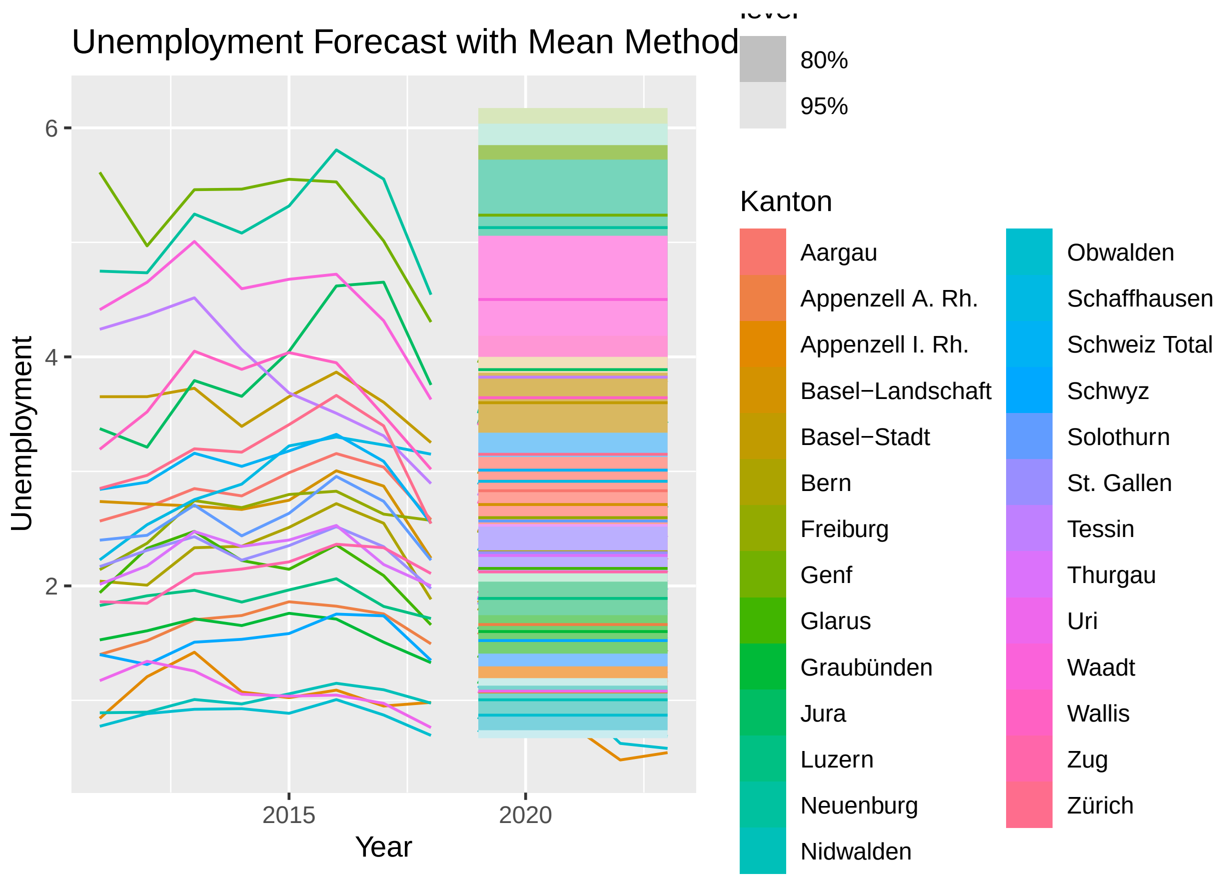


Next we try the mean method to get another forecast:

```
# Split the data into training and test sets
train_data <- data_tsibble |>
    filter(Year <= 2018)

test_data <- data_tsibble |>
    filter(Year > 2018)


# Mean Forecasting Method
mean_model <- train_data |>
    model(mean_fc = MEAN(Unemployment))

mean_fc <- mean_model |>
    forecast(new_data = test_data)

# Plot the forecast
autoplot(train_data, series = "Training Data") +
    autolayer(test_data, series = "Test Data") +
    autolayer(mean_fc, series = "Mean Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment Forecast with Mean Method")
```

