Varicella Prediction Notebook

Code

Introduction

This is an education case where we're trying to predict the number of monthly varicella cases based on some time series forecasting methods. We will be comparing Holt-Winters, SARIMA and Random Forest models.

Libraries used

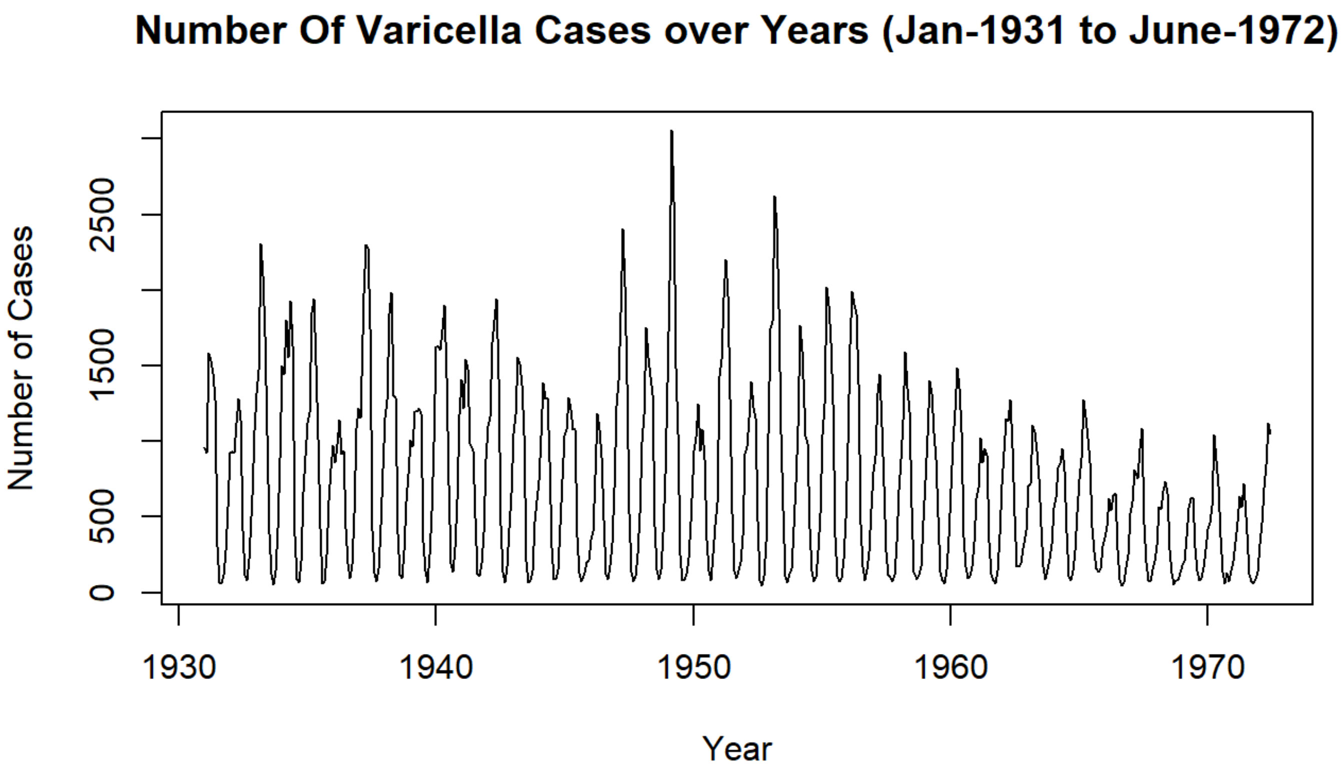
Hide

```
library(readr)
library(VSURF)
library(forecast)
library(randomForest)
```

Data preview

Hide

```
varicelle <- readr::read_csv("varicelle.csv", col_types = cols(x = col_number()))
varicelle_ts <- stats::ts(
  varicelle$x, start = c(1931, 1), end = c(1972, 6), frequency = 12)
graphics::plot(
  varicelle_ts,
  main = "Number Of Varicella Cases over Years (Jan-1931 to June-1972)",
  xlab = "Year",
  ylab = "Number of Cases"
)
```

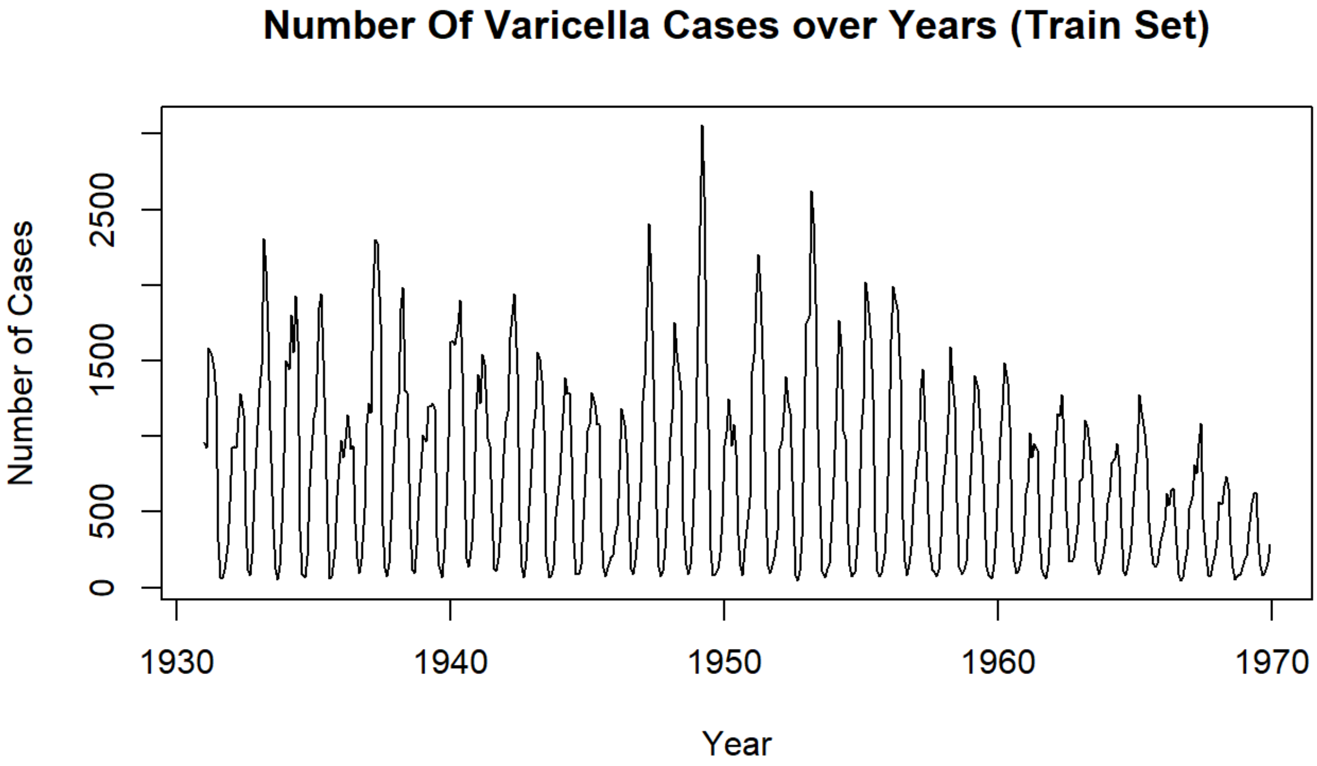


There seems to be a seasonality in the time series, but no trend is present : if there's one, it's hidden by the seasonality.

Specify train and test sets

Hide

```
varicelle_train_ts <- stats::window(varicelle_ts, start=c(1931, 1), end=c(1969, 12))
varicelle_test_ts <- stats::window(varicelle_ts, start=c(1970, 1))
h <- base::length(varicelle_test_ts)

graphics::plot(
  varicelle_train_ts,
  main = "Number Of Varicella Cases over Years (Train Set)",
  xlab = "Year",
  ylab = "Number of Cases"
)
```



As graphically shown, there's some seasonality present in data.

As first models proposed for this time series, we consider additive seasonal Holt-Winters ones. We will also consider an eventual boxcox transformation on the time-series, and damped versions : we will choose the best one of them (damped, no damped) based on errors produced on the test set.

Additive Seasonal Holt-Winters models (Trend + Seasonality)

Hide

```
hw_damped_model <- forecast::hw(
  varicelle_train_ts,
  seasonal = "additive",
  damped = TRUE,
  level = c(80, 95),
  alpha = NULL,
  beta = NULL,
  gamma = NULL,
  phi = NULL,
  lambda = NULL,
  h = h
)

hw_no_damped_model <- forecast::hw(
  varicelle_train_ts,
  seasonal = "additive",
  damped = FALSE,
  level = c(80, 95),
  alpha = NULL,
  beta = NULL,
```

```
  gamma = NULL,
  phi = NULL,
  lambda = NULL,
  h = h
)

actual <- base::as.numeric(varicelle_test_ts)

# Predicitons on test set
pred_damped <- hw_damped_model$mean
pred_no_damped <- hw_no_damped_model$mean

mae_damped <- base::mean(base::abs(pred_damped - actual))
rmse_damped <- base::sqrt(base::mean((pred_damped - actual)^2))
mape_damped <- base::mean(base::abs((pred_damped - actual)/actual)) * 100

mae_no_damped <- base::mean(base::abs(pred_no_damped - actual))
rmse_no_damped <- base::sqrt(base::mean((pred_no_damped - actual)^2))
mape_no_damped <- base::mean(base::abs((pred_no_damped - actual)/actual)) * 100

# Print results
base::cat("Damped: MAE =", mae_damped, ", RMSE =", rmse_damped, ", MAPE =", mape_dampe
d, "%\n")
```

```
Damped: MAE = 279.2936 , RMSE = 318.5941 , MAPE = 122.8611 %
```

