```
##HoltWinter model
#Load the desired package
library(forecast)
# Create the Holt-Winters additive model
hw_additive <- HoltWinters(flu_ts_train, seasonal = "additive")
# Create the Holt-Winters multiplication modelhw_multiplicative <- HoltWinters(flu_ts_train,
seasonal = "multiplicative")

# The SSE and RMSE were calculated for the Holt-Winters additive modelsse_additive <-
sum((flu_ts_train - hw_additive$fitted)^2)
rmse_additive <- sqrt(mean((flu_ts_train - hw_additive$fitted)^2))

# The SSE and RMSE were calculated for the Holt-Winters multiplicative model
sse_multiplicative <- sum((flu_ts_train - hw_multiplicative$fitted)^2)
rmse_multiplicative <- sqrt(mean((flu_ts_train - hw_multiplicative$fitted)^2))

# Fitting the olt-Winters additive model
hw_model <- hw(flu_ts_train, seasonal = "additive")
# Number of influenza predicted at future time points
forecasted_flu_cases <- forecast(hw_model, h = 12)    # Predict the number of episodes in the
next 12 months
#Print the model overview and prediction results
summary(hw_model)
print(forecasted_flu_cases)
# Get the fit values on the training set
hw_model_values <- fitted(hw_model)
print(hw_model_values)
write.xlsx(hw_model_values, file
="C:/Users/User/Desktop/ARIMA--X/2013-2021/Holt-Winters/The training set was fitted to the
value.xlsx")
# count MSE
mse <- mean((flu_ts_train - hw_model_values)^2)
# countMAE
mae <- mean(abs(flu_ts_train - hw_model_values))
# countRMSE
rmse <- sqrt(mean((flu_ts_train - hw_model_values)^2))

data <- data.frame(observed = flu_ts_train, predicted = hw_model_values)
# By specifying type = "all", we can calculate the confidence intervals for the MSE, MAE, and
RMSE simultaneously
boot_MSE <- boot(data, calc_MSE, R = 1000)
boot_MAE <- boot(data, calc_MAE, R = 1000)
boot_RMSE <- boot(data, calc_RMSE, R = 1000)
```

```
# Output results
boot_MSE_ci <- boot.ci(boot_MSE, type = "all", conf = 0.95)
boot_MAE_ci <- boot.ci(boot_MAE, type = "all", conf = 0.95)
boot_RMSE_ci <- boot.ci(boot_RMSE, type = "all", conf = 0.95)

print("95% Confidence Interval for MSE:")
print(boot_MSE_ci)
print("95% Confidence Interval for MAE:")
print(boot_MAE_ci)
print("95% Confidence Interval for RMSE:")
print(boot_RMSE_ci)

##Fit values on the fitted test set
forecasted_flu_cases <- forecast(hw_model, h = 12)
write.xlsx(forecasted_flu_cases, file ="C:/Users/User/Desktop/ARIMA--X/2013-2021/Holt-Winters/The test set fits the value.xlsx")
# Save the prediction results as time-series objects
forecast_ts <- as.ts(forecasted_flu_cases$mean)
# count MSE
mse1 <- mean((flu_ts_test - forecast_ts)^2)
# count MAE
mae1 <- mean(abs(flu_ts_test - forecast_ts))
# count RMSE
rmse1 <- sqrt(mean((flu_ts_test - forecast_ts)^2))

data <- data.frame(observed = flu_ts_test, predicted = forecast_ts)
# By specifying type = "all", we can calculate the confidence intervals for the MSE, MAE, and RMSE simultaneously
boot_MSE <- boot(data, calc_MSE, R = 1000)
boot_MAE <- boot(data, calc_MAE, R = 1000)
boot_RMSE <- boot(data, calc_RMSE, R = 1000)

# Output results
boot_MSE_ci <- boot.ci(boot_MSE, type = "all", conf = 0.95)
boot_MAE_ci <- boot.ci(boot_MAE, type = "all", conf = 0.95)
boot_RMSE_ci <- boot.ci(boot_RMSE, type = "all", conf = 0.95)

print("95% Confidence Interval for MSE:")
print(boot_MSE_ci)
print("95% Confidence Interval for MAE:")
print(boot_MAE_ci)
print("95% Confidence Interval for RMSE:")
print(boot_RMSE_ci)
```

```
# Draw the training set, prediction set and prediction resultsplot(flu_ts, xlim = c(2013, 2023), ylim = c(min(flu_ts), max(flu_ts)), main = "Forecasts from Holt-Winters ",
     ylab="Influenza cases",xlab="")
axis(1, at = seq(2013, 2022, 1), labels = seq(2013, 2022, 1))
lines(flu_ts,pch=1, lty=1,col = "grey", lwd = 2)    #    true value
lines(hw_model_values, lty=1,col = "green", lwd = 2)    # The training set was fitted to the value
lines(forecast_ts, lty=6,col = "green", lwd = 2)    # Forecast results
legend("topleft", legend = c("Actual", "Holt-Winters fitted", "Holt-Winters forecast"), col = c("grey", "green", "green"),lty=c(1, 1, 6), lwd = 3)
```