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

```
## Warning in geom_line(...): Ignoring unknown parameters: ‘series‘

## Plot variable not specified, automatically selected ‘.vars = Unemployment‘

## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: ‘series‘

## Warning in ggdist::geom_lineribbon(without(intvl_mapping, "colour_ramp"), :
## Ignoring unknown parameters: ‘series‘

## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## ‘series‘
```

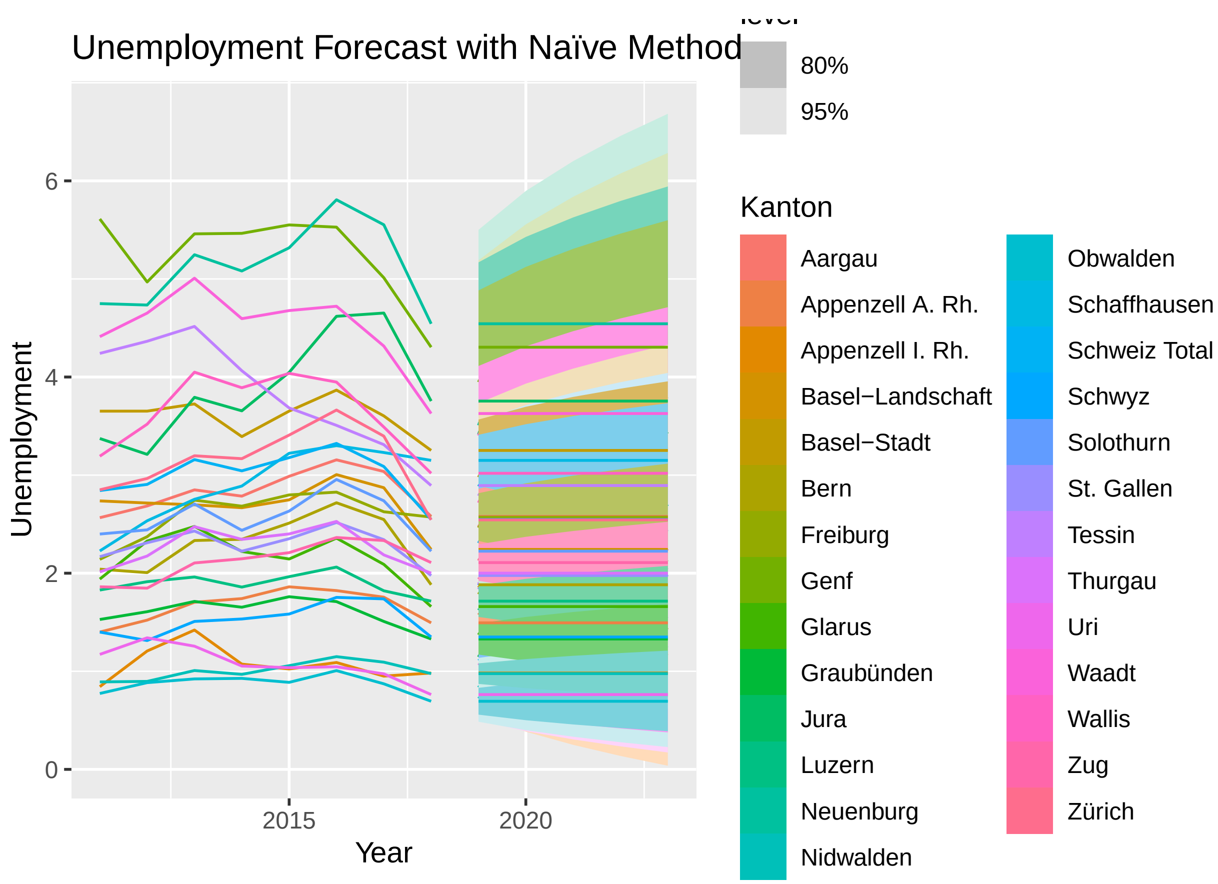


Naive method:

```r
# Naïve Forecasting Method
naive_model <- train_data |>
    model(naive_fc = NAIVE(Unemployment))

naive_fc <- naive_model |>
    forecast(new_data = test_data)

# Plot the forecast
autoplot(train_data, series = "Training Data") +
```

```
    autolayer(test_data, series = "Test Data") +
    autolayer(naive_fc, series = "Naïve Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment Forecast with Naïve Method")
```

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