Hide

```
base::cat("No damped: MAE =", mae_no_damped, ", RMSE =", rmse_no_damped, ", MAPE =", ma
pe_no_damped, "%\n")
```

```
No damped: MAE = 387.0642 , RMSE = 432.2144 , MAPE = 176.7857 %
```

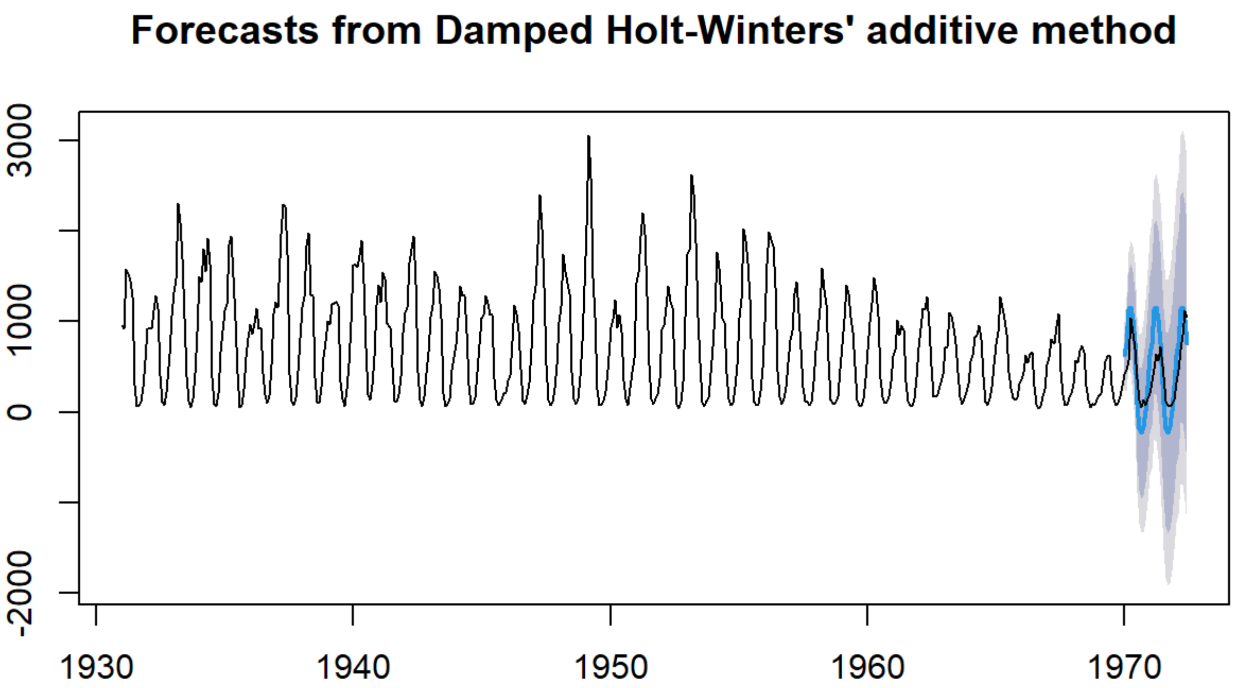
The damped version is showing the best results, so we will consider it as our base model.

Hide

```
base::rm(
  pred_damped, pred_no_damped, actual,
  mae_no_damped, rmse_no_damped, mape_no_damped, hw_no_damped_model
)
```

Hide

```
graphics::plot(hw_damped_model)
graphics::lines(varicelle_ts)
```



SARIMA models (Trend + Seasonality + Stochasticity)

We compute a base SARIMA model using auto.arima() then we'll compare it to our own model.

Hide

```
auto_sarima_model <- forecast::auto.arima(varicelle_train_ts)
auto_sarima_model
```

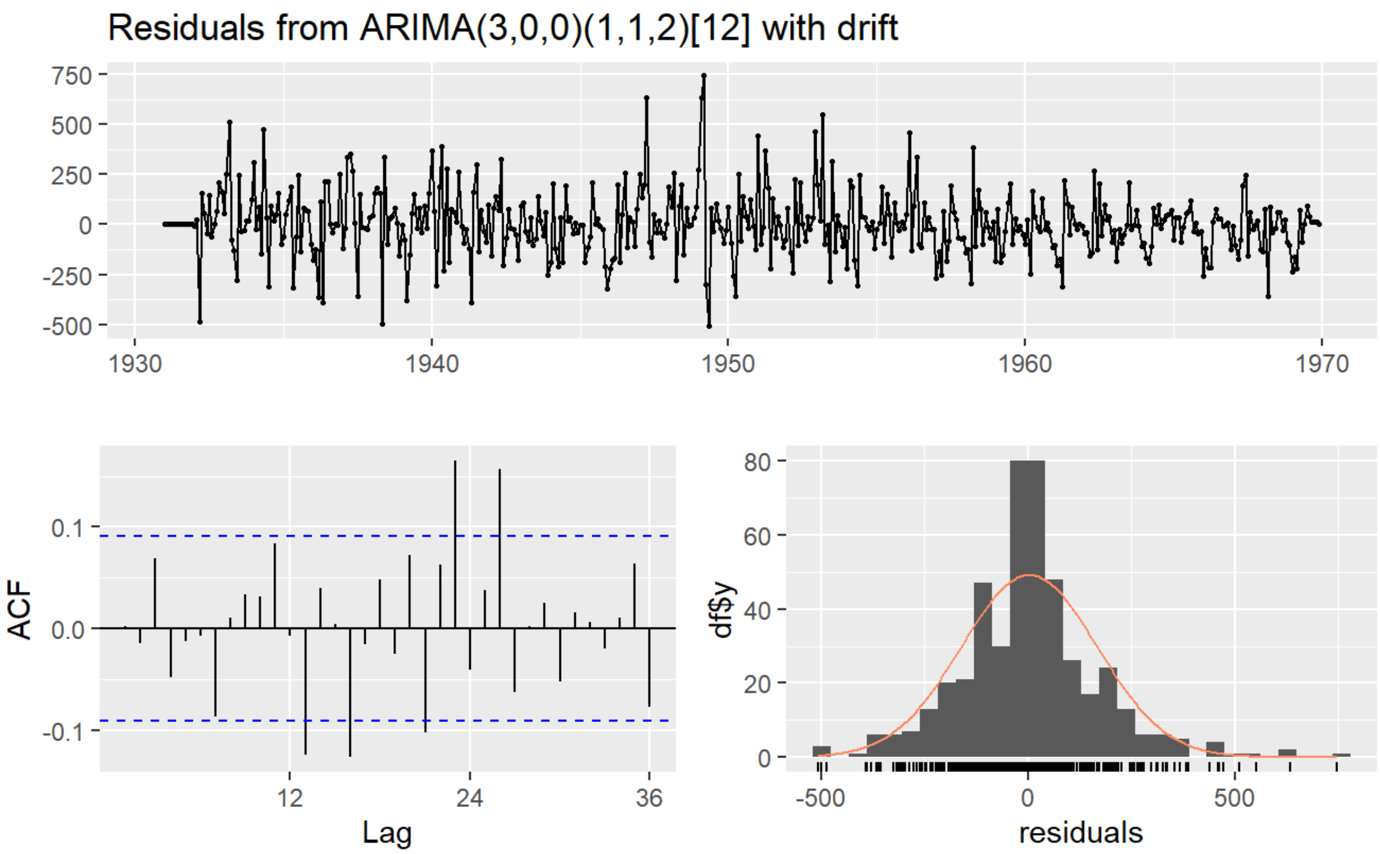
```
Series: varicelle_train_ts
ARIMA(3,0,0)(1,1,2)[12] with drift

Coefficients:
         ar1      ar2     ar3     sar1     sma1     sma2    drift
      0.9480  -0.2363  0.0142  -0.8048  -0.0989  -0.5797  -1.0326
s.e.  0.0491   0.0639  0.0476   0.1853   0.2128   0.1918   0.4791

sigma^2 = 28233:  log likelihood = -2987.92
AIC=5991.84   AICc=5992.16   BIC=6024.82
```

Hide

```
forecast::checkresiduals(auto_sarima_model)
```

```

	Ljung-Box test

data:  Residuals from ARIMA(3,0,0)(1,1,2)[12] with drift
Q* = 53.235, df = 18, p-value = 2.413e-05
```

```
Model df: 6.   Total lags used: 24
```

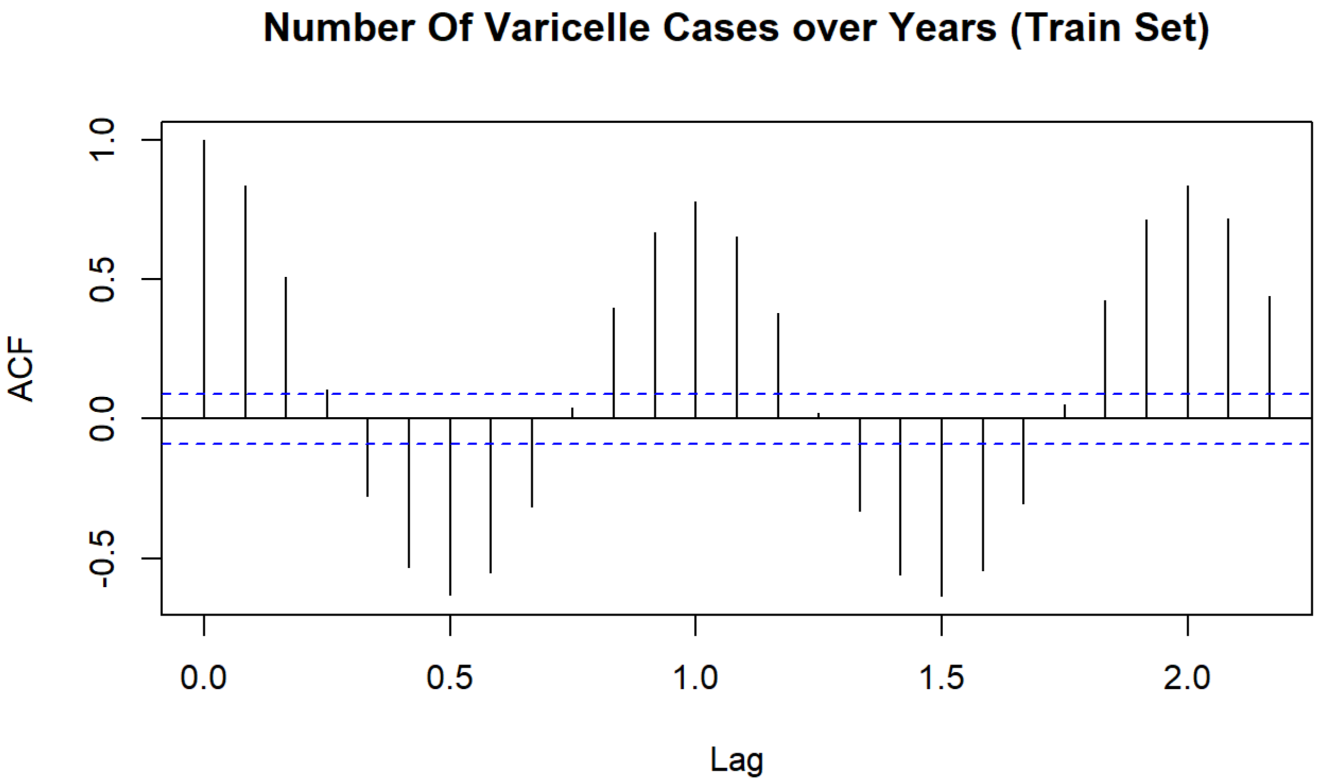


The model proposed by auto.arima() has significant auto-correlations present in its residuals : this violated one of the base hypothesis of SARIMA models.

We try now to compute a better version (optimized) of SARIMA model using our own knowledge of the studied time series.

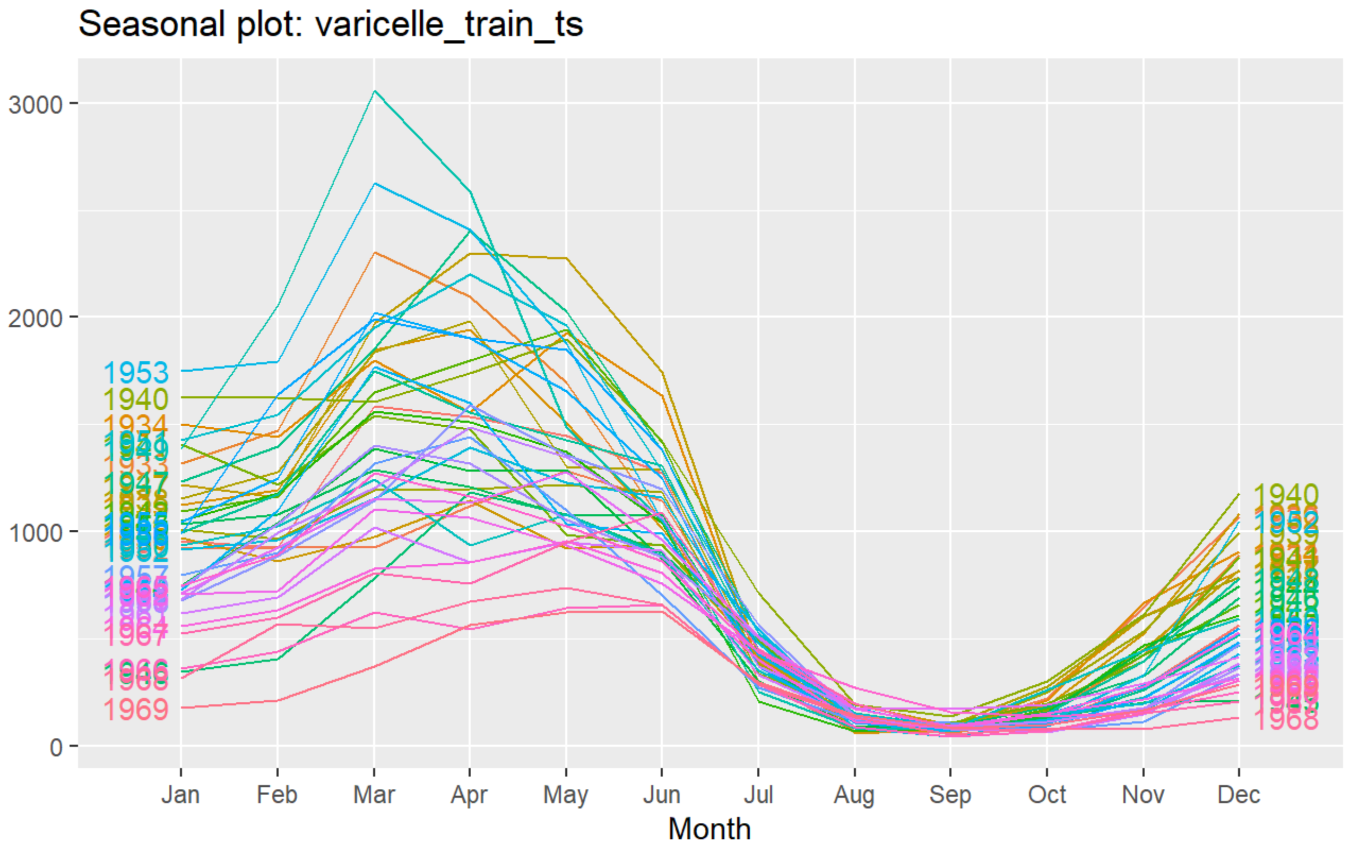
Hide

```
# Check for any presence of seasonality
stats::acf(
  varicelle_train_ts,
  type = "cor", main = "Number Of Varicelle Cases over Years (Train Set)"
)
```



Hide

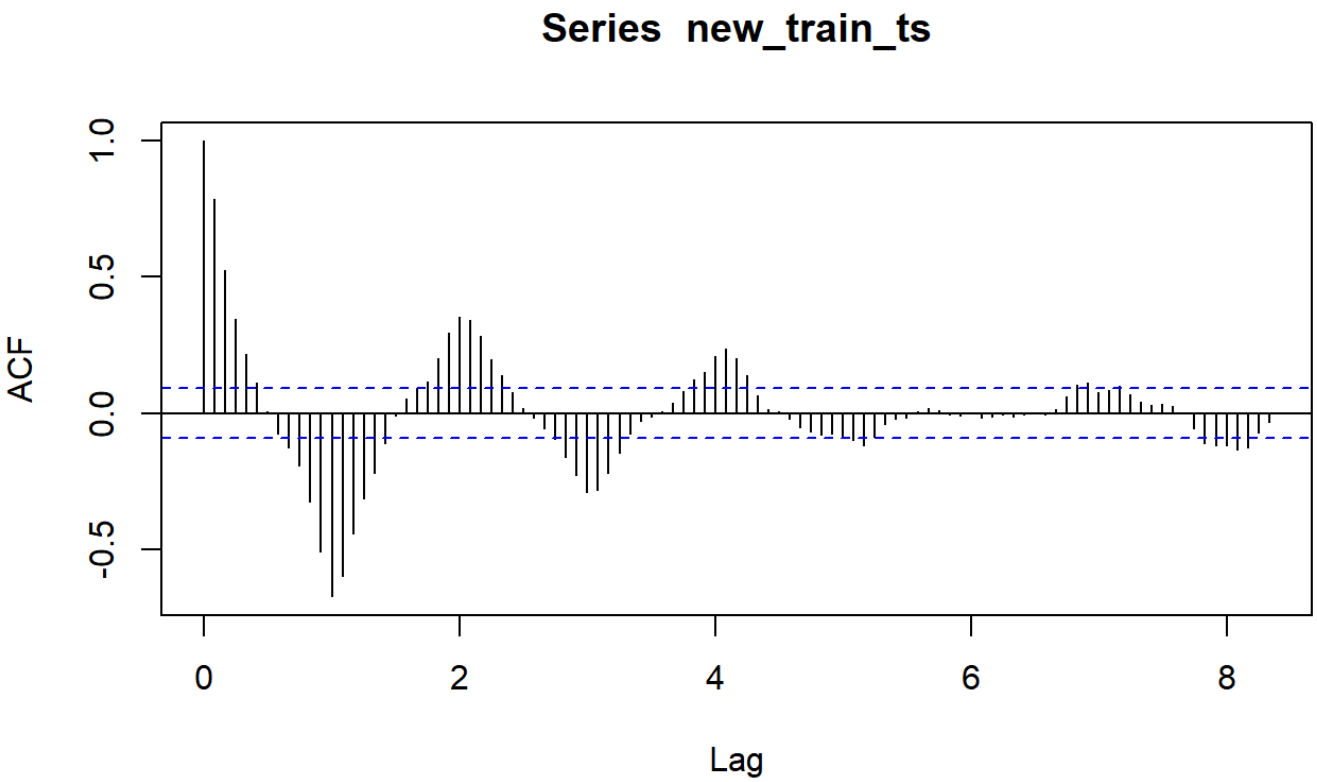
```
forecast::ggseasonplot(varicelle_train_ts, year.labels = TRUE, year.labels.left = TRUE)
```



Both plots show a presence of seasonality of period at least equal to 12. Let's differentiate at least one time to remove the seasonality.