## Warning in geom_line(...): Ignoring unknown parameters: `series`

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

## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`

## Warning in ggdist::geom_lineribbon(without(intvl_mapping, "colour_ramp"), :
## Ignoring unknown parameters: `series`

## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## `series`
```



Drift Method:

```r
# Drift Forecasting Method
drift_model <- train_data |>
    model(drift_fc = RW(Unemployment ~ drift()))

drift_fc <- drift_model |>
    forecast(new_data = test_data)

# Plot the forecast
autoplot(train_data, series = "Training Data") +
    autolayer(test_data, series = "Test Data") +
    autolayer(drift_fc, series = "Drift Forecast") +
    xlab("Year") +
    ylab("Unemployment") +
    ggtitle("Unemployment Forecast with Drift Method")
```

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

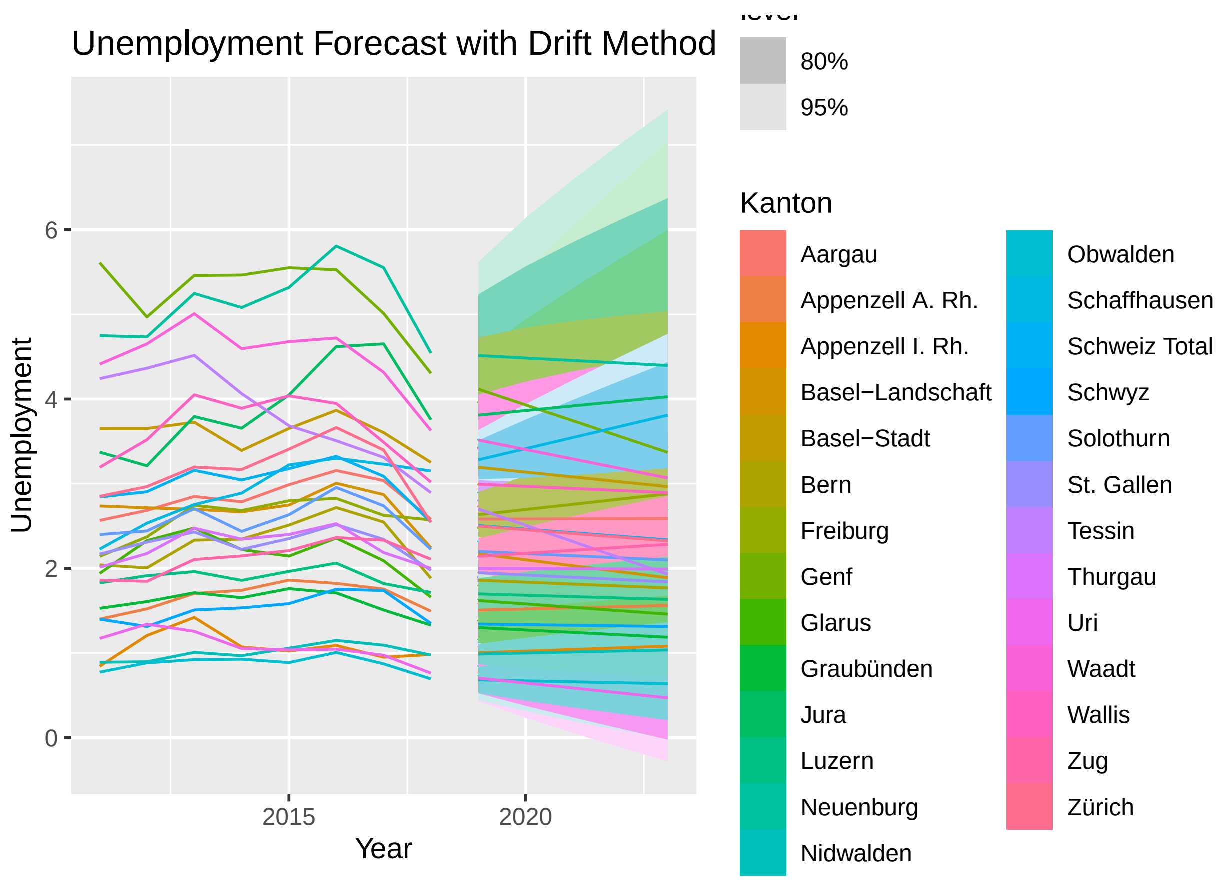
```
## Warning in geom_line(...): Ignoring unknown parameters: `series`
```

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

```
## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`
```

```
## Warning in ggdist::geom_lineribbon(without(intvl_mapping, "colour_ramp"), :
## Ignoring unknown parameters: `series`
```

```
## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## `series`
```



Just from looking at the plots, it is hard to tell which method is the best. We will now calculate the accuracy of the forecasts to get a better idea of which method is the best.

```
# Calculate accuracy metrics
mean_accuracy <- mean_fc |>
    accuracy(test_data)

naive_accuracy <- naive_fc |>
    accuracy(test_data)

drift_accuracy <- drift_fc |>
    accuracy(test_data)

mean_RMSE <- mean(mean_accuracy$RMSE)
naive_RMSE <- mean(naive_accuracy$RMSE)
drift_RMSE <- mean(drift_accuracy$RMSE)
cat("Mean RMSE: ", mean_RMSE, "\n")
```

```
## Mean RMSE:  0.6047311
```

```
cat("Naïve RMSE: ", naive_RMSE, "\n")
```

```
## Naïve RMSE:  0.4651364
```

```r
cat("Drift RMSE: ", drift_RMSE, "\n")
```

```
## Drift RMSE:  0.5048573
```

```r
cat("We can see that the Naïve method has the lowest RMSE, which means it is the most accurate method. 
```

```
## We can see that the Naïve method has the lowest RMSE, which means it is the most accurate method.
```