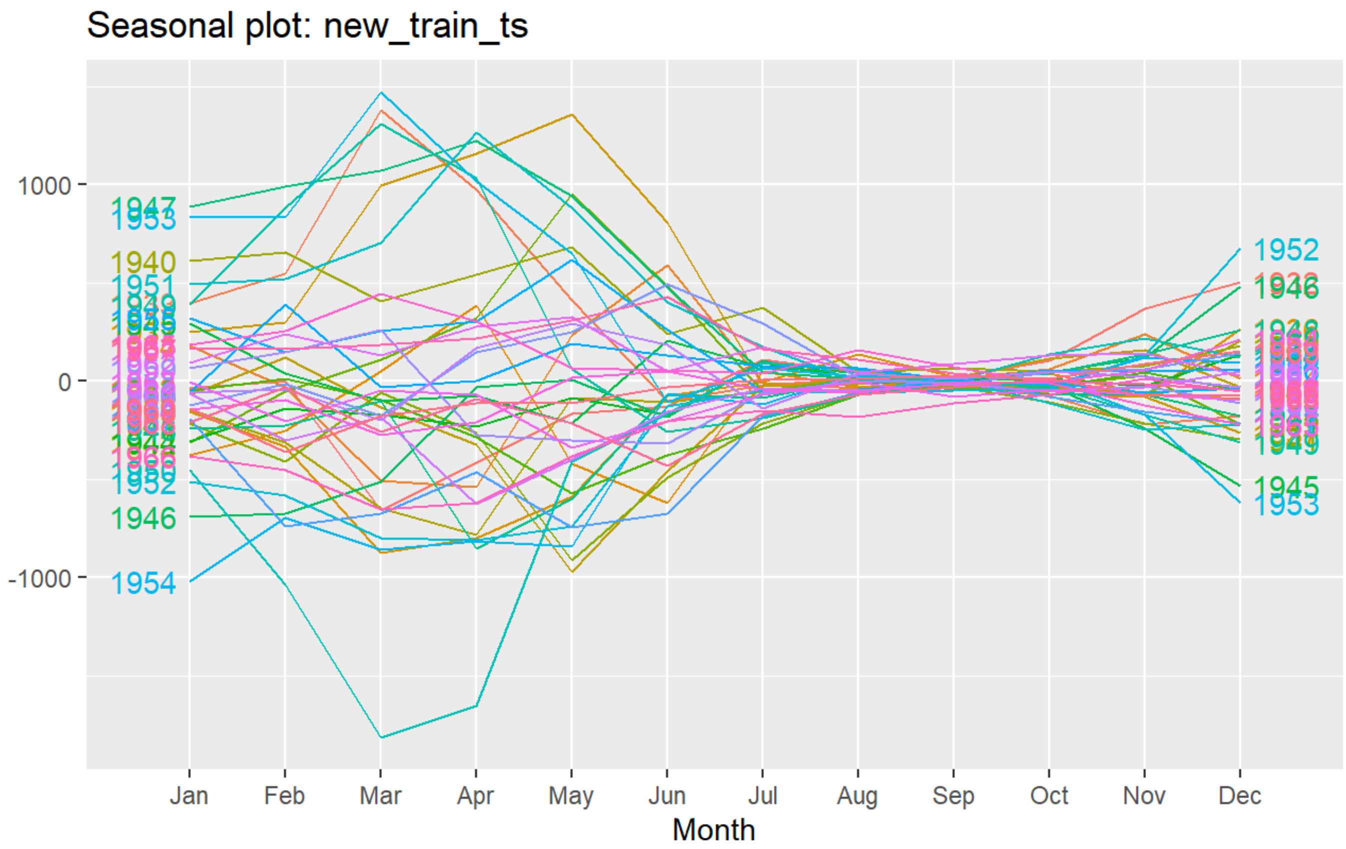
Hide

```
new_train_ts <- base::diff(varicelle_train_ts, lag = 12, differences = 1)

stats::acf(new_train_ts, type = "cor", lag.max = 100)
```



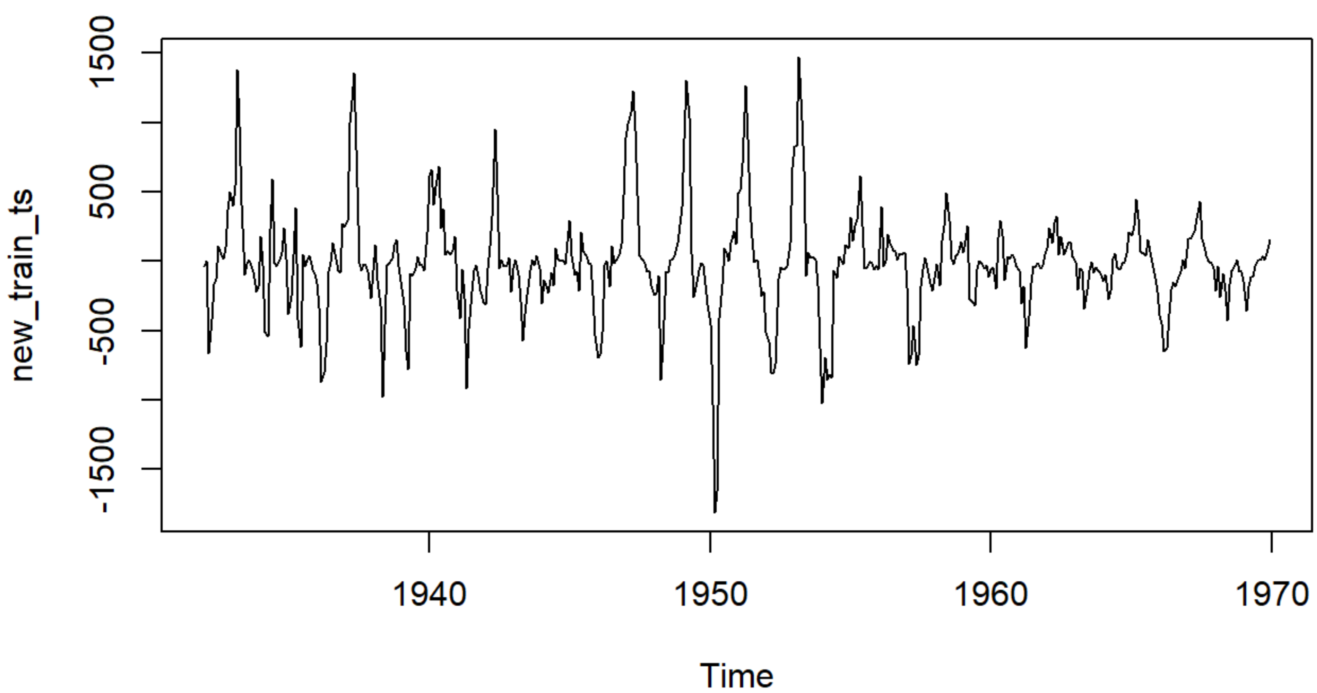
Hide

```
forecast::ggseasonplot(new_train_ts, year.labels = TRUE, year.labels.left = TRUE)
```



Hide

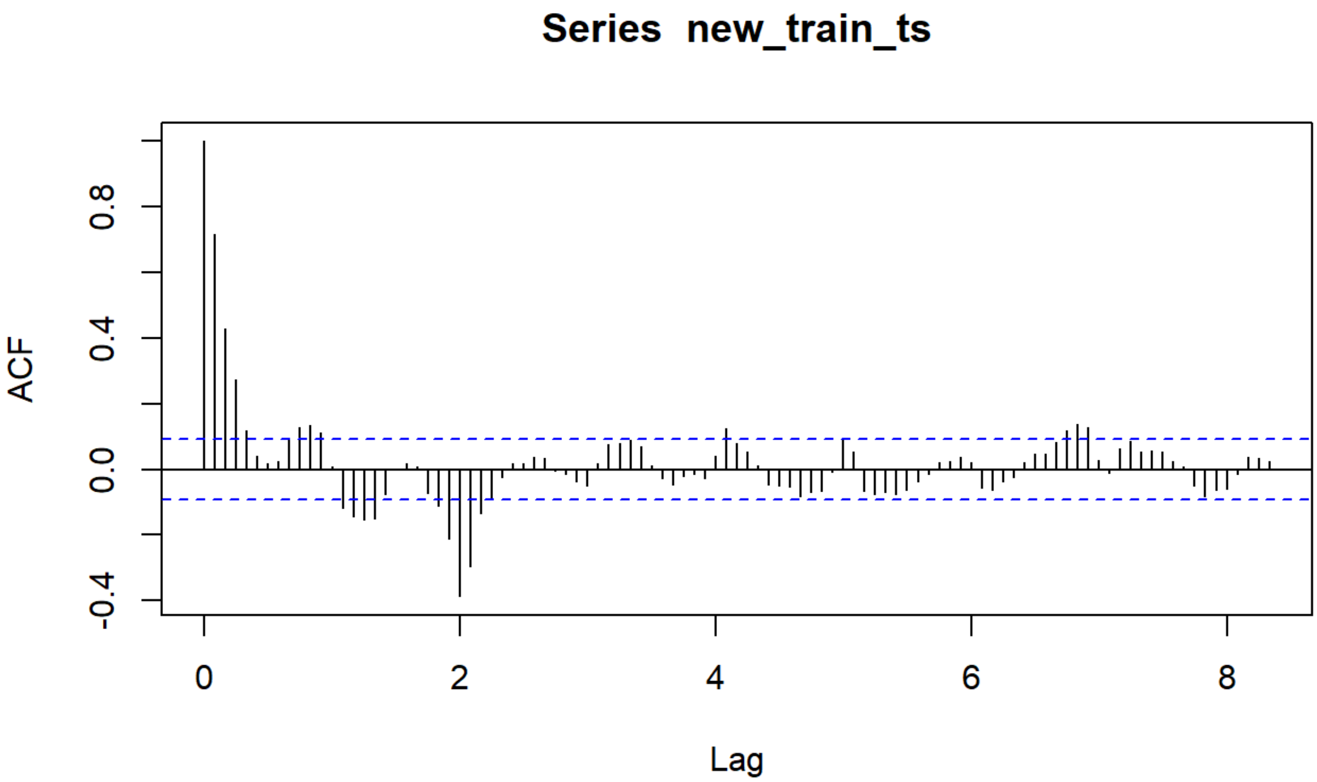
```
graphics::plot(new_train_ts)
```



Let's work on the lag value as much as needed to remove any presence of seasonality.