```r
mean(naive_accuracy$MAPE)
```

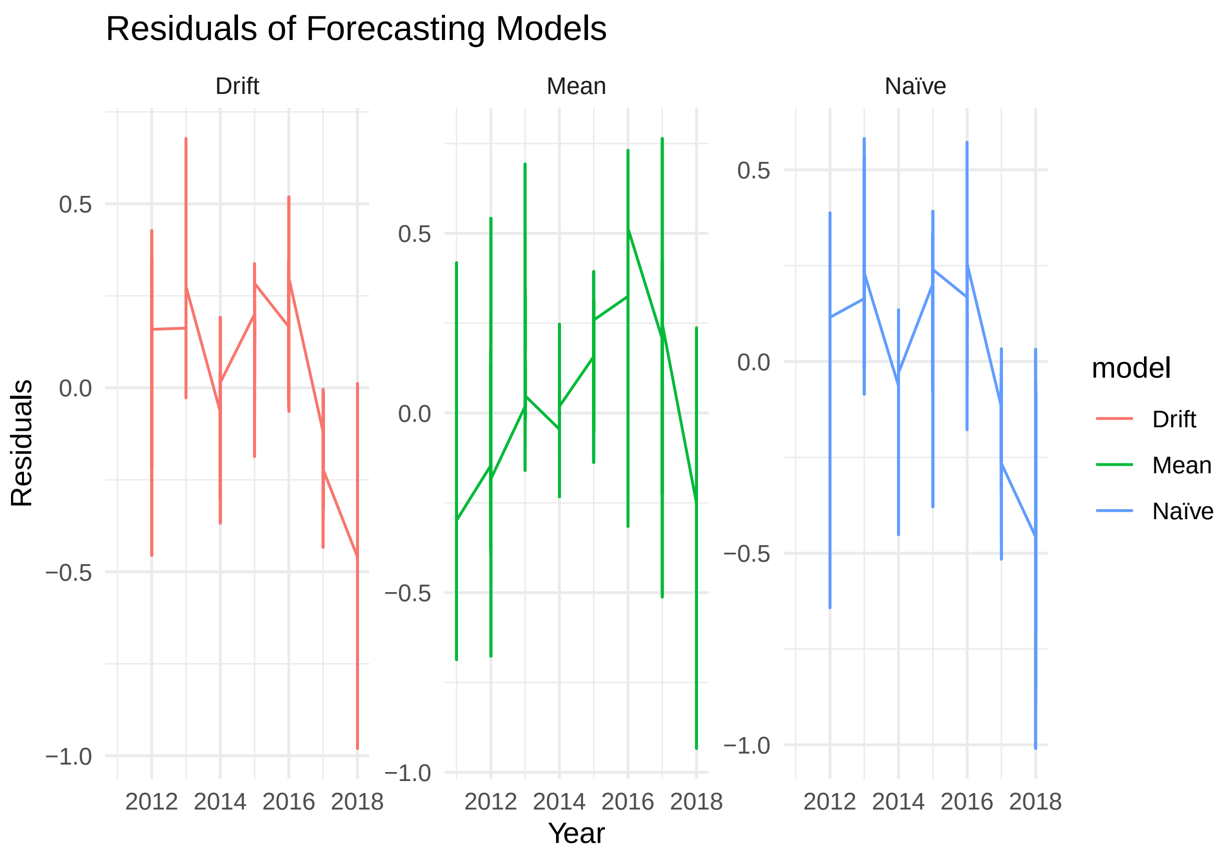
```
## [1] 20.45493
```

We can see that the Naïve method has the lowest RMSE, which means it is the most accurate method. It also has the lowest MAPE (20.45), which means it is the most accurate method so far.