Hide

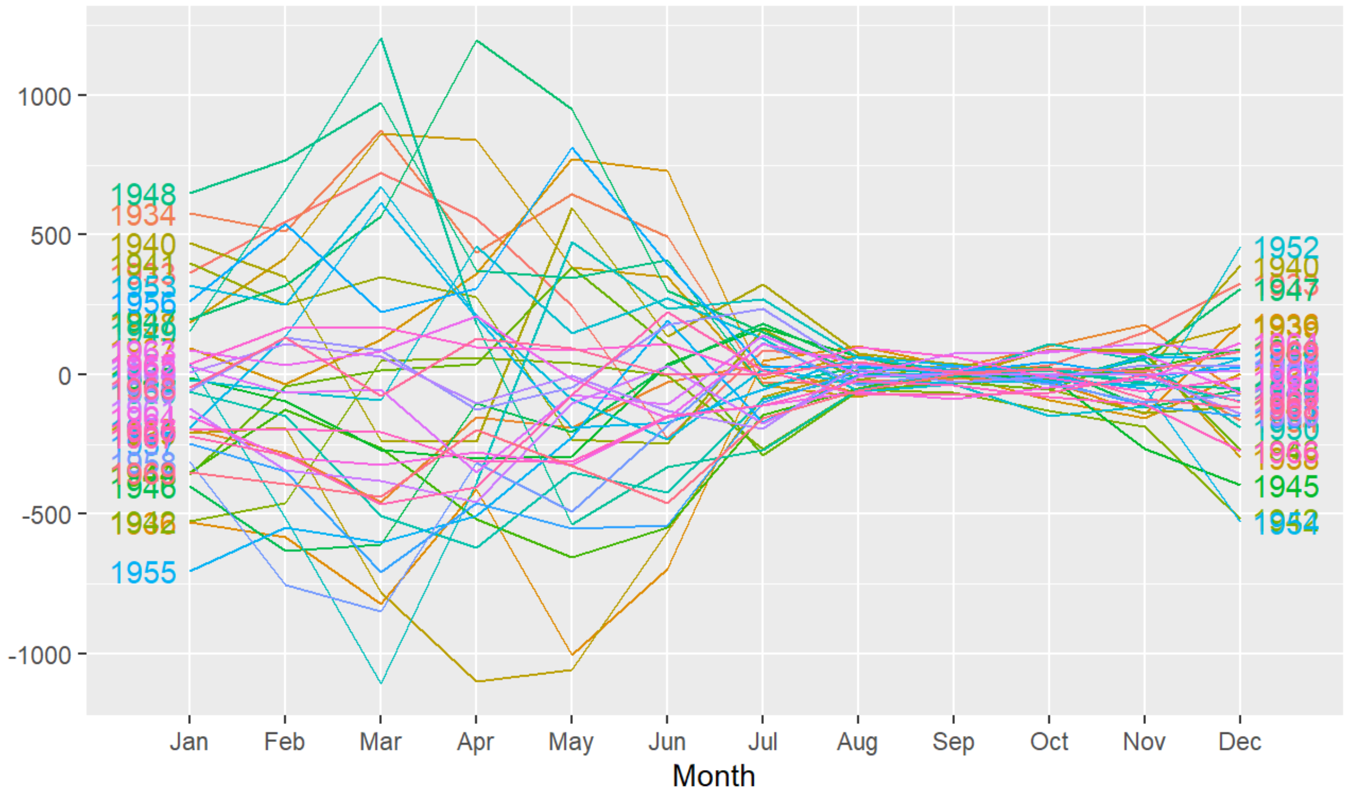
```
new_train_ts <- base::diff(varicelle_train_ts, lag = 24, differences = 1)

stats::acf(new_train_ts, type = "cor", lag.max = 100)
```



Hide

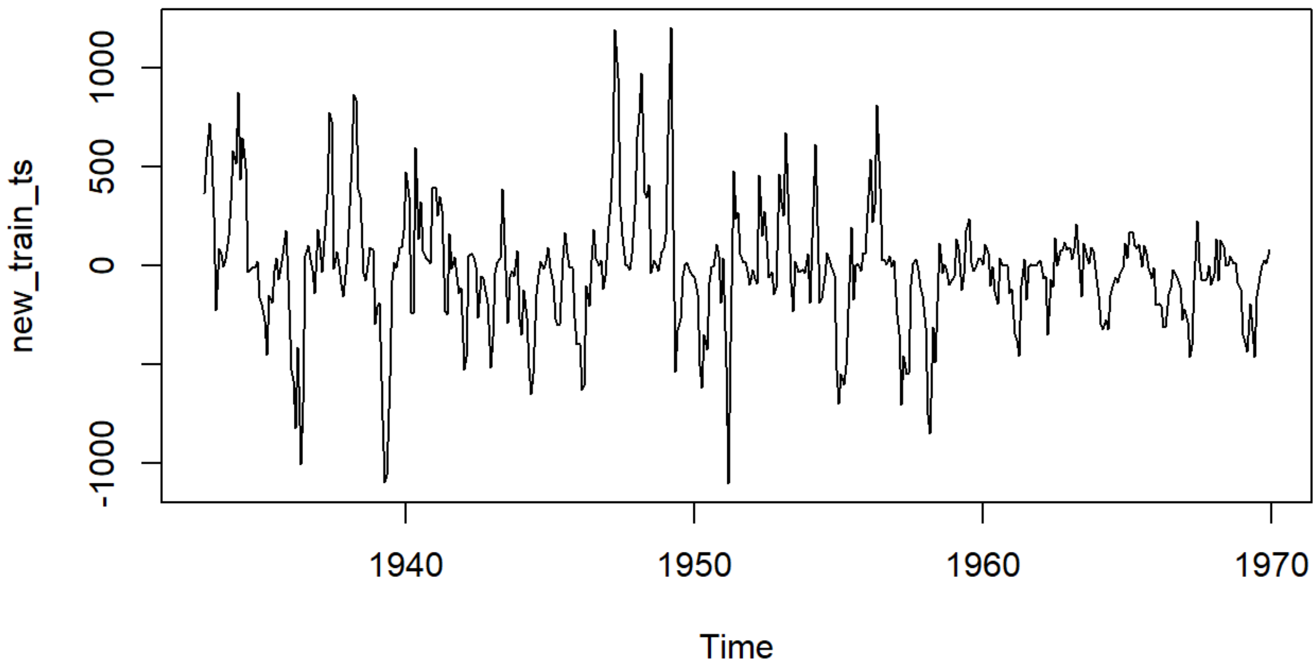
```
forecast::ggseasonplot(new_train_ts, year.labels = TRUE, year.labels.left = TRUE)
```

Seasonal plot: new_train_ts



Hide

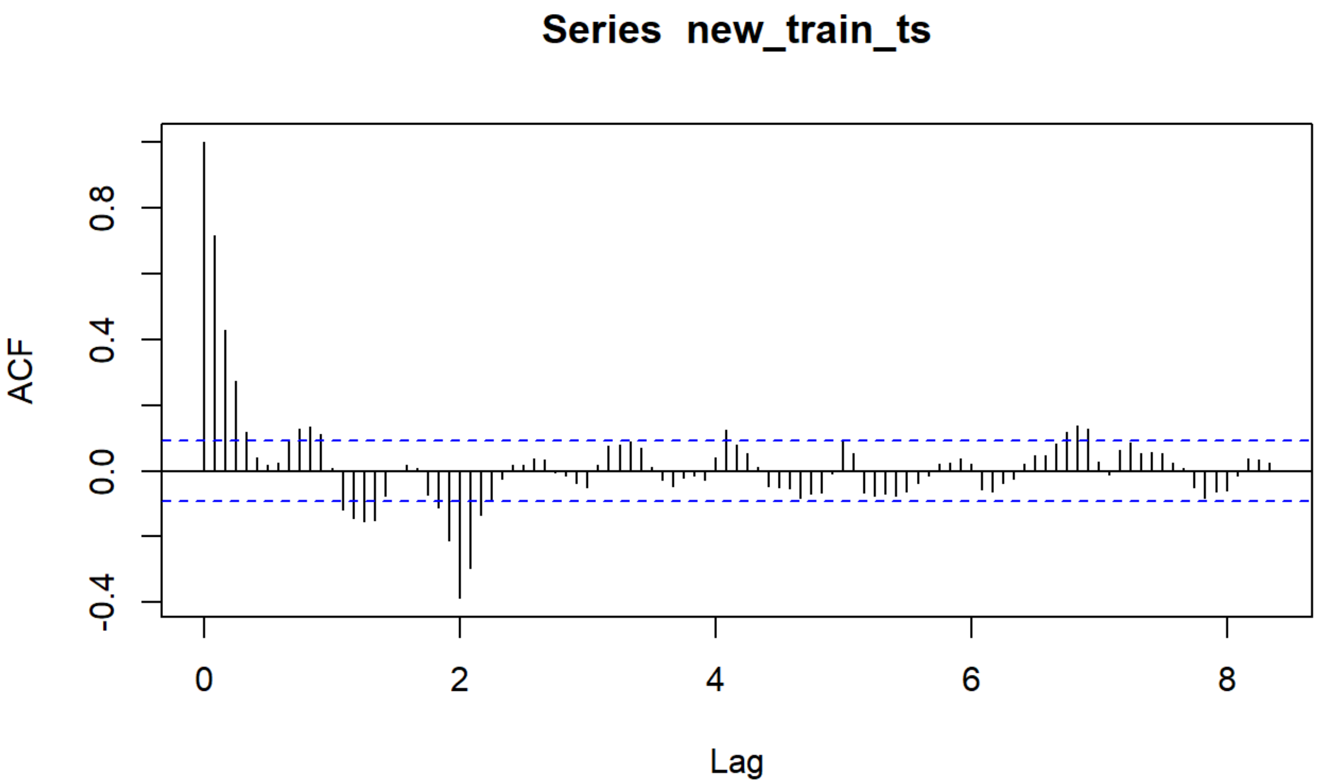
```
graphics::plot(new_train_ts)
```



No strong seasonality still seems to be present anymore. We also detect no special trend. We assume the new time series is stationary. We'll use ARMA models for this part.