To get a better idea of the accuracy we visualize the performance of the models by combining the residuals:

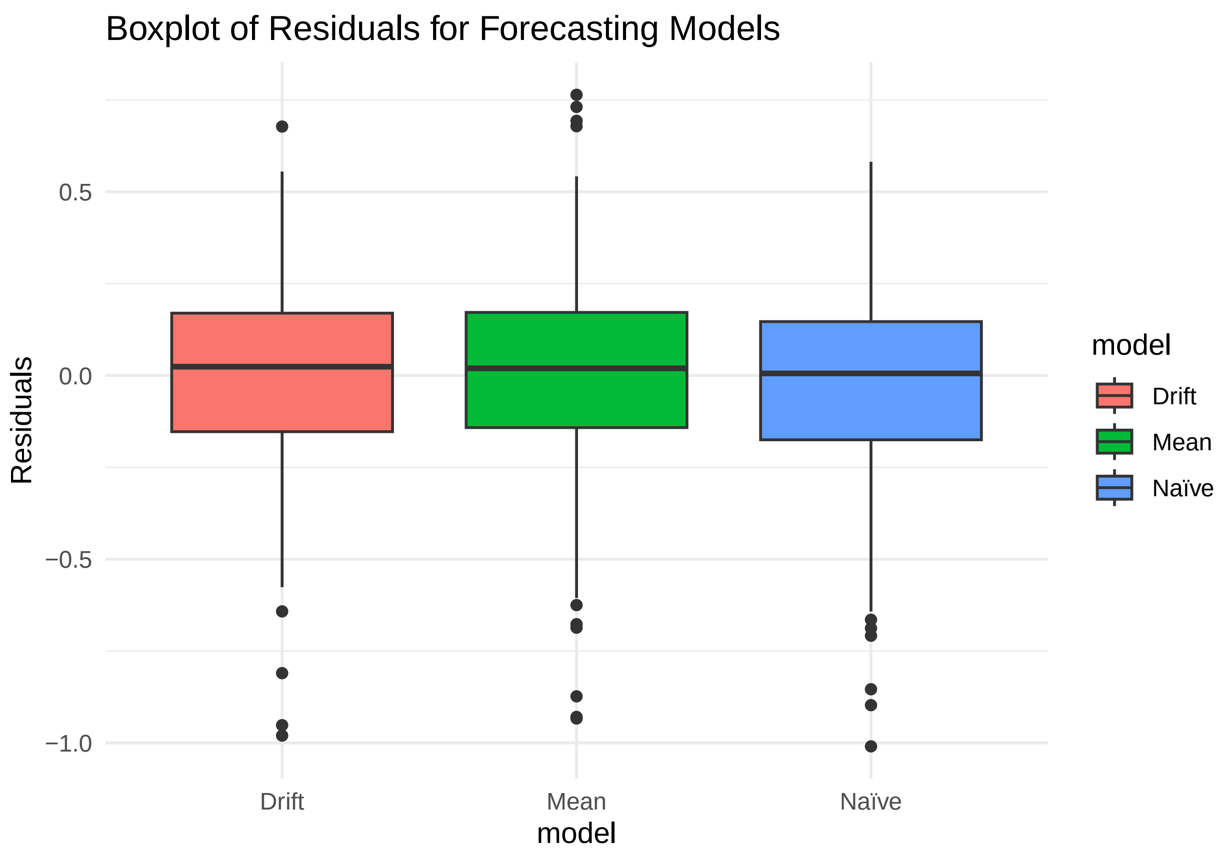
```r
# Residuals for each model
mean_resid <- augment(mean_model) |>
    mutate(model = "Mean")