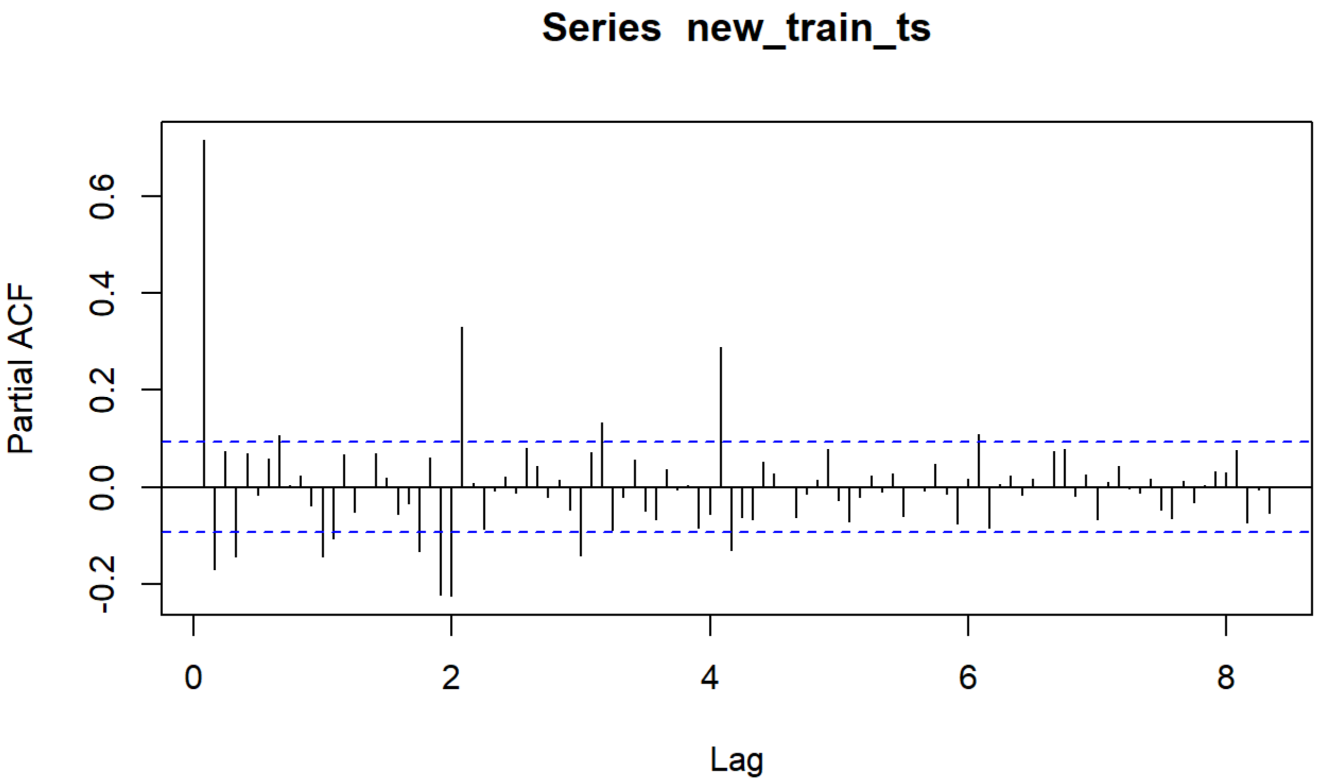
Hide

```
stats::acf(new_train_ts, type = "cor", lag.max = 100)
```



Hide

```
stats::pacf(new_train_ts, lag.max = 100)
```



Only the ACF is showing some decrease towards 0: this infers the use of an AR model . Based on pacf, the order can be up to an AR(48) or SAR(4). Let’s write our final SARIMA model max parameters that we will pass to the auto.arima function which will choose the best parameters based on AIC.

Hide

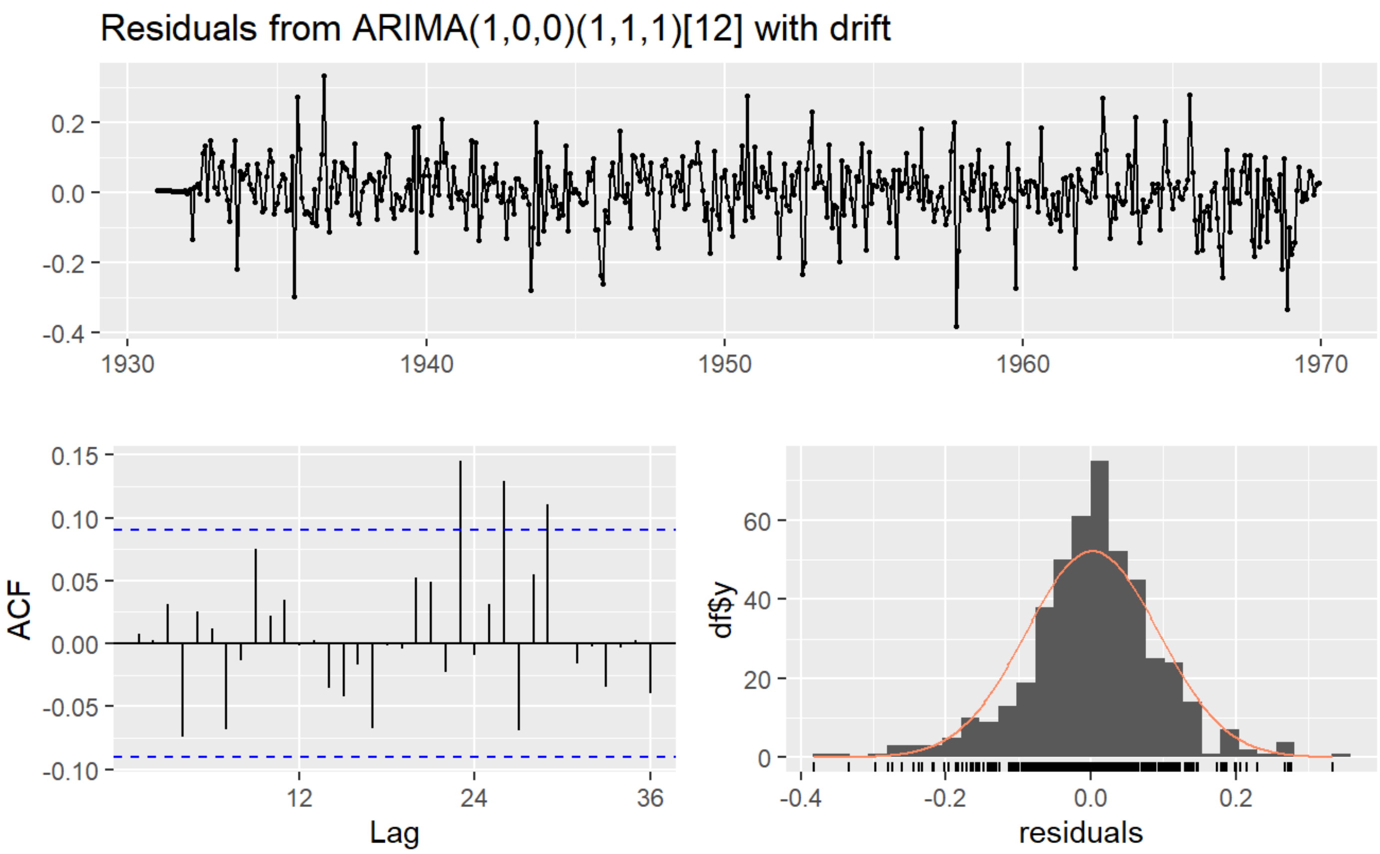
```
sarima_model <- forecast::auto.arima(
  varicelle_train_ts,
  # (add +1 to values we've chosen using acf and pacf previously)
  max.p = 1, # 0 + 1
  max.q = 1, # 0 + 1
  max.P = 5, # 4 + 1
  max.Q = 1, # 0 + 1
  max.d = 1, # 0 + 1
  max.D = 2, # 1 + 1
  seasonal = TRUE,
  stepwise = FALSE, # thorough grid search
  approximation = FALSE, # use exact likelihood
  lambda = "auto", # Box-Cox transformation
  # trace = TRUE, # optional: see progress
  method = "ML", # maximum likelihood
  allowdrift = TRUE, # allow drift if needed
  allowmean = TRUE, # allow mean if needed
  ic = "aic", # use AIC for model selection
  seasonal.test = "seas", # default seasonal test
)

sarima_model
```

```
Series: varicelle_train_ts
ARIMA(1,0,0)(1,1,1)[12] with drift
Box Cox transformation: lambda= -0.1562783

Coefficients:
         ar1     sar1     sma1   drift
      0.6815  -0.1058  -0.7583  -5e-04
s.e.  0.0345   0.0587   0.0402   3e-04

sigma^2 = 0.008684:  log likelihood = 430.71
AIC=-851.43   AICc=-851.3   BIC=-830.82
```

Hide

```
forecast::checkresiduals(sarima_model)
```

```

	Ljung-Box test

data:  Residuals from ARIMA(1,0,0)(1,1,1)[12] with drift
Q* = 26.434, df = 21, p-value = 0.1904

Model df: 3.   Total lags used: 24
```



A whole view on residuals ACF graph shows the presence of auto-correlations at order 23, 26 and 29; but using the default lag provided in the checkresiduals() function we can accept that no significant autocorrelation is present.