naive_resid <- augment(naive_model) |>
    mutate(model = "Naïve")

drift_resid <- augment(drift_model) |>
    mutate(model = "Drift")

# Combine residuals
residuals <- bind_rows(mean_resid, naive_resid, drift_resid)

# Plot residuals
residuals |>
    ggplot(aes(x = Year, y = .resid, color = model)) +
    geom_line() +
    facet_wrap(~model, scales = "free_y") +
    theme_minimal() +
    labs(title = "Residuals of Forecasting Models", y = "Residuals")
```

```
## Warning: Removed 54 rows containing missing values or values outside the scale range
## (‘geom_line()‘).
```



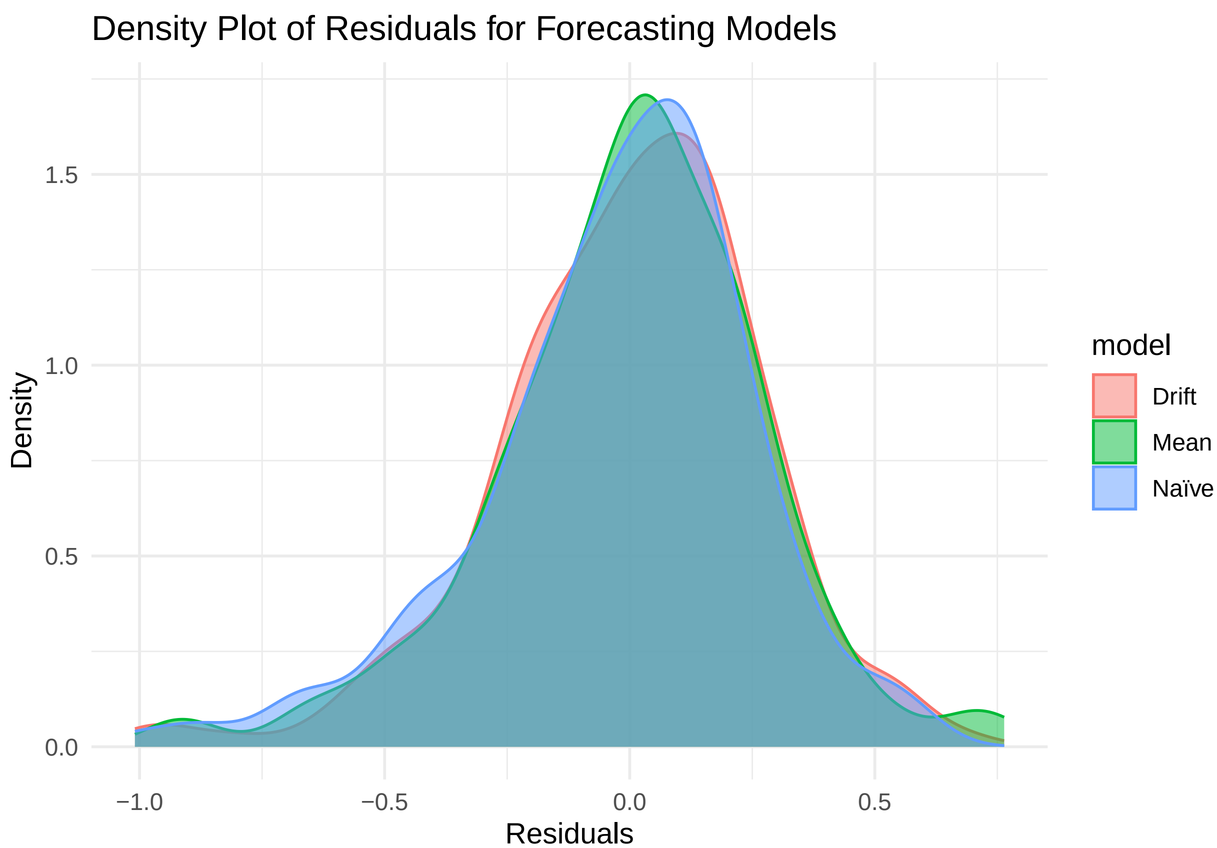

```r
# Boxplot of residuals
residuals |>
    ggplot(aes(x = model, y = .resid, fill = model)) +
    geom_boxplot() +
    theme_minimal() +
    labs(title = "Boxplot of Residuals for Forecasting Models", y = "Residuals")
```

```
## Warning: Removed 54 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
# Density plot of residuals
residuals |>
    ggplot(aes(x = .resid, fill = model, color = model)) +
    geom_density(alpha = 0.5) +
    theme_minimal() +
    labs(title = "Density Plot of Residuals for Forecasting Models", x = "Residuals", y = "Density")
```

```
## Warning: Removed 54 rows containing non-finite outside the scale range
## (`stat_density()`).
```



Especially the density plot shows that the residuals of the Naïve method are the closest to a normal distribution, which is a good sign for the accuracy of the model.

Actual Forecast with Naive Method

```
# Extend the forecast horizon to December 2024
last_year <- max(data$Year)
forecast_horizon <- 2024 - last_year

# Generate extended Naïve forecasts for each canton
extended_naive_fc <- data |>
    group_by(Kanton) |>
    model(naive_fc = NAIVE(Unemployment)) |>
    forecast(h = forecast_horizon)

# Extract the forecasted values
forecast_table <- extended_naive_fc |>
    as_tibble() |>
    select(Year, Kanton, .mean)

forecast_table
```

```
## # A tibble: 27 x 3
##     Year Kanton           .mean
```

```
##    <dbl> <chr>             <dbl>
##  1  2024 Aargau            2.23
##  2  2024 Appenzell A. Rh.  1.16
##  3  2024 Appenzell I. Rh.  0.544
##  4  2024 Basel-Landschaft  1.82
##  5  2024 Basel-Stadt       3.06
##  6  2024 Bern              1.42
##  7  2024 Freiburg          2.12
##  8  2024 Genf              3.78
##  9  2024 Glarus            1.30
## 10  2024 Graubünden        0.957
## # i 17 more rows
```

```
# TODO better display the values
```