Let’s analyze coefficients of our model :

Hide

```
base::summary(sarima_model)
```

```
Series: varicelle_train_ts
ARIMA(1,0,0)(1,1,1)[12] with drift
Box Cox transformation: lambda= -0.1562783

Coefficients:
         ar1     sar1     sma1   drift
      0.6815  -0.1058  -0.7583  -5e-04
s.e.  0.0345   0.0587   0.0402   3e-04

sigma^2 = 0.008684:  log likelihood = 430.71
AIC=-851.43   AICc=-851.3   BIC=-830.82

Training set error measures:
                   ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set 23.02784 156.2877 102.1339 -1.483727 16.89809 0.4202471 0.2431746
```

SAR has a coefficient in absolute value less than twice its standard deviation so we decide to remove the SAR part in the model.

Hide

```
sarima_model <- forecast::Arima(
  varicelle_train_ts,
  order = c(1, 0, 0), # (p, d, q)
  seasonal = c(0, 1, 1), # (P, D, Q)
  lambda = "auto",
  method = "ML",
  include.mean = TRUE,
  include.drift = TRUE,
  include.constant = TRUE
)

base::summary(sarima_model)
```

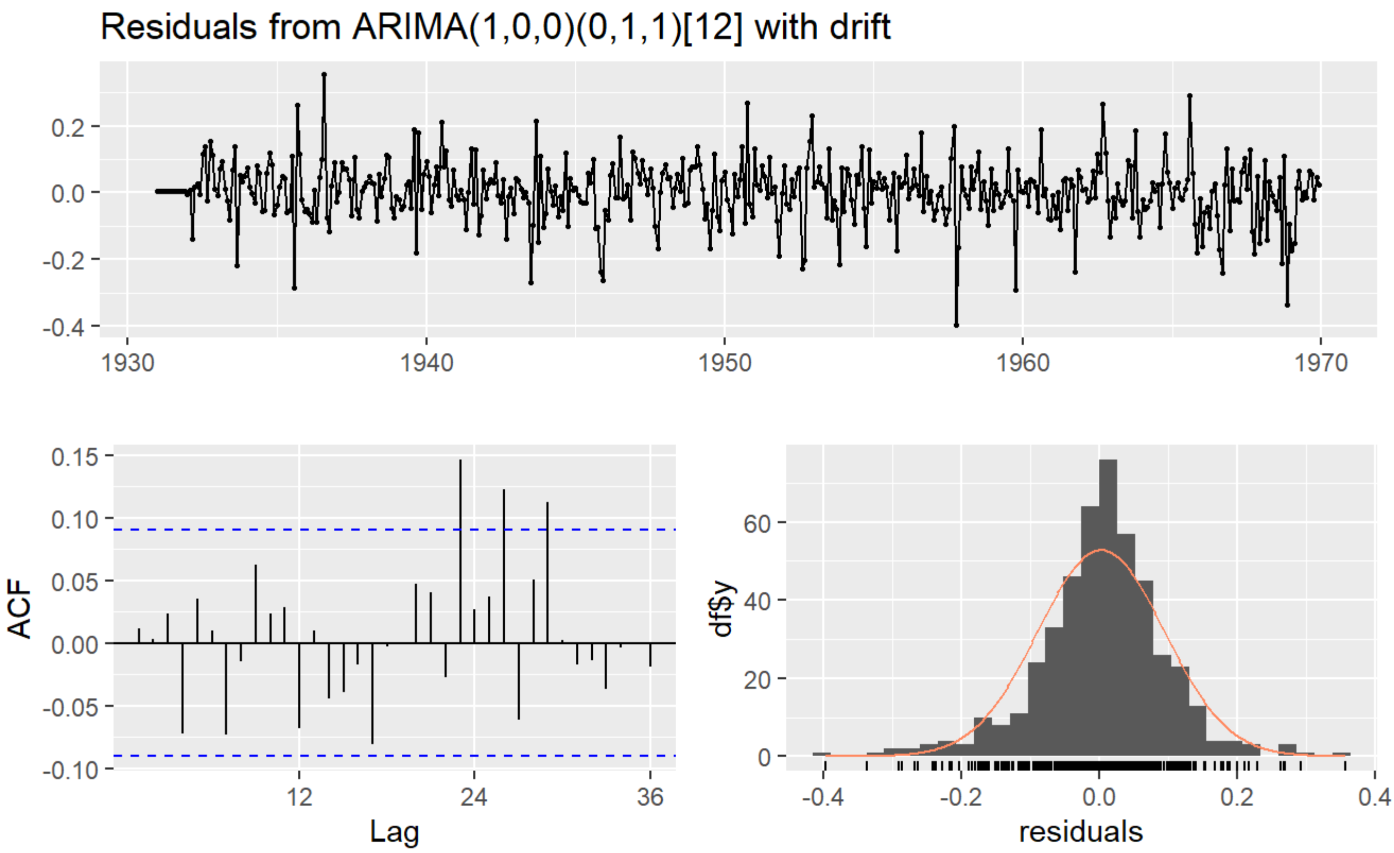
```
Series: varicelle_train_ts
ARIMA(1,0,0)(0,1,1)[12] with drift
Box Cox transformation: lambda= -0.1562783

Coefficients:
         ar1     sma1   drift
      0.6861  -0.7983  -5e-04
s.e.  0.0341   0.0282   3e-04

sigma^2 = 0.008727:  log likelihood = 429.11
AIC=-850.21   AICc=-850.12   BIC=-833.72

Training set error measures:
```

```
                    ME     RMSE      MAE    MPE     MAPE      MASE      ACF1
Training set 23.33882 157.3905 102.9085 -1.455 16.97148 0.4234344 0.2479118
```

Hide

```
forecast::checkresiduals(sarima_model)
```

```

    Ljung-Box test

data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
Q* = 29.164, df = 22, p-value = 0.1403

Model df: 2.   Total lags used: 24
```



Let’s try to compare this new model to the first one obtained by the auto.arima which has some significant auto-correlations.

Hide

```

actual <- base::as.numeric(varicelle_test_ts)

# Predicitons on test set
forecast1 <- forecast::forecast(auto_sarima_model, h = h)
forecast2 <- forecast::forecast(sarima_model, h = h)
pred1 <- base::as.numeric(forecast1$mean)
pred2 <- base::as.numeric(forecast2$mean)

# Model 1
```

```
mae1 <- base::mean(base::abs(pred1 - actual))
rmse1 <- base::sqrt(base::mean((pred1 - actual)^2))
mape1 <- base::mean(base::abs((pred1 - actual)/actual)) * 100

# Model 2
mae2 <- base::mean(base::abs(pred2 - actual))
rmse2 <- base::sqrt(base::mean((pred2 - actual)^2))
mape2 <- base::mean(base::abs((pred2 - actual)/actual)) * 100

# Print results
base::cat("Auto.arima SARIMA : MAE =", mae1, ", RMSE =", rmse1, ", MAPE =", mape1, "%
\n")
```

```
Auto.arima SARIMA : MAE = 120.3257 , RMSE = 151.7628 , MAPE = 38.72478 %
```

Hide

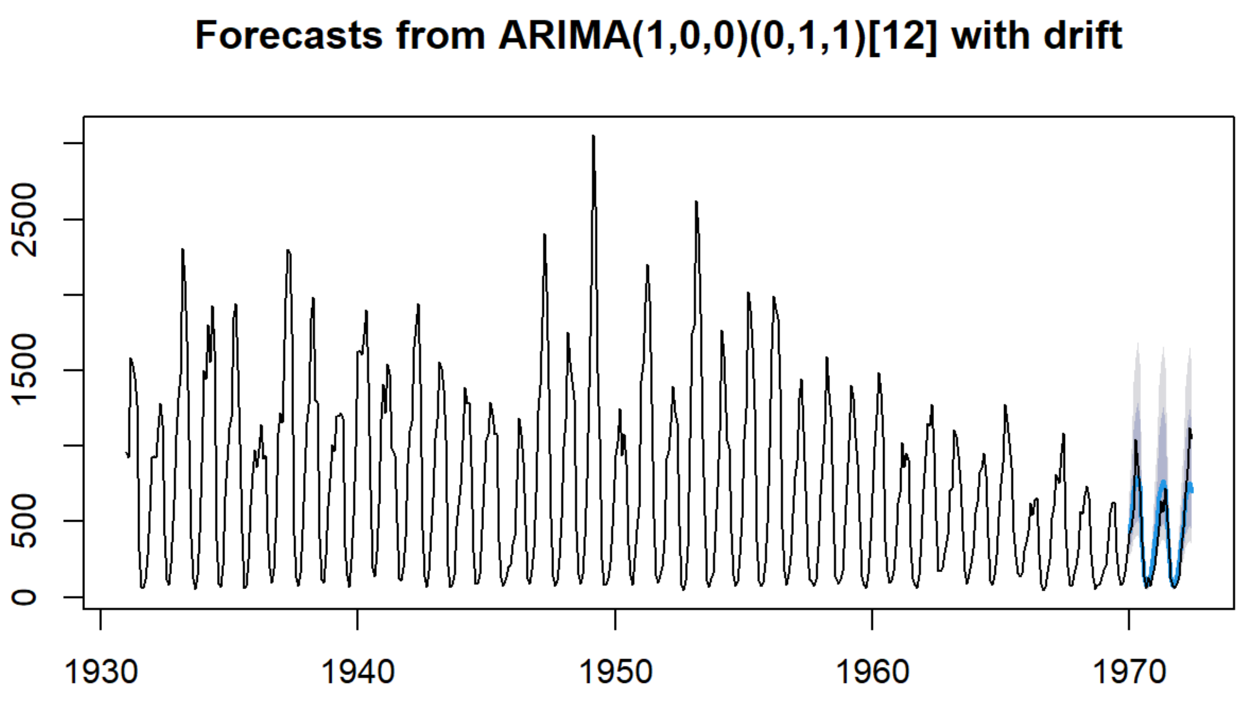
```
base::cat("SARIMA optimized : MAE =", mae2, ", RMSE =", rmse2, ", MAPE =", mape2, "%
\n")
```

```
SARIMA optimized : MAE = 97.04403 , RMSE = 137.1524 , MAPE = 28.94554 %
```

Our optimized SARIMA version is much better than than the SARIMA model proposed by the auto.arima() in the first place, and way much better than the best Holt-Winters model found in the previous section.

Hide

```
# Generate forecast
sarima_forecast <- forecast::forecast(sarima_model, h = h)

# Plot forecast with historical data
graphics::plot(sarima_forecast)
graphics::lines(varicelle_ts)
```



Hide

```
base::rm(
  auto_sarima_model, mae1, rmse1, mape1, forecast1, forecast2,
  pred1, pred2, new_train_ts
)
```

Random Forest models

Firstly, we do a variable selection procedure using the VSURF package (for more information, you can look to this link https://journal.r-project.org/archive/2015-2/genuer-poggi-tuleaumalot.pdf), then with the selected variable we will compute the model to use. We decide to use only up to 24 features in the model.

Hide

```
data <- base::as.vector(varicelle_train_ts)[1:24]
for (i in 1:(base::length(as.vector(varicelle_train_ts))-24)) {
  data <- base::rbind(data, base::as.vector(varicelle_train_ts)[(i+1):(i+24)])
}
base::rownames(data) <- base::seq(1, base::nrow(data))
base::colnames(data) <- paste0("X", 1:24)

base::set.seed(23062025)
data_vsurf <- VSURF::VSURF(x = data[,-24], y = data[,24], verbose = F)

data_vsurf$varselect.pred
```

```
[1] 23 12  1 22  2  6
```

Let’s fit the model with the covariates suggested by VSURF, then construct all necessary elements to forecast varicelle_test_set.

Hide

```
selected_cols <- data_vsurf$varselect.pred
rf_model <- randomForest::randomForest(
  x=data[,c(selected_cols)],
  y=data[,24],
  mtry = base::ncol(data[,c(selected_cols)]) # Use all covariates in each tree construc
ted
)
```

Hide

```
# Combine the end of train and all of test to get a continuous series
full_series <- c(
  base::as.vector(varicelle_train_ts)[(length(varicelle_train_ts)-23):base::length(vari
celle_train_ts)],
  base::as.vector(varicelle_test_ts)
)

# Build the test data matrix: each row is a window of length 24
test_data <- NULL
n_windows <- base::length(varicelle_test_ts)

for (i in 1:n_windows) {
  test_data <- base::rbind(test_data, full_series[i:(i+23)])
}

base::rownames(test_data) <- base::seq_len(base::nrow(test_data))
base::colnames(test_data) <- paste0("X", 1:24)

# Predict the next value for each window in test data
rf_forecast <- stats::predict(rf_model, newdata = test_data[, c(selected_cols)])

mae_rf <- base::mean(base::abs(rf_forecast - actual))
rmse_rf <- base::sqrt(base::mean((rf_forecast - actual)^2))
mape_rf <- base::mean(base::abs((rf_forecast - actual)/actual)) * 100

# Print results
base::cat("Random Forest : MAE =", mae_rf, ", RMSE =", rmse_rf, ", MAPE =", mape_rf, "%
\n")
```

```
Random Forest : MAE = 163.4372 , RMSE = 208.2039 , MAPE = 51.23152 %
```

Hide

```
base::rm(data, test_data, n_windows, i, full_series, selected_cols, data_vsurf, actual)
```

As shown by the performance on the test set, even though the Random Forest is better than the Holt-Winters model, it's worse than the optimized SARIMA model.

Hide

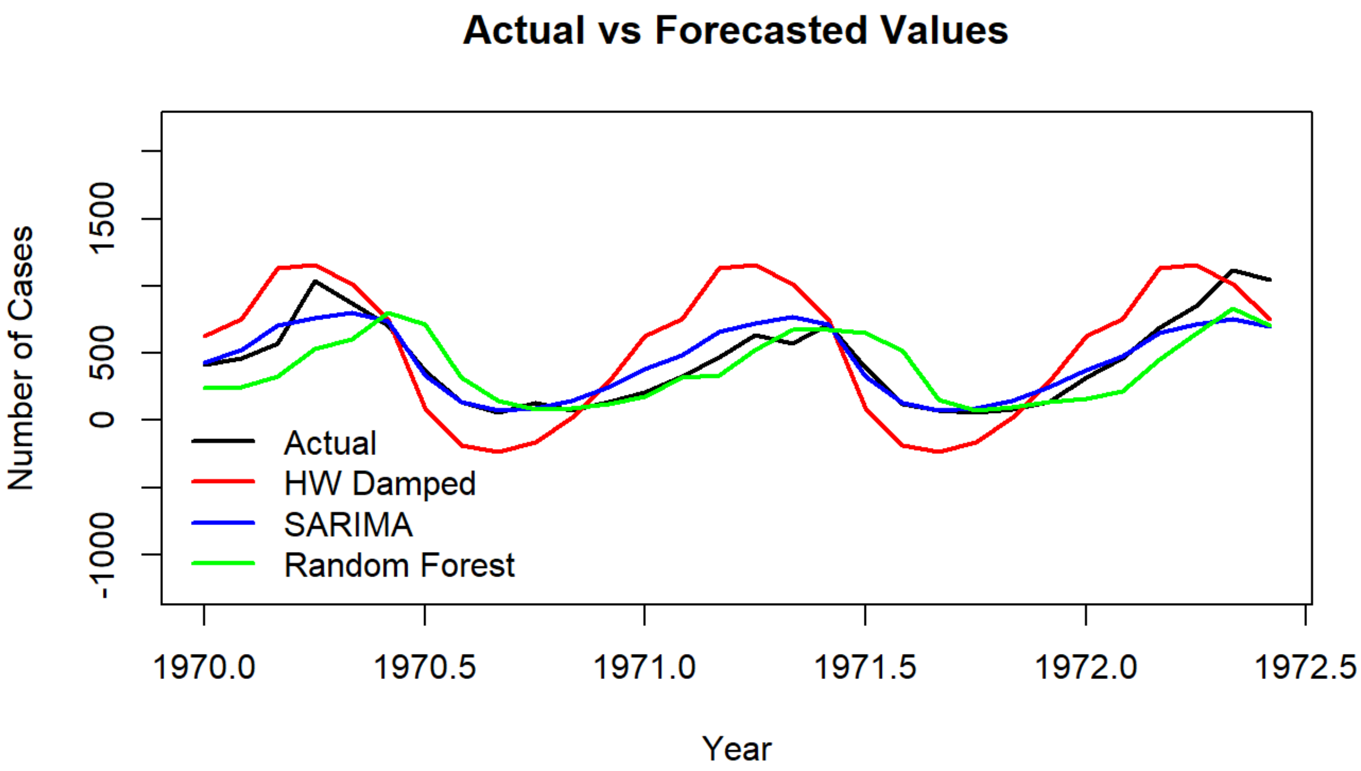
```
# Compute y-axis limits to cover all series
y_min <- base::min(
  varicelle_test_ts, hw_damped_model$mean, sarima_forecast$mean, rf_forecast)
y_max <- base::max(
  varicelle_test_ts, hw_damped_model$mean, sarima_forecast$mean, rf_forecast)

# Plot actual values with custom y-axis limits
graphics::plot(varicelle_test_ts, type = "l", col = "black", lwd = 2,
     main = "Actual vs Forecasted Values",
     xlab = "Year", ylab = "Number of Cases",
     ylim = c(y_min - 1000, y_max + 1000))

# Add forecast lines
graphics::lines(hw_damped_model$mean, col = "red", lwd = 2)
```

Hide

```
graphics::lines(sarima_forecast$mean, col = "blue", lwd = 2)
graphics::lines(
  stats::ts(rf_forecast, start = c(1970, 1), end = c(1972, 6), frequency = 12),
  col = "green", lwd = 2)
```

Hide

```
# Add a legend
graphics::legend("bottomleft",
       legend = c("Actual", "HW Damped", "SARIMA", "Random Forest"),
       col = c("black", "red", "blue", "green"),
       lwd = 2,
       bty = "n")
```



Conclusion

We evaluate a range of time series forecasting methods on the monthly varicelle cases, and a SARIMA model was the best (a MAPE only up to 29%). Although the introduction and use of Machine Learning methods is very interesting, in this case they (Random Forest) weren't able to provide better results than the SARIMA model, even though they were able to perform better than Holt-Winters models.

Another fact to precise on this education case is that due to the limited number of observations (only 498 values) on the initial time series, we're not able to use cross-validation to evaluate models, but instead we used a fixed train and test sets. In the case of a much larger number of samples, using it would enables us to get more precise evaluation of models' performance.

Hide

```
# Plot actual data
graphics::plot(
  varicelle_test_ts,
  type = "l", col = "black", lwd = 2,
  main = "Actual vs Forecasted Values", xlab = "Year", ylab = "Number of Cases",
  ylim = base::range(
    c(
      varicelle_test_ts,
      sarima_forecast$lower[,2] - 100,
      sarima_forecast$upper[,2] + 1000)
    )
)

# Add mean forecast
graphics::lines(sarima_forecast$mean, col = "blue", lwd = 2)
```

Hide

```r
# Add confidence intervals (95% shown; adjust [,2] for other levels)
graphics::polygon(
  x = c(stats::time(sarima_forecast$mean), rev(stats::time(sarima_forecast$mean))),
  y = c(sarima_forecast$lower[,2], base::rev(sarima_forecast$upper[,2])),
  col = adjustcolor("grey", alpha.f = 0.3),
  border = NA
)

# Add legend
graphics::legend("topleft",
       legend = c("Actual", "Forecast", "95% CI"),
       col = c("black", "blue", "grey"),
       lwd = c(2, 2, 10),
       lty = c(1, 1, 1),
       pch = c(NA, NA, 15))